

Dinamička web aplikacija za razmjenu usluga

Petak, Ivan

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://um.nsk.hr/um:nbn:hr:200:349443>

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

Download date / Datum preuzimanja: **2024-12-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Stručni studij

**DINAMIČKA WEB APLIKACIJA ZA RAZMJENU
USLUGA**

Završni rad

Ivan Petak

Osijek, 2017.

Sadržaj

1. UVOD	1
2. APLIKACIJE ZA RAZMJENU USLUGA	2
2.1. Zahtjevi na i funkcionalnosti web aplikacije za razmjenu vještina	3
3. TEHNOLOGIJE ZA IZRADU WEB APLIKACIJE.....	4
3.1. HTML.....	4
3.2. CSS	5
3.3. JavaScript.....	5
3.3.1. jQuery.....	6
3.4. PHP.....	7
3.5. MySQL.....	7
3.6. Bootstrap.....	8
4. POSTUPAK IZRADU I VREDNOVANJA PROGRAMSKOG RJEŠENJA	9
4.1. Raspored elemenata naslovnice.....	9
4.2. Poveznice na vanjske dokumente	10
4.3. Izrada navigacijske trake	11
4.3.1. Primarna navigacijska traka	11
4.3.2. Sekundarna navigacijska traka	12
4.4. Dinamička pohrana vještina	13
4.5. Sustav pretrage i kontaktiranja korisnika	16
4.6. Uređivanje izgleda web aplikacije za razmjenu usluga.....	20
4.7. Vrednovanje programskog rješenja	23
5. ZAKLJUČAK	25
SAŽETAK.....	26
ABSTRACT	27
ŽIVOTOPIS	28
LITERATURA.....	29

1. UVOD

Učenje je postalo temelj današnjeg društva koje se prenosi s generacije na generaciju i istodobno se širi nevjerojatnom brzinom. Čovjek više nije primoran učiti kako bi preživio, no toliko toga postoji za naučiti pa ljudi uče određene vještine jer im one donose neke osjećaje zadovoljstva ili zabave kao što su ples, stvaranje glazbe i još mnoge. Za olakšani pristup učenju takvih vještina bit će stvorena web aplikacija za razmjenu usluga pod nazivom „XchangeSkills“. Web aplikacija omogućavat će krajnjim korisnicima mogućnost učenja i podučavanja vještina bez ikakvih novčanih naknada po principu „usluga za uslugu“. Kako bi mogla omogućiti takav princip, web aplikacija će se morati sastojati od korisničkog računa korisnika s mogućnošću unosa već naučenih vještina, te vještina koje korisnik želi naučiti. Nakon unosa vještina, potrebno je razviti sustav pretrage između ostalih korisnika kako bi se pronašli rezultati koji odgovaraju prethodno pohranjenim vještinama. Pri pronalasku rezultata koji odgovaraju kriterijima, korisnik bi trebao dobiti mogućnost kontaktiranja i dogovora s pronađenim korisnikom oko potencijalne razmjene vještina. Pri izradi web aplikacije korištene su osnovne tehnologije za izradu web sadržaja.

U nastavku rada, tj. poglavlju 2 opisat će se aplikacije za razmjenu usluga, njihove funkcionalnosti i tipovi kao i zahtjevi koje web aplikacija „Xchange Skills“ treba ispunjavati. Poglavlje 3 sadrži primjere i kratke opise web tehnologija korištenih u izradi web aplikacije „Xchange Skills“. U poglavlju 4 detaljno su opisani načini na koji su web tehnologije korištene u izradi aplikacije „Xchange Skills“.

2. APLIKACIJE ZA RAZMJENU USLUGA

Aplikacije za razmjenu usluga su web servisi koji funkcioniraju na principu razmjene usluga. Princip po kojem korisnici razmjenjuju svoje znanje ili materijalnu imovinu u zamjenu za znanje ili materijalnu imovinu drugih korisnika istog servisa za razmjenu usluga. Razmjena započinje registracijom na jedan od mnogih servisa za razmjenu usluga te unosom stvari koje korisnik želi zamijeniti za nešto što je njemu potrebno. Nakon toga može pregledati što se sve nudi na određenom servisu ili jednostavno može čekati dok mu se ne jave ostali korisnici koji su voljni sudjelovati u razmjeni za materijalna dobra ili znanja koje je registrirani korisnik voljan razmijeniti. Razmjenom se stvaraju nova poznanstva i omogućuje korisnicima da dobiju upravo ono što žele bez ikakvih novčanih naknada. Jedina valuta na web aplikacijama za razmjenu usluga su zapravo usluge. Neke od najpoznatijih web aplikacija za razmjenu usluga su:

- Swapz – aplikacija koja omogućava razmjenu svih vrsta materijalne imovine od auta do antiknih vaza. Uz razmjenu materijalnih dobara, aplikacija omogućava razmjenu vještina te ponuda za posao. Unutar web aplikacije *swapz* također je moguća prodaja i kupnja materijalnih dobara, vještina ili ostalih potreba za kojima korisnik traga.
- Read it swap it – aplikacija koja omogućava razmjenu knjiga. Pročitane knjige ili knjige koje stoje na policama su možda upravo one za kojima neki ljudi tragaju. Korisnik na svoj profil upiše detalje o knjizi koju želi zamijeniti za nešto što je njemu potrebno ili koje se jednostavno želi riješiti. Ostali korisnici šalju zahtjeve korisniku s tom knjigom te dogovaraju potencijalnu razmjenu.
- Green plant swap – aplikacija koja omogućava korisnicima, koji su obožavatelji biljaka i vrtlarenja, da se pronađu i razmjene različite vrste biljaka. Omogućava pretragu za korisnicima u vašem području te potiče vrtlare da si međusobno pomažu i razmjenjuju vještine vrtlarenja kao i različitih vrsta biljaka.
- Swap a skill – aplikacija koja omogućava korisnicima da razmjenjuju vještine kao usluge jedni drugima. Ovisno o vještinama koje su potrebne jednom korisniku, on će ih potražiti pomoću ove aplikacije te pronaći osobe koje su tu vještinu svladale, a korisnik će u zamjenu ponuditi neku od svojih vještina kako bi razmijenili vještine kao uslugu jedno drugome.

Bez obzira na stvari koje se razmjenjuju, web aplikacije za razmjenu usluga imaju jednu stvar koja ih povezuje, razmjena vještina bez novčane naknade. Kako bi naučili ili podučavali vještine nije potrebna nikakva novčana naknada te je korištenje web aplikacije besplatno.

2.1. Zahtjevi na i funkcionalnosti web aplikacije za razmjenu vještina

Dok su u drugim zemljama aplikacije za razmjenu usluga uvelike prihvaćene, unutar Republike Hrvatske još uvijek ne postoji servis koji omogućava razmjenu vještina po principu razmjene usluga. Građani koriste princip razmjene usluga većinom verbalnom razmjenom, no samo mala populacija unutar određene regije. To dovodi do problema da neke potrebne vještine unutar te regije još nije nitko savladao. Kako su upravo te vještine potrebne, građani se odlučuju na plaćanje novčane naknade da pronađu osobu koja je tu određenu vještinu svladala. Problem koji se dalje razvija je da građani kojima su potrebne određene vještine nemaju potrebne resurse za plaćanje novčane naknade. Web aplikacija za razmjenu usluga „Xchange Skills“ je aplikacija s bazom podataka u kojoj su pohranjeni podaci o građanima kojima je potrebna određena vještina i koji su spremni na razmjenu vještina na principu razmjene usluga.

