

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2015./ 2016.

Renata Waller

**Primjer informacijskog sustava u mrežnom okruženju:
upisani/diplomirani studenti informacijskih i komunikacijskih znanosti
Filozofskog fakulteta u Zagrebu na interaktivnoj karti**

Završni rad

Mentor: dr.sc. Kristina Kocijan, doc.

Zagreb, 2016.

Sadržaj

1. Uvod.....	3
2. Osnovni pojmovi.....	4
2.1. Informacijski sustav	4
2.2. Web 2.0 okruženje.....	4
2.3. HTML.....	5
2.4. CSS.....	6
2.5. JavaScript	6
2.6. Baze podataka	7
2.7. SQL i MySQL	7
2.8. PHP.....	7
2.9. Google Maps API.....	8
3. Priprema za izradu web sjedišta.....	9
3.1. Ideja i svrha projekta	9
3.2. Planiranje izrade web sjedišta	10
3.3. Informacijska shema i struktura web sjedišta	10
4. Projektiranje i realizacija web sjedišta.....	12
4.1. Google Maps i PHP/MySQL	12
4.2. HTML struktura web sjedišta.....	19
4.3. CSS struktura web sjedišta.....	21
5. Zaključak	24
6. Literatura.....	26

**Primjer informacijskog sustava u mrežnom okruženju:
upisani/diplomirani studenti informacijskih i komunikacijskih znanosti
Filozofskog fakulteta u Zagrebu na interaktivnoj karti**

Sažetak

U ovom radu opisan je proces kreiranja informacijskog sustava u obliku web sjedišta od početne ideje do završnog proizvoda i planova za budućnost. Sadržaj rada obuhvaća faze razvoja web sjedišta od nastanka ideje i svrhe projekta, planiranja izrade web sjedišta, informacijske sheme i strukture, izrade interaktivne karte uz pomoć Google Maps API-ja i njeno programiranje u klijentskim i poslužiteljskim skriptnim jezicima, modeliranje baze podataka, te kodiranja stranica u HTML i CSS kodu.

Ključne riječi: informacijski sustav, HTML, CSS, PHP, JavaScript, SQL, Google Maps API, interaktivna karta

**An example of an information system in a network environment:
interactive map of enrolled / graduates of information and communication
sciences, Faculty of humanities and social sciences in Zagreb**

Abstract

This paper describes the process of creating an information system in the form of web sites from the initial idea to the final product and plans for the future. The content of paper includes development phase of web sites from the idea and purpose of the project, the planning of making web sites, information schemes and structures, making interactive maps using Google Maps API and its programming in the client and server scripting languages, database modeling and coding page in HTML and CSS code.

Key words: information system, HTML, CSS, PHP, JavaScript, SQL, Google Maps API, interactive map

1. Uvod

U ovom radu opisat ću postupak nastajanja i dizajniranja informacijskog sustava u mrežnom okruženju. Cilj ovog sustava je vizualni prikaz broja studenata koji su upisali, odnosno završili, studij informacijskih i komunikacijskih znanosti na Filozofskom fakultetu u Zagrebu.

Rad je temeljen na konkretnom primjeru mrežne stranice na adresi <http://ikzstudentskiprojekti.ffzg.hr/StudentiFFZG/index.html>, koja na interaktivan način na karti Hrvatske prikazuje brojčane podatke o upisanim i diplomiranim studentima od uvođenja bolonjskog procesa.

U drugom dijelu rada objasniti ću neke osnovne pojmove korištene za izradu ovog web sjedišta. Također ću opisati kako funkcionira projektiranje i realizacija web sjedišta u što spada planiranje izrade i informacijska struktura web sjedišta. U trećem dijelu rada navest ću ideju i svrhu ovog projekta, objasniti zašto sam se odlučila za izradu ovakvog projekta i kome bi on mogao pomoći. U završnom dijelu rada objasniti ću faze nastanka ovog web sjedišta počevši od ideje do završnog proizvoda što obuhvaća kodiranje stranice u HTML i CSS kodu, programiranje u klijentskim i poslužiteljskim skriptnim jezicima i modeliranje baze podataka. Veći naglasak staviti ću na mrežne tehnologije, programske jezike, i metode korištene u izradi ovog sustava.