Kako bi se korisnicima omogućila razmjena vještina po principu razmjene usluga, na web aplikaciju „Xchange Skills“ navode se sljedeći zahtjevi u obliku funkcionalnosti koje ona treba sadržavati:

- registracija korisnika – pohrana korisničkog imena i podataka o korisniku u bazu podataka. Podaci poput lozinke bi se trebali prilikom unosa u bazu podataka kodirati zbog sigurnosti očuvanja osjetljivih podataka,
- prijava korisnika – prilikom prijave na korisnički račun, potrebna je provjera odgovaraju li uneseni podaci administratorskom ili korisničkom računu te bi, na temelju toga, korisnik bio preusmjeren na odgovarajuću stranicu profila,
- unos vještina – nakon prijave s odgovarajućim korisničkim podacima, potrebno je omogućiti korisniku prikaz polja za unos vještina. Polja bi trebala biti podijeljena na dvije podvrste – naučene vještine i željene vještine. Unesene vještine se pohranjuju u bazu podataka zajedno s korisničkim imenom,
- pretraga ostalih korisnika – korisniku bi se trebao omogućiti pregled ostalih registriranih korisnika koji odgovaraju kriterijima pretrage. Kriteriji pretrage bit će preuzeti iz baze podataka u obliku prethodno unesenih vještina prema kojima će se baze pretraživati za ostalim korisnicima sličnih potreba,
- kontaktiranje ostalih korisnika – nakon pretrage između ostalih registriranih korisnika, prijavljeni korisnik bi trebao imati mogućnost odabira jednog od pronađenih korisnika. Pri odabiru, prijavljenom bi korisniku trebalo omogućiti opciju kontaktiranja odabranog korisnika kako bi se daljnji način razmjene vještina detaljno dogovorio.

3. TEHNOLOGIJE ZA IZRADU WEB APLIKACIJE

Kako bi web aplikacija za razmjenu vještina na principu razmjene usluga, „Xchange Skills“, ispunjavala sve zahtjeve navedene u prethodnom poglavlju, bit će korištene osnovne web tehnologije za izradu web sadržaja. Razlog korištenja osnovnih web tehnologija je taj što one dopuštaju slobodu uređivanja izgleda i rasporeda elemenata web aplikacije ručnim unosom programskog koda. Kako bi se zahtjevi što učinkovitije ispunili i kako bi se što bolje izradili dijelovi web aplikacije „Xchange skills“, za ispunjavanje svakog zahtjeva koristit će se u nastavku opisani programski, stilski i skriptni jezici.

3.1. HTML

Prema [2] HTML je kratica za „HyperText Markup Language“ što se grubo prevodi u prezentacijski jezik korišten za izradu web stranica. Web preglednici koriste HTML za tumačenje kako se određeni sadržaj prikazuje unutar preglednika. HTML popisuje strukturu Web stranice ili aplikacije. HTML se često naziva programskim jezikom što on nije, unutar njega nije moguće izvršavati nikakve operacije te ga to čini prezentacijskim jezikom.

Prvi javno dostupan opis HTML jezika napisao je fizičar Tim Berners-Lee 1991. kao što je spomenuto u [3]. Taj opis sadrži 18 početnih elemenata potrebnih za stvaranje HTML dokumenta. Tim Berners-Lee radio je za Europsku organizaciju za nuklearno istraživanje (fra. *Conseil Européen pour la Recherche Nucléaire*, CERN) dok je dobio ideju za izradu univerzalnog HTML jezika koji će biti korišten diljem interneta. Godine 1990. Tim Berners-Lee je predložio zahtjev za projekt o stvaranju HTML jezika CERN-u, no zahtjev je odbijen. Nakon toga Tim Berners-Lee 1994. napušta radno mjesto u CERN-u te osniva organizaciju pod imenom „World Wide Web Consortium“ ili „W3C“ koja se bavi standardizacijom internetskih tehnologija.

Na slici 3.1. prikazan je primjer jednostavne web stranice pisane u HTML-u. U navedenoj web stranici, prvo se upisuje deklaracija HTML-a koja govori pregledniku u kojoj verziji HTML-a je stranica izrađena (!DOCTYPE html). Upisuje se oznaka „html“ koja označava početak HTML dokumenta te oznaka „head“ unutar koje se upisuje naslov kartice („title“) i po potrebi veze na vanjske JavaScript, CSS ili neke druge dokumente. Nakon toga, otvara se tijelo HTML dokumenta oznakom „body“ gdje se upisuje sadržaj stranice te se na kraju zatvara istom oznakom uz dodavanje znaka „/“.

```

<!DOCTYPE html>
<html>
<head>
<title>Naziv stranice</title>
</head>
<body>
<p>Sadržaj stranice</p>
</body>
</html>

```

Sl. 3.1. Primjer jednostavnog HTML koda

3.2. CSS

CSS je kratica za „Cascading Style Sheets“. CSS je stilski jezik koji služi za opisivanje načina na koji će se prikazati sadržaj prethodno napisanog, pomoću prezentacijskog jezika HTML kao što je opisano u [4, str.121]. Prije pojave CSS-a, HTML je uključivao određene znakove kako bi opisao prikaz sadržaja. Ubrzo je uočeno da je opisivanje stila stranice unutar samog HTML-a pomalo nezgodno, te je potreban poseban jezik koji će služiti samo za raspored i izgled HTML sadržaja. CSS primarno služi za odvajanje sadržaja dokumenta od načina prikaza tog sadržaja. To omogućava većem broju HTML dokumenata da koriste jedan CSS dokument te da sadržaj bude dosljedan kroz više web stranica. Također, omogućava opciju učitavanja različitih CSS dokumenata s obzirom na način učitavanja stranice. Pod načinom učitavanja, podrazumijevaju se različite veličine prikaza, dodirne površine, zvučno iščitavanje ili učitavanje sa dodirnih površina koje koriste Brailovo pismo.

Slika 3.2. prikazuje jednostavni CSS dokument unutar kojeg se prvom linijom koda opisuje tip stila dokumenta. Nakon toga, upisuje se naziv elementa iz HTML dokumenta koji je povezan sa CSS dokumentom. Nadalje, otvaraju se zagrade unutar kojih se određuju parametri sadržaja koji se prikazuje unutar elementa s nazivom.

```

<style type="text/css">
p
{
color: red;
}
</style>

```

Sl. 3.2. Primjer jednostavnog stila napisanog u CSS-u

3.3. JavaScript

JavaScript je, za razliku od HTML-a i CSS-a, programski jezik, kao što je spomenuto u [4, str.192.] JavaScript podržava objektno orijentirano programiranje i dijelom je nastao od

programskog jezika JAVA te također uz utjecaj programskih jezika „Self“ i „Scheme“. Funkcija mu je izvršavanje operacija i zadaća unutar web preglednika bez potrebe za dodatnim priključcima instaliranim na web preglednik. JavaScript kod može se nalaziti u istom HTML dokumentu kao i ostali sadržaj, a može ga se smjestiti i u vanjski dokument (s nastavkom .js) te povezati na HTML. Dok je sadržaj HTML-a većinom statičan, sadržaj JavaScripta je upravo suprotno – izvršava se nakon što je stranica učitana pritiskom na tipku ili na određeni element, točnije interakcijom korisnika s elementima JavaScripta.

Slika 3.3. prikazuje funkciju pod imenom „zbroj“ napisanu unutar oznaka „script“ unutar kojih se pišu sve naredbe JavaScripta. Funkcija zbroj prvo stvara dvije poznate varijable x i y te postavlja njihove vrijednosti na 1 i 2. Nakon toga stvara treću varijablu z kojoj je vrijednost zbroj prethodne dvije varijable.

```
<script>
function zbroj {
  var x = 1;
  var y = 2;
  var z = x + y;
}
</script>
```

Sl. 3.3. *Primjer jednostavne funkcije napisane JavaScriptom*

3.3.1. jQuery

jQuery zasebno nije skriptni niti programski jezik već skupina pojednostavljenih naredbi JavaScript-a koja se još naziva i knjižnica (eng. *library*). Omogućava pojednostavljeno stvaranje animacija, elemenata web stranica te omogućava i upravljanje ostalim skriptama unutar HTML dokumenta. Kako bi se mogle koristiti naredbe jQuery-a, potrebno je navesti izvor jQuery knjižnice kao što je prikazano prvom linijom programskog koda na slici 3.4.

Slika 3.4. prikazuje jQuery skriptu koja omogućava postepeno prikazivanje skrivenog elementa „element_id“. U prvoj liniji koda, dokument se povezuje sa jQuery knjižnicom na internetu. Nakon toga je postavljeno ime tipke „button_id“, pritiskom na koju se poziva funkcija koja postepeno prikazuje element. Moguće je dodavanje parametara postepenog prikazivanja elementa kao što su brzina, veličina, položaj unutar HTML dokumenta i slično.

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
<script>
    $( "#button_id" ).click(function()
    {
        $("#element_id").fadeIn();
    });
</script>

```

Sl. 3.4. Primjer jednostavne funkcije napisane jQuery naredbama uz uključenu jQuery knjižnicu.

3.4. PHP

Prema [4, str.480.] PHP (eng. *Hypertext Preprocessor*) je programski jezik korišten najčešće u izradi web sadržaja, no koristi se i u opće programerske svrhe. PHP je orijentiran prema sintaksi programskih jezika C i Perl. Jedan je od mnogih programskih jezika takve vrste, no ističe se podrškom mnogih baza podataka i internet protokola. Stvoren je 1995. godine s ciljem izrade dinamičkih web aplikacija ili stranica. PHP skripte, za razliku od HTML-a i JavaScript-a, izvršava server te se rezultat vraća i obrađuje unutar HTML-a i JavaScripta. Poput JavaScripta i CSS-a, PHP se može upisivati izravno u HTML, no može se i povezati na vanjsku PHP skriptu.

Slika 3.5. prikazuje skriptu pisanu PHP-om koja vraća HTML-u vrijednost „Hello World“. Skripta započinje otvaranjem oznaka koji označavaju da se radi o PHP skripti (<?php i ?>). Unutar skripte vidimo naredbu „echo“ koja ima značenje „ispis“.

```

<?php
echo '<p> Hello World </p>';
?>

```

Sl. 3.5. Primjer jednostavnog ispisa korištenjem PHP-a

3.5. MySQL

Prema [5] MySQL je stvoren po SQL (eng. *Structured Query Language*) jeziku te služi za upravljanje bazom podataka. Raširen je diljem svijeta jer je besplatan, a njegov je kod dostupan svima. Ima podršku mnogih programskih jezika, kao što su PHP, Java, Python, što mu daje prednost nad ostalim, isto orijentiranim, programskim jezicima za upravljanje bazama podataka. Baze MySQL-a relacijskog su tipa što omogućuje brzo i učinkovito skladištenje podataka te njihovo čitanje. To je postala neophodna potreba većine poslovnih subjekata čije se poslovanje temelji na brzini pristupa podacima.

Slika 3.6. prikazuje tri najčešće korištene naredbe SQL-a koje koristi MySQL:

- naredba „SELECT“ odabire podatke iz baze podataka sa zadanim parametrima (najčešće naziv stupca i naziv tablice),
- naredba „UPDATE“ mijenja podatke unutar tablice pod nazivom „ime_tablice“, te unutar stupca pod nazivom „ime_stupca“,
- naredba „DELETE FROM“ briše podatke iz tablice (tj. stupca) pod nazivom „ime_tablice“ ako podaci unutar stupca odgovaraju zadanim (,„vrijednost“).

```
1 SELECT ime_stupca FROM ime_tablice;
2 UPDATE ime_tablice SET ime_stupca='vrijednost';
3 DELETE FROM ime_tablice WHERE ime_stupca='vrijednost';
```

Sl. 3.6. Primjer često korištenih naredbi napisanih SQL-om

3.6. Bootstrap

Prema [6] Bootstrap je besplatan programski okvir za izradu web stranica i web aplikacija otvorenog koda. On sadrži pohranjene predloške koji stvaraju, ili samo mijenjaju, elemente unutar HTML dokumenta web stranice. Sadrži i JavaScript proširenja za promjenu sadržaja unutar web preglednika. Bootstrap se, za razliku od ostalih okvira, bavi samo dizajnerskim uređivanjem izgleda koda u web pregledniku. Bootstrap je jedan od najpogodnijih načina za izradu responzivne¹ web aplikacije.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-6">Prva polovica reda</div>
    <div class="col-sm-6">Druga polovica reda</div>
  </div>
</div>
```

Sl. 3.7. Primjer HTML programskog koda koji koristi Bootstrap koordinatnu mrežu odjeljaka

Na slici 3.7. prikazan je primjer podjele odjeljaka unutar Bootstrap-ove koordinatne mreže. Jedan red sastoji se od 12 manjih odjeljaka od kojih su, u ovom primjeru, korištena samo 2. Pozivaju se 2 odjeljka te im se postavljaju klase vrijednosti 6. Stvoreni odjeljci imaju svojstvo responzivnosti što je pretpostavljeno unutar Bootstrap programskog okvira.

¹ Responzivnost je svojstvo koje omogućava prilagodbu elemenata stranice ovisno o veličini zaslona na kojoj se prikazuje.

4. POSTUPAK IZRADE I VREDNOVANJA PROGRAMSKOG RJEŠENJA

U nastavku će biti objašnjen postupak izrade i vrednovanja programskog koda web aplikacije „Xchange Skills“. Opis postupka bit će podijeljen na glavne korake izrade programskog rješenja ove web aplikacije. Koraci postupka izrade web aplikacije podijeljeni su na potpoglavlja koja sadržavaju opise izrade programskih rješenja pojedinačnih zahtjeva koji su navedeni u poglavlju 2. Unutar svakog potpoglavlja opisan je način na koji su web tehnologije, prethodno objašnjene u poglavlju 3, korištene kako bi se određeni zahtjev na web stranicu ispunio, tj. kako bi se izradilo programsko rješenje koje rješava odgovarajući zahtjev. Na kraju poglavlja bit će opisan postupak vrednovanja programskog rješenja.