Za izradu ovakvog projekta odlučila sam se zbog drugačijeg pristupa pisanju rada, jer uz sam rad bilo je potrebno izraditi i web sjedište. Velikim dijelom, za ovaj projekt sam se odlučila radi izrade interaktivne karte pomoću Google Maps API-ja, pošto do sada nisam imala prilike raditi s njima. Uz sve prethodno znanje o kodiranju stranica u HTML i CSS kodu, programiranja u klijentskim i poslužiteljskim skriptnim jezicima i modeliranju baza podataka, ovim projektom naučila sam se služiti Google Maps JavaScript API-jem i dodatno proširila prethodno znanje.

2. Osnovni pojmovi

U ovom poglavlju ću dati više informacija o pojmovima koje sam koristila u izradi ovog rada. Uz njihove definicije, navest ću i zašto mi je određena tehnologija bila potrebna i gdje sam ju koristila.

2.1. Informacijski sustav

„Informacijski je sustav mreža informacijskih djelatnosti, službi i odnosa; kao takva, ta je mreža inkorporirana u sva društvena područja i djelatnosti i tvori društvenu infrastrukturu. Kao nešto različito od drugih društvenih djelatnosti, podsustava ili sustava, informacijski je sustav to tek po svojim funkcijama, organizaciji i korisnicima.“ (Tuđman et al, 1991).

Informacijski sustav je sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje potrebne informacije tako da su one dostupne svim članovima neke organizacije. (Radovan, 1991). Tako i u ovom projektu, služi za pohranjivanje podataka o mjestima rođenja, godinama upisa ili završetka studija i broju studenata u bazu podataka i isporučuje ih na zahtjev korisnika ovog web sjedišta u obliku grafičkog prikaza na karti.

2.2. Web 2.0 okruženje

Sam pojam Web 2.0 postao je značajan nakon konferencije O'Reilly Media 2004. godine. O'Reilly (2005) Web 2.0 definira kao poslovnu revoluciju u kompjutorskoj industriji uzrokovanu tretiranjem mreže kao platforme i nastojanje da se shvate pravila uspjeha na toj novoj platformi. Također, provodi i hijerarhijsku podjelu aplikacija, među kojima se nalaze i aplikacije koje jednako funkcioniraju i online i offline, a primjer jedne od tih aplikacija je Google Maps koji je korišten u ovom projektu za izradu interaktivne karte.

Kao i mnogi važni koncepti, za Web 2.0 bismo mogli reći da nema čvrste granice nego više neku gravitacijsku jezgru. Web 2.0 možemo zamisliti kao skup načela i postupaka koji povezuju stvarni sustav web lokacija (internetskih stranica i aplikacija) koje pokazuju neke ili svako od tih načela, na različitim udaljenostima od te jezgre.

Ono što karakterizira Web 2.0 je da se web gleda kao platformu, što znači da korisnici aplikacije mogu koristiti u potpunosti kroz web preglednik, korisnici su ti kojima podaci na nekoj stranici pripadaju i koji nad njima imaju kontrolu, dok i sama arhitektura weba potiče korisnike da i oni, tijekom korištenja, doprinesu nekom web sadržaju ili aplikaciji, za razliku od jake hijerarhije i kontrole pristupa tipične za aplikacije u kojima sustavi kategoriziraju korisnike na različite razine pristupa i funkcionalnosti (Carnet, 2009).

2.3.HTML

HyperText Markup Language (HTML) skup je oznaka koji se koristi za grafički opis izgleda stranica na World Wide Web-u. HTML dokument se sastoji od 2 dijela, od teksta i HTML oznaka. Oznake se odnose na opis izgleda pojedinih dijelova dokumenta (poglavlja, naslova, popisa, citata, slova itd.). Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekstualnih dokument tako da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik, njime ne možemo izvršiti nikakvu zadaću, on nam služi samo za opis naših hipertekstualnih dokumenata. HTML je jednostavan za upotrebu i lako se uči, zbog čega je opće prihvaćen i popularan (Henick, 2010).

Svaki HTML dokument sastoji se od osnovnih HTML elemenata, a HTML element sastoji se od para HTML oznaka (eng. *tag*). Također, svaki element može imati i attribute kojim se definiraju svojstva tog elementa. U ovom projektu korišten je HTML5 koji na početku dokumenta sadrži `<!DOCTYPE html>` element. Prijašnje verzije HTML-a imale su samo `<!DOCTYPE>` kojim se označavao DTD (eng. *Document Type Declaration*), čime se definira točna inačica standarda koja se koristi za izradu HTML dokumenta. Uz to, HTML5 zahtijeva i kodiranje znakova koje preglednicima govori koji set znakova treba koristiti pri učitavanju web stranice, a definiramo ga pomoću `<meta charset="UTF-8">` elementa. Za izradu ovog projekta korišteni su i elementi `<header>`, `<section>` i `<nav>`, također uvedeni verzijom HTML5, a svaki od njih objasnit ću u poglavlju 4.2 (Schmitt, Simpson, 2011).