4.1. Raspored elemenata naslovnice

Raspored elemenata naslovnice označava način prikaza sadržaja pri otvaranju web aplikacije unutar web preglednika. Koncept mora zadovoljavati neke uvjete kako bi web aplikacija imala što efikasniji učinak na korisnika koji se njome služi. Cilj web aplikacije i njezinog dizajna jest zadržati korisnika na web stranici kako bi iskusio njezin puni potencijal. Pri tome treba paziti na nekoliko svojstava web stranice:

- količina informacija – korisnik će se lakše snalaziti na stranici te će mu ona izgledati privlačnije, ako nema puno redova teksta nabacanih na stranicu. Ako je potreban tekstualni opis, treba biti kratak te u što manje riječi opisivati svoj cilj,
- raspored sadržaja – sadržaj mora biti raspoređen smisljeno bez nepotrebno naglašenih manje bitnih dijelova koji odvlače pozornost korisnicima,
- dosljednost dizajna – odabrani i razrađeni dizajn mora biti konzistentan na svim stranicama i njezinim dijelovima.



Sl. 4.1. *Koncept osnovnog dizajna*

Slika 4.1. prikazuje izgled osnovnog dizajna te rasporeda elemenata pri izradi web aplikacije. Unutar zaglavlja nalazit će se navigacijska traka i logo web aplikacije. Sadržaj stranice, ovisno o interakciji korisnika, sastojat će se od:

- polja za unos podataka pri registraciji ili prijavi,
- polja za unos željenih vještina nakon prijave u korisnički račun,
- ispisa podataka o korisniku,
- pregleda poslanih i primljenih poruka,
- podnožja u kojemu će biti podaci potrebni za kontaktiranje administratora.

4.2. Poveznice na vanjske dokumente

Pri pisanju programskog koda, koji je potreban za izvršavanje nekog zadatka, može se dogoditi da je određena skripta potrebna u više različitih HTML dokumenata. Programski kod se može kopirati, odnosno može se ponovno pisati unutar HTML dokumenta web stranice ili se može spremiti u poseban dokument te na početku novog HTML dokumenta nove stranice samo stvoriti poveznicu na njega, kao što je prikazano na prvoj liniji programskog koda prikazanog na slici 4.2.

```
<script src="vanjska_skripta.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

Sl. 4.2. *Poveznice na vanjske dokumente*

U nastavku izrade ove web aplikacije koristit će se programski okviri prethodno objašnjeni u poglavlju 3. Na početku svakog HTML dokumenta web stranice upisane su poveznice na vanjske dokumente kao što je prikazano na slici 4.2. Poveznice u linijama 2, 3 i 4 omogućuju pristup dokumentima te korištenje istih, koji su spremljeni na udaljenim poslužiteljima. Upisane su poveznice na dva vanjska JavaScript dokumenta – biblioteka jQuery i Bootstrap, u kojima se nalaze predodređene skripte za promjenu sadržaja na stranici. Na posljednjoj liniji programskog koda sa slike upisana je i poveznica na vanjski dokument koji sadrži stilove Bootstrapa i omogućava korištenje boja te stilova iz Bootstrap programskog okvira.

4.3. Izrada navigacijske trake

Interaktivan sadržaj se sastoji od elemenata i objekata unutar HTML dokumenta koji omogućuju korisniku da upravlja određenim dijelom web stranice ili elementa. Takav sadržaj uvelike povećava ukupan dojam koji stranica ostavlja na korisnika te pomaže pri ravnoteži rasporeda unutar stranice. Unutar web aplikacije za razmjenu usluga nalazi se dosta interaktivnog sadržaja, no u nastavku će biti objašnjen jedan dio tog sadržaja pod nazivom „navigacijska traka“.

4.3.1. Primarna navigacijska traka

Navigacijska traka jest element na web aplikaciji koji omogućava korisniku kretanje između različitih web stranica. Ovisno o vrsti elemenata od kojih je napravljena, navigacijska traka će stvarati, uređivati ili preusmjeravati na nove elemente stranice. Kao što je prikazano na slici 4.3., traka se sastoji od 4 elementa koji vode na ostale dijelove web aplikacije te loga web aplikacije.



Sl. 4.3. Izgled primarne navigacijske trake

Na slici 4.4. prikazan je HTML kod za prikaz primarne navigacijske trake. Korišten je Bootstrap okvir za olakšani raspored elemenata te stalnu poziciju navigacijske trake na početku web preglednika. Za izgled trake korištene su klase koje se nalaze unutar Bootstrap okvira te koje su naknadno uređene u CSS dokumentu ove aplikacije. Unutar elemenata „Registracija“ i „Prijava“ koji pokreću prozore za unos podataka, nalazi se naredba za izvršenje funkcije „erase“ koja briše sve prethodno unesene, no ne spremljene podatke.

```

<div class="collapse navbar-collapse" id="navig">
  <ul id="list" class="nav navbar-nav navbar-text">
    <li>
      <button type="button" onClick="window.location.assign('index.php');">
        <span class="glyphicon glyphicon-home"></span> Naslovnica</button>
      </li>
    <li>
      <button type="button" onClick="window.location.assign('opis.php');">
        <span class="glyphicon glyphicon-info-sign"></span> 0 stranici</button>
      </li>
    </ul>
  <ul id="list" class="nav navbar-nav navbar-right navbar-text">
    <li>
      <button type="button" onClick="eraser();" data-toggle="modal" data-target="#RegMod">
        <span class="glyphicon glyphicon-list-alt"></span> Registracija</button>
      </li>
    <li>
      <button type="button" onClick="erase();" data-toggle="modal" data-target="#LogMod">
        <span class="glyphicon glyphicon-log-in"></span> Prijava</button>
      </li>
    </ul>
</div>

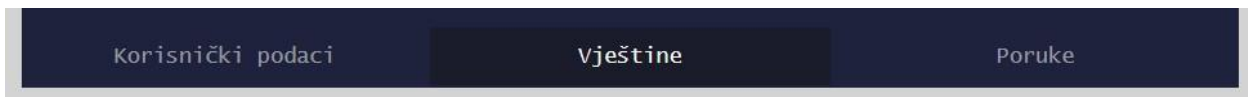
```

Sl. 4.4. HTML kod primarne navigacijske trake

4.3.2. Sekundarna navigacijska traka

Sekundarna navigacijska traka omogućava olakšano kretanje između elemenata na korisničkom profilu (slika 4.5.). Korisnički profil se sastoji od tri dijela:

- korisnički podaci,
- vještine,
- poruke.



Sl. 4.5. Izgled sekundarne navigacijske trake

4.4. Dinamička pohrana vještina

Dinamički sadržaj web aplikacije je sadržaj koji se mijenja ovisno o interakciji s korisnikom ili nekim drugim utjecajem. Razlika od statičnog je što za promjenu dinamičkog sadržaja osoba ne mora biti programer ili dizajner već samo krajnji korisnik koji koristi tu web stranicu. Takav sadržaj privlači više korisnika te omogućava ugodniju te lakšu radnu okolinu pri radu s web aplikacijom.

```
<div id="selection" class="col-md-8 letters" >
<?php
$username=$_SESSION['user'];
$sql="SELECT tolearn FROM oglas WHERE user='".$username."'";
$result = mysqli_query($con, $sql);
$row = mysqli_fetch_assoc($result);
if (empty($row['tolearn']))
{
    echo "Nema unosa";
}
else
{
    echo $row['tolearn'];
}
?>
</div>
</div>
<div class="col-md-4">
<div class="text-right">
<button id="editing" class="insertbutton" onClick="skillinsert();" type="button">
    <span id="glyph" class="glyphicon glyphicon-edit"></span>
</button>
```

Sl. 4.6. Izgled koda za ispis unesenih vještina

Prilikom prijave korisničkim računom, prikazat će se forma za unos vještina unutar koje će biti ispisana poruka „Nema unosa“. Na slici 4.6. može se vidjeti dohvaćanje pohranjene vještine iz baze podataka pomoću PHP-a. Ako prethodno nije unesena ni jedna vještina, pritiskom na tipku oznake „editing“ pokrenut će se funkcija „skillinsert()“ za unos vještina. Izgled već unesene vještine te izgled izvršenog koda može se vidjeti na slici 4.7.



Sl. 4.7. Izgled forme za unos vještina


```

function skillinsert()
{
    document.getElementById("selection").innerHTML = "";
    var here = document.getElementById("selection");
    var array = ["Računalne znanosti", "Informatika: Software"];
    var selectList = document.createElement("select");
    selectList.id = "s_wish";
    here.appendChild(selectList);
    for (var j = 0; j < array.length; j++)
    {
        var option = document.createElement("option");
        option.value = array[j];
        option.text = array[j];
        selectList.appendChild(option);
    }
    $("select").css('background-color', 'lightgrey');
    $("select").css('color', '#1e223d');
    $("select").css('border-radius', '5px');
    $("select").css('padding', '1%');
    $("select").addClass("form-control");
    $("#glyph").removeClass("glyphicon-edit");
    $("#glyph").addClass("glyphicon-check");
    document.getElementById('editing').onclick = skillsave;
}

```

Sl. 4.8. Izgled koda koji mijenja formu za unos vještina

Nakon interakcije korisnika sa formom za unos, poziva se JavaScript funkcija koja dinamički stvara popis svih vještina koje je moguće odabrati (slika 4.8.). Funkcija započinje brisanjem sadržaja unutar elementa pod oznakom „selection“. Nakon toga u varijable pohranjuje oznake elementa u kojem treba stvoriti popis, sadržaj popisa te stvaranje elementa odabira. Naredbom „appendChild“ dodjeljuje se stvoreni odabir elementu unutar kojeg će biti stvoren. Unutar „for“ petlje za opcije odabira postavljaju se prethodno spremljene vještine. Pomoću jQuery-a mijenjaju se parametri (pozadinska boja, boja teksta, itd.) novo stvorenog elementa za odabir. Izmijenjen je i izgled tipke za uređivanje vještina te se na kraju postavlja da tipka izvršava novu funkciju „skillsave()“.

Sl. 4.9. Izgled forme za unos vještine nakon izvršene funkcije „skillinsert()“

```

function skillsave()
{
    document.getElementById('editing').onclick = skillinsert;
    $("#glyph").removeClass("glyphicon-check");
    $("#glyph").addClass("glyphicon-edit");
    var zelja = $('select').val();
    $.post("saving.php",
    {
        zelja,
    },
    function(skill)
    {
        if (skill=="Pohrana neuspješna!" ) {alert(skill);}
        else
        {
            document.getElementById('selection').innerHTML = skill;
        }
    });
}

```

Sl. 4.10. Izgled funkcije za slanje vještine i izmjenu izgleda tipke za pohranu

Na slici 4.10. prikazana je funkcija „skillsave()“ koja pohranjuje odabranu vještinu u bazu podataka. Funkcija vraća da se izvrši prethodna funkcija „skillinsert()“, ako je tipka „editing“ ponovo pritisnuta, što će ponovo stvoriti odabir s popisom vještina. Nakon toga, mijenja se izgled tipke „editing“ te se uzima vrijednost opcije koja je odabrana iz popisa. Vrijednost pohranjena u varijablu „zelja“ šalje se jQuery metodom „post“ na vanjski PHP dokument. Ovisno o povratnim podacima, funkcija unutar „post“ metode prikazat će te podatke.

```

<?php
include "spoj.php";
session_start();

$stolearn=$_POST['zelja'];
$username=$_SESSION['user'];

$sql="UPDATE oglas SET tolearn='$stolearn' WHERE user='$username'";
if (!mysqli_query($con,$sql))
{
    die('Error: ' .mysql_error());
}
echo $stolearn;
?>

```

Sl. 4.11. Izgled PHP koda za pohranu vještine

Na slici 4.11. vidi se spremanje podataka primljenih „post“ metodom u varijablu, a iz globalne se varijable podaci o korisniku spremaju u lokalnu kako bi daljnji kod bio uredniji. Nakon toga, zamjenjuju se podaci unutar baze podataka koji stoje uz korisničko ime sa novo unesenim. Naposljetku, naredbom za ispis „echo“ vraćaju se podaci funkciji „skillsave()“ koja ih ispisuje kao što je prikazano na slici 4.9.

4.5. Sustav pretrage i kontaktiranja korisnika

Nakon unosa vještina, sljedeći korak prema razmjeni vještina je pretraga baze podataka za ostalim korisnicima koji odgovaraju vještinama koje je logirani korisnik upravo unio. Sustav pretrage izvršava se na način da traži vještine koje korisnik želi naučiti u odjeljku s vještinama koje su drugi korisnici već svladali. Iz popisa tih korisnika traži korisnike čije vještine, koje žele naučiti, odgovaraju vještini koju je logirani korisnik unio kao naučenu. Nakon pretrage, sustav prikazuje korisnike koji odgovaraju vještinama koje je korisnik unio, ako takvi korisnici ne postoje, sustav će ispisati rezultate koji najbliže odgovaraju unesenim vještinama.

```
$("#search").click(function()
{
    $.post("searchscript.php",
    {},
    function(res)
    {
        document.getElementById("results").innerHTML = res;
    });
});
```

Sl. 4.12. *jQuery funkcija za pokretanje pretrage*

Pretraga se pokreće pritiskom na tipku „Traži“ koja nosi oznaku „search“. Ona poziva jQuery funkciju koja „post“ metodom poziva vanjski PHP dokument za pretraživanje baze podataka. Povratne podatke ispisuje unutar prostora za rezultate pretrage (slika 4.12.).

```
$username=$_SESSION['user'];

$sql = "SELECT * FROM oglas WHERE user='".$username."'";
$resusr=mysqli_query($con,$sql);

while($r_usr = mysqli_fetch_array($resusr))
{
    $sqls = "SELECT * FROM oglas WHERE known='".$r_usr['tolearn']."' and tolearn='".$r_usr['known']."'";
    $resadd=mysqli_query($con,$sqls);

    if (mysqli_num_rows($resadd)!=0)
    {
        while($r_add = mysqli_fetch_array($resadd))
        {
```

Sl. 4.13. *Programski kod koji izvršava pretragu*

Na slici 4.13. može se vidjeti kako pomoću PHP-a i SQL-a skripta pretražuje bazu podataka za korisnicima koji odgovaraju kriterijima pretrage. Na početku dokumenta, pretražuje se tablica „oglas“ u kojoj su spremljene sve vještine i imena korisnika koji su ih unijeli. Prvo se dohvaćaju vještine koje je korisnik unio i koje će se koristiti kao filter pretrage vještina ostalih korisnika. Dok god korisnik postoji u bazi podataka, njegove će vještine biti korištene za pretragu vještina

ostalih korisnika. Ako su rezultati pronađeni, pokrenut će se još jedna „while“ petlja koja će proći kroz sve rezultate, dok god ih ima, i ispisati ih (slika. 4.14.)