2.4.CSS

Cascading Style Sheets (CSS) opisuje kako će HTML elementi biti prikazani na ekranu, papiru ili nekom drugom mediju. HTML je bio zadužen za kompletnu strukturu, izgled i sadržaj web stranice, ali nakon HTML 4.0 verzije uvodi se CSS koji se koristi za stil stranice dok je HTML ostao u funkciji definiranja strukture i sadržaja. CSS kao ni HTML, nije programski jezik. CSS sintaksa sastoji se od opisa izgleda elemenata u dokumentu. Opis može definirati izgled više elemenata kao što i više opisa može definirati jedan element. Na taj način se opisi slažu jedan preko drugog (poput kaskada) da bi definirali konačan izgled određenog elementa (W3Schools, 2016).

Za izradu ovog projekta, korištena je verzija CSS3 koja uvodi rad s manjim dijelovima koda s kojima je lakše i efikasnije raditi. Uz to, unutar CSS3 postoje i oznake koje omogućuju svojstvima da se prikazuju jednako u svima preglednicima, a u ovom projektu korištene su oznake `-o-`, `-moz-` i `-webkit-`. Osim toga, od prijašnje verzije uvedeno je i svojstvo `inline-block` koje znatno olakšava izradu izbornika, a objasniti ću ga detaljnije kasnije u radu, u poglavlju 4.3 (Crowther, 2013).

2.5.JavaScript

JavaScript je objektno zasnovan skriptni jezik. Cilj kreiranja JavaScript jezika je dodati interaktivnost HTML stranicama. JavaScript je interpreter, što znači da se skripta izvršava odmah naredbu po naredbu, bez prethodnog prevođenja cijelog programa i kreiranja izvršne datoteke. Po definiciji može biti unutarnji i vanjski. Unutarnji je napisan unutar HTML dokumenta, dok je vanjski napisan u posebnom dokumentu koji se zatim poziva na sličan način kao i CSS datoteka. JavaScript je ujedno i najpopularniji skriptni jezik na svijetu (Nixon, 2009).

Za izradu interaktivne karte, korišten je Google Maps JavaScript API koji omogućuje uređivanje i prikazivanje karte na web stranici. Nešto više o Google Maps JavaScript API-ju objasniti ću dalje u radu i prikazati na primjerima korištenim u izradi projekta.

2.6. Baze podataka

„Baza podataka je organizirani skup informacija ili podataka u kompjutorski čitljivom obliku. Općenito je pristup takvim bazama moguć putem različitih online servisa za pretraživanje koje nazivamo i online sustavima odnosno informacijskim sustavima. Takvi servisi za pretraživanje imaju kompjutore i programe (software) koji omogućuju vanjskim korisnicima da pretražuju jednu ili više baza podataka kako bi mogli pronaći informaciju ili podatak gdje se ta informacija (referenca) nalazi.“ (Tuđman et al, 1991).

U ovom slučaju, svi podatci potrebni za prikaz na karti smješteni su u bazu podataka nazvanom Karta. U toj bazi kreirana je tablica, izrađena pomoću SQL-a, a sadrži podatke o studentima potrebnim za pretraživanje i prikaz na karti. Izrada baze i tablice korištene za ovaj projekt objašnjena je u poglavlju 4.1.

2.7. SQL i MySQL

Structured Query Language (SQL) je strukturirani upitni jezik napravljen za rad s relacijskim bazama podataka (koje se sastoje od skupa povezanih tablica), a služi za pristup i manipuliranje velikim količinama podataka unutar baze podataka (Nixon, 2009). Da bi te podatke mogli prikazati na karti, potreban je SQL kojim je stvorena tablica s potrebnim podacima, a za upravljanje tim podacima potreban nam je MySQL.

Za razliku od SQL-a, MySQL je open source sustav za upravljanje bazom podataka (DBMS = Database Management System). On kontrolira rad s podacima i organizira ih kako bi im pristup bio brži i jednostavniji.