```
echo "<div class='row adds'>";
echo "<div class='text-center'><div class='col-md-3'>".$r_add['user'].</div></div>";
echo "<div class='col-md-4'>".$r_add['known'].</div>";
echo "<div class='col-md-4'>".$r_add['tolearn'].</div>";
echo "<div class='col-md-1'>
<button class='sbutton'
onClick='tosetid(this.id);' data-toggle='modal' data-target='#".$r_add['user']."' id='".$r_add['user']."'>
Odaberi
</button>
</div>";
echo "</div>";
echo "<br>";
```

Sl. 4.14. Ispis rezultata pokrenute pretrage

Kao što je prikazano na slici 4.14., rezultat pretrage bit će ispisano kao korisničko ime, naučena vještina i željena vještina. Uz te podatke stvorit će se i tipka koja će omogućavati kontaktiranje svakog pronađenog korisnika. Stvorena tipka izvršavat će funkciju „tosetid()“ koja pamti i oznaku tipke. Oznaka tipke postavljena je kao korisničko ime uz koje je tipka stvorena, tj. korisnik će imati vlastitu tipku sa vlastitom oznakom. Ta će oznaka kasnije biti korištena kako bi se kontaktirao korisnik kraj čijeg imena je tipka pritisnuta.

```
$sqlss ="SELECT * FROM oglas WHERE tolearn='".$rowusr['known']."'";
$resnomatch=mysqli_query($con,$sqlss);
```

Sl. 4.15. Kriterij pretrage ako prva pretraga ne pronađe rezultate

Nastavak skripte za pretragu baze podataka, ako prilikom prve pretrage nisu pronađeni rezultati, nastavlja se identičnim ispisom i „while“ petljom uz izmijenjen filter pretrage. Pretraga se izvršava na način da se uzimaju samo vještine koje logirani korisnik želi naučiti, te se baza podataka pretražuje po korisnicima koji su te vještine naučili i odlučili podijeliti s drugima (slika 4.15.).

```
function tosetid(choosebt)
{
    $('fromfunc').attr('id',choosebt);
    $.post("saveid.php",
    {
        choosebt,
    },
    function(name)
    {
        document.getElementById("recname").innerHTML = name;
    });
}
```

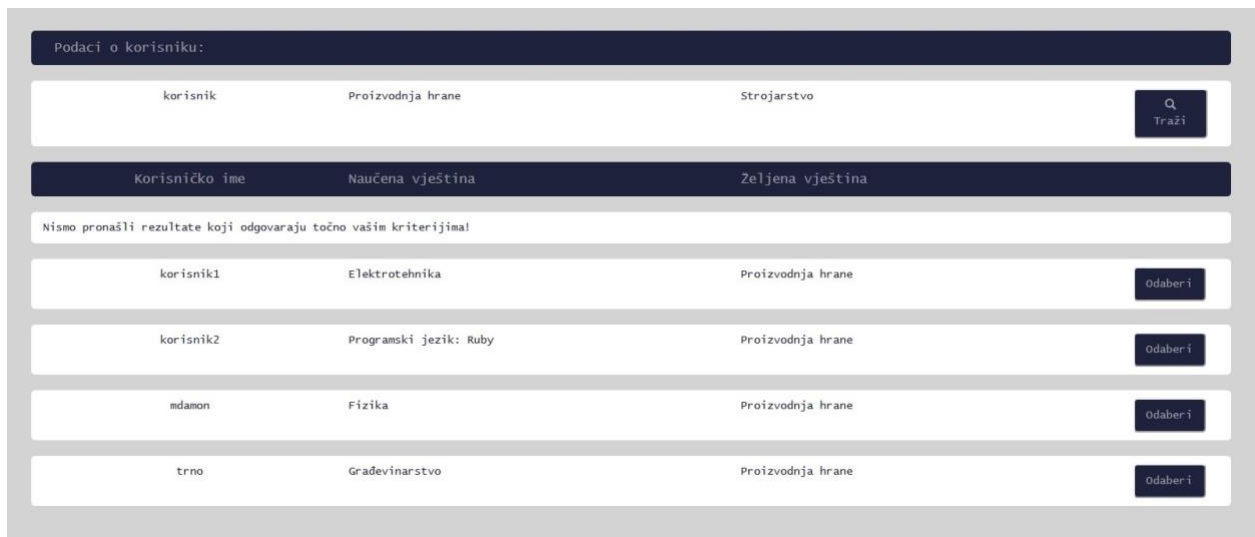
Sl. 4.16. JavaScript funkcija sa jQuery metodom „post“

Na slici 4.16. prikazana je funkcija „toseid()“ koja se poziva pritiskom na tipku kraj ispisanih rezultata te korisničkih imena. Oznaka stvorene tipke pohranjena je u varijablu imena „choosebt“. Funkcija koristeći jQuery naredbe mijenja oznaku elementa s klasom „fromfunc“, te metodom „post“ šalje oznaku tipke na vanjski PHP dokument „saveid“.

```
session_start();
$user = $_POST['choosebt'];
$_SESSION['userr']=$user;
echo $user;
```

Sl. 4.17. PHP dokument imena „saveid“

Na slici 4.17. vidljivo je kako se primljena varijabla pohranjuje u PHP varijablu „user“ te se nakon toga pohranjuje kao globalna varijabla „user“ što označava ime korisnika pronađenog pretragom i oznaku pripadajuće tipke. Nadalje, naredbom „echo“ vraća tu vrijednost funkciji.



Sl. 4.18. Izgled izvršene pretrage

Slika 4.18. prikazuje izgled rezultata pronađenih koristeći sustav za pretragu. Svaki od rezultata je prikazan kao jedan red koji sadrži korisničko ime, naučenu i željenu vještinu te tipku za kontaktiranje pronađenog korisnika.

```

<div class="modal fade fromfunc" id="test" role="dialog">
  <div class="modal-dialog modal-md">
    <div class="loginbox modal-content text-center">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Kontaktirajte korisnika: <p id="recname"></p></h4>
      </div>
      <div class="modal-body">
        <br>
        <textarea id="contactbox" class="txt form-control" rows="10" cols="25" placeholder="Poruka"></textarea>
      </div>
      <div class="modal-footer">
        <button id="tosend" class="sbutton" type="button">Pošalji</button>
      </div>
      <script>
        $("#tosend").click(function()
        {
          var message = $('#contactbox').val();
          $.post("contact.php",
            {
              message,
            },
            function(there)
            {
              alert(there);
            });
          $('#contactbox').val('');
        });
      </script>
    </div>
  </div>
</div>
</div>

```

Sl. 4.19. Programski kod prozora za kontaktiranje odabranog korisnika

Element sa slike 4.16. s klasom „fromfunc“ je prozor za unos podataka uređen koristeći Bootstrap. Na slici 4.19. može se vidjeti programski kod prozora za podatke (eng. *modal*). S obzirom da je oznaka prozora za podatke ime korisnika, pritiskom na tipku odabira kraj korisničkog imena kreirat će se prozor za unos podataka koji pripada tom korisniku (slika 4.20.). Pritiskom na tipku slanja s oznakom „tosend“ poziva se jQuery funkcija koja metodom „post“ šalje poruku korisniku koji je odabran.



Sl. 4.20. Izgled prozora za kontakiranje odabranog korisnika