2.8. PHP

Personal Home Page (PHP) je skriptni jezik koji se izvodi na poslužiteljskoj strani (eng. *server-side*), a koristi se za kreiranje dinamičkih i interaktivnih web mjesta. Za PHP možemo još reći da je hipertekstualni pretprocesor za generiranje sadržaja web stranice u realnom vremenu. Besplatan je, otvorenog koda, a radi pod Windowsima i Unixom te ima objektno orijentiranu

sintaksu. Još jedna prednost PHP je što je kompatibilan sa skoro svim serverima koji se danas koriste (Apache, MySQL/MariaDB, Perl/PHP/Python) i podržava širok raspon baza podataka. PHP naredbe se mogu ugraditi u HTML kod stranice ili čuvati u posebnoj datoteci s ekstenzijom (W3Schools, 2016).

PHP je u izradi ovog projekta potreban za spajanje na MySQL server i postavljanje aktivne MySQL baze podataka, što ću detaljnije prikazati uz pomoć koda. Za spajanje na Apache, MySQL, PHP i phpMyAdmin server, korištena je multi-platforma XAMPP koja se može besplatno preuzeti na XAMPP službenim stranicama.

2.9. Google Maps API

Google Maps API (Application Programming Interface) je sučelje za programiranje primarno namijenjeno integraciji u web stranice i prilagodbi vlastitim potrebama. Dostupan je za Android, iOS, web preglednike i putem HTTP web servisa. Za širu upotrebu Google Maps tehnologije, Google je razvio posebno programsko korisničko okruženje, Google Maps API, koji mogu koristiti druge aplikacije. Korisničko okruženje zasnovano na XML-u (eng. *eXtensible Markup Language*) dodatno omogućava korisnicima da prilagode karte i da, uz pomoć položajnih podataka, kreiraju interaktivne karte za potrebe svojih aplikacija. Tako je i u ovom projektu korišten za izradu interaktivne karte čiju ću izradu detaljno opisati u poglavlju 4.1. Sve Google Maps JavaScript API aplikacije zahtijevaju autorizaciju, pa je tako potrebno zatražiti API ključ. Potrebno je registrirati se Google korisničkim računom i lozinkom te potom kreirati API ključ. Jednom kada se kreira ključ, isti se može koristiti u svim softverskim aplikacijama koje se implementiraju (Google Developers, 2016).

3. Priprema za izradu web sjedišta

U ovom poglavlju opisat ću proces kreiranja web sjedišta, te ideju i svrhu ovog projekta. Web mjesto (eng. *Web site*) ili web sjedište je mjesto na lokalnom ili udaljenom računalu (web poslužitelju) koje obuhvaća više povezanih web stranica. Svako web sjedište ima početnu stranicu koja opisuje osnovnu namjenu web sjedišta te sadrži poveznice prema ostalim stranicama web sjedišta.

3.1. Ideja i svrha projekta

Ideja ovog projekta bila je izraditi interaktivnu kartu koja sadrži podatke o tome koje godine je koliko studenata upisalo, završilo preddiplomski ili diplomiralo na Filozofskom fakultetu u Zagrebu. Trenutno, baza sadrži podatke od 2009. do 2016. godine, odnosno od uvođenja bolonjskog procesa do danas. Uz to, za sada je moguće vidjeti broj studenata samo s Odsjeka informacijskih i komunikacijskih znanosti, no pošto je projekt trenutno u testnoj fazi, u budućnosti planiram proširiti bazu, tako što ću dodati u izbornik i ostale odsjeke kao i izbor smjerova na diplomskom studiju. Jedan oblik vizualizacije upisanih studenata ima i studij informatike Prirodoslovno-matematičkog fakulteta na Sveučilištu u Splitu na svojim službenim stranicama. Za svaki smjer studija, moguće je vidjeti broj upisanih studenata kroz godine. Za razliku od ovog projekta, podatci su prikazani na grafovima.

Svrha ovog projekta bila bi pomoći budućim studentima pri odabiru željenog studija tako što bi mogli vidjeti na kojim smjerovima ima manje, odnosno više diplomiranih studenata što bi im moglo pomoći pri zapošljavanju. Isto tako, ovaj projekt mogao bi dobro doći i budućim poslodavcima koji će moći planirati zapošljavanje nakon što imaju podatke koliko studenata je te godine završilo neku studijsku grupu iz njihovog grada. Osim toga, projekt može poslužiti i za usporedbu upisnih kvota i broja diplomiranih studenata na određenim smjerovima, ali i usporedbu upisnih kvota ostalih fakulteta u Hrvatskoj i broja studenata koji upisuju te smjerove i završavaju ih.

3.2. Planiranje