```

$msg=$ _POST['message'];
$usr=$_SESSION['user'];
$rec=$_SESSION['userrr'];
if (empty($msg))
{
    echo "Unesi poruku";
}
else
{
    $sql="SELECT * FROM korisnici WHERE user='".$rec."'";
    $res = mysqli_query($con,$sql);
    if (mysqli_num_rows($res)==1)
    {
        $sqlinsert="INSERT INTO msg (sender,reciever,message)
VALUES ('$ _SESSION[user]','$ _SESSION[userrr]','$ _POST[message] ')";
        if (!mysqli_query($con,$sqlinsert))
        {
            die('Error: ' .mysqli_error());
        }
        echo "Poruka poslana";
    }
    else
    {
        echo "Poruka nije poslana";
    }
}
}

```

Sl. 4.21. Programski kod za slanje poruke odabranom korisniku

Slika 4.21. prikazuje skriptu za pohranu poruka u bazu podataka pomoću PHP-a i SQL-a. Primljena poruka sprema se u varijablu te se iz globalnih varijabli dohvaćaju pošiljatelj i primatelj poruke, ovisno o tome koji je korisnik odabran. Provjerava se i je li išta upisano u polje za unos poruke te postoji li korisnik kojemu se poruka šalje. Nadalje, u tablicu „msg“ unose se podaci o pošiljatelju, primatelju i poruka koja se šalje. Prije završetka skripte, provjerava se postoje li greške prilikom pohrane podataka. Naredbama „echo“ šalju se povratne informacije o statusu poruke.

4.6. Uređivanje izgleda web aplikacije za razmjenu usluga

Nakon podešavanja skripti, funkcija i ostalog interaktivnog sadržaja, potrebno je urediti izgled i način na koji se pojavljuju određeni elementi web aplikacije. U nastavku biti će prikazan izgled web aplikacije koji je uređen koristeći CSS.

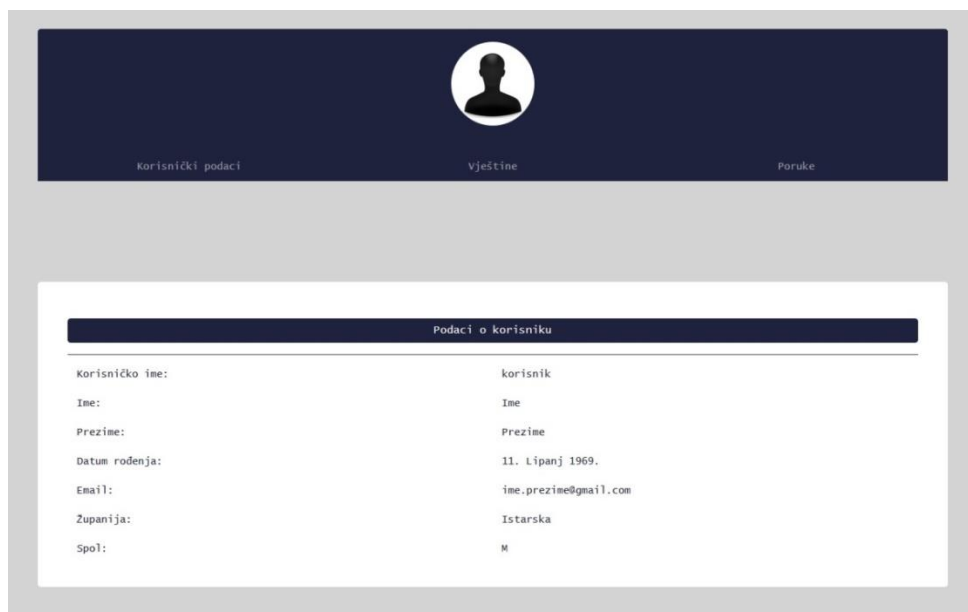


Sl. 4.23. Naslovnica web aplikacije.

Sl. 4.24. Registracija na web aplikaciju.

Sl. 4.25. Prijava na web aplikaciju.

Nakon registracije (Sl. 4.24.) te prijave (Sl. 4.25.), korisnik će biti preusmjeren na stranicu svog korisničkog računa (Sl. 4.26.).



Sl. 4.26. *Korisnički profil na web aplikaciji.*

Na slici 4.26. vidljive su opcije koje su ponuđene na korisničkom računu. Na vrhu stranice korisničkog računa nalazi se sekundarna navigacijska traka koja olakšava navigaciju kroz stranicu korisničkog računa. Omogućava korisniku pregled svojih osobnih podataka i izmjenu u bilo koje vrijeme, odnosno omogućava dodavanje novih vještina (Sl. 4.27.). Korisniku se prikazuje i sustav poruka koji omogućava kontaktiranje svih registriranih korisnika (Sl. 4.28.).



Sl. 4.27. *Unos vještina.*



Sl. 4.28. *Sustav kontaktiranja ostalih korisnika.*

4.7. Vrednovanje programskog rješenja

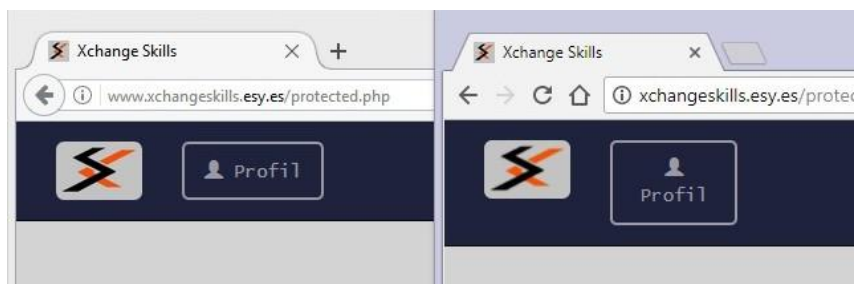
Programsko rješenje većinom ispunjava zahtjeve, navedene u poglavlju 2, uz neke preinake. Prilikom izrade forme za unos vještina bilo je potrebno stvoriti polje unutar kojeg će korisnici upisati vještine (potrebne i željene). Zbog problema s korištenjem tih vještina, kao kriterija za pretragu, bilo je potrebno zamijeniti polje za unos s već generiranim popisom općenitih vještina. Zbog broja narječja i velikog broja sinonima unutar hrvatskoga jezika, bilo je potrebno stvoriti generalne oblike naziva vještina i stvoriti njihov popis. Korisnici odabiru generalni oblik koji najbliže opisuje njihovu vještinu te se prilikom kontaktiranja korisnika vještine detaljno opisuju.

Izgled programskog rješenja dosljedan je kroz sve stranice ove aplikacije za razmjenu usluga. Font i veličina teksta mijenja se ovisno o dijelu stranice na kojoj se korisnik nalazi, no on ostaje čitljiv i jasan kroz cijelu aplikaciju. Web aplikacija za razmjenu usluga stvorena je kako bi se njezin sadržaj dosljedno prikazao bez obzira na veličinu zaslona na kojem se prikazuje

Vrednovanje web aplikacije izvršavalo se na 3 najpoznatija web preglednika:

- Mozilla Firefox,
- Google Chrome,
- Opera.

Prikaz određenih elemenata web aplikacije, zbog različitih postavki unutar web preglednika, različito se prikazuje. Dok je na Mozilli Firefox prikaz poveznica unutar navigacijske trake prikazan u jednom redu, Opera i Google Chrome stvaraju posebno red za ikonu te posebno za naziv poveznice. Razlika u prikazivanju poveznica između Mozille Firefox (lijevo) i Google Chromea (desno) prikazana je na slici 4.29.



Sl. 4.29. Razlika u prikazu poveznice unutar navigacijske trake.

Kako bi se uklonila razlika u prikazu poveznica na različitim web preglednicima, potreban je unos programskog koda u obliku komentara za svaki podržani web preglednik. Kao što je prikazano na slici 4.30., programski kod se izvršava samo ako je pokrenut web preglednik Internet Explorer.

```
<!--[if lt IE 9]>
  <script src="/js/html5shiv.js"></script>
<![endif]-->
```

Sl. 4.30. Primjer programskog koda pisanog u obliku komentara.

Prilikom prikaza tipke za odabir korisnika pri ispisu rezultata pretrage, na zaslonima razlučivosti manjoj od 1366x768 tipka prelazi preko ruba odjeljka u kojem se nalazi. Razlog leži u činjenici što je tipka stvorena dinamički od strane vanjskog PHP dokumenta te svojstvo responzivnosti, koje tipka nasljeđuje od odjeljka u kojem se nalazi, nije pravilno preneseno. Izgled tipke na rezolucijama manjim od prethodno spomenute vidljiv je na slici 4.30.



Sl. 4.31. Prijelaz tipke preko odjeljka rezultata.

Kako bi se spriječio prijelaz tijela tipke preko odjeljka za rezultate potrebno je pristupiti pretpostavljenim klasama Bootstrapa korištenima pri uređivanju prikaza tog odjeljka i upisati parametre koji će tijelu tipke vratiti svojstvo responzivnosti. Na taj bi se način ograničilo njezino pozicioniranje unutar odjeljka za rezultate.

5. ZAKLJUČAK

Tema ovoga rada bila je izraditi web aplikaciju za razmjenu usluga koristeći osnovne web tehnologije za izradu web sadržaja. Aplikacije za razmjenu usluga su web aplikacije koje omogućavaju svojim korisnicima međusobnu razmjenu materijalnih dobara ili vještina po principu razmjene „usluga za uslugu“. Korisnici, kojima je potrebna određena vještina, unose ono što oni mogu ponuditi u zamjenu za ono što im je potrebno. Kako bi to omogućili, stvorena aplikacija ima mogućnost registracije korisnika, unos vještina, pretragu između ostalih korisnika ovisno o unesenim vještinama te kontaktiranje odabranih korisnika. Prije same izrade bilo je potrebno osmisliti koncept i raspored elemenata unutar web stranice koji će kasnije biti stvoren, bio on statičan ili dinamičan. Unutar sustava pretrage web aplikacije za razmjenu usluga implementiran je sustav kontaktiranja korisnika koji zajedno omogućavaju brz i jednostavan pristup ostvarivanju kontakta s odabranom osobom. Kako bi korisnik mogao koristiti web aplikaciju za razmjenu usluga, uređen je njezin izgled i prikaz sadržaja. Web aplikacija za razmjenu usluga korisna je aplikacija koja će omogućiti svojim korisnicima, bez potrebe za znanjem stranog jezika, pristup učenju novih vještina. Aplikacija uvijek može biti poboljšana te će je autor kroz vrijeme razvijati kako bi što bolje odgovarala potrebama korisnika.

SAŽETAK

Kako bi se građanima Republike Hrvatske omogućio novi način učenja novih vještina, izrađena je web aplikacija za razmjenu vještina na principu razmjene usluga pod nazivom „Xchange Skills“. Princip razmjene usluga omogućava korisnicima web aplikacije razmjenu njihovih imovina ili vještina za imovinu ili vještine ostalih korisnika iste aplikacije. Web aplikacija „Xchange Skills“ omogućava korisnicima unos znanih i željnih vještina. Pohranjene vještine sustav za pretragu koristi pri pretraživanju baze korisnika za odgovarajućim korisnicima pogodnima za razmjenu vještina s prijavljenim korisnikom. Nakon odabira jednog od pronađenih korisnika, detaljnije se dogovara vrijeme i mjesto razmjene vještina putem sustava za kontaktiranje. Za izradu web aplikacije za razmjenu usluga koristile su se osnovne web tehnologije za izradu web sadržaja – HTML, CSS, JavaScript, PHP te SQL. Izrađena web aplikacija ima mogućnost registracije i prijave korisnika, unosa vještina u stvarnom vremenu, pretragu za ostalim registriranim korisnicima unutar baze podataka korisnika te mogućnost kontaktiranja odabranih korisnika kako bi se dogovorila potencijalna razmjena vještina.

Ključne riječi: web aplikacija, princip razmjene usluga, razmjena vještina, web tehnologije

ABSTRACT

To introduce a new way of learning new skills to citizens of the Republic of Croatia a web application was developed under the name of “Xchange Skills” which is based on the principle of trading favors. Principle of trading favors enables users of a specific application, which uses that principle, to trade skills or their property for skills or property of another user using the same application. Web application “Xchange Skills” allows its users to enter and save skills they already learned and the ones they want to learn. Saved skills are later used by the search system which finds other users that are most compatible to the logged user. After selecting one of the recently found users, the time and place of a skill exchange can be arranged using the system for contacting the desired user. To develop skill exchange web application basic web technologies for creating web content were used, such as – HTML, CSS, JavaScript, PHP and SQL. The “Xchange Skills” application enables its users to register and login, to enter and edit saved skills in real time, to search for other registered users and to contact them for further arrangement of potential skill exchange.

Key words: web application, principle of trading favors, skills exchange, web technologies.

ŽIVOTOPIS

Ivan Petak rođen je 6. veljače 1995. u Bjelovaru, Republika Hrvatska. Godine 2013. upisuje stručni studij smjera Informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Godine 2009. po završetku Osnovne škole Berek u Bereku, upisuje srednju Tehničku školu u Bjelovaru, smjer tehničar za računalstvo. Iskusi je korisnik u razumijevanju i govoru engleskog jezika te spreman na svako moguće usavršavanje navedenog ili učenje nekog drugog stranog jezika. Sklon timskom kao i individualnom radu. Komunikativan, snalažljiv i ažuran u rješavanju svih svojih obveza.

Potpis:

LITERATURA

- [1] Shakespeare, Adventures in the skill trade [online], Daily Mail, Ujedinjeno Kraljevstvo, 2010., dostupno na: <http://www.dailymail.co.uk/home/you/article-1267532/Adventures-skill-trade--Nearly-new-Jimmy-Choos-exchange-legal-advice-anyone.html#ixzz0r2l6qFSz>,
[pristup ostvaren 12.07.2017.]
- [2] C. Flynn, 14 Technologies Every Web Developer Should Be Able to Explain [online], Differential, Sjedinjene Američke Države, 2015., dostupno na: [\(https://differential.com/insights/14-technologies-every-web-developer-should-be-able-to-explain/\)](https://differential.com/insights/14-technologies-every-web-developer-should-be-able-to-explain/),
[pristup ostvaren 27.05.2017.]
- [3] D. Connolly, A Little History of the World Wide Web [online], The World Wide Web Consortium, 2000., dostupno na: <https://www.w3.org/History.html>,
[pristup ostvaren 01.06.2017.]
- [4] J.C. Jackson, Web technologies : a computer science perspective, Pearson Education, Inc., Upper Saddle River, New Jersey, 2007.
- [5] C. Heng, What is MySQL? What is a Database? What is SQL? [online], The Site Wizard, 2010., dostupno na: <https://www.thesitewizard.com/faqs/what-is-mysql-database.shtml>,
[pristup ostvaren 01.06.2017.]
- [6] Wikipedia, Bootstrap (front-end framework), [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)),
[pristup ostvaren 12.06.2017.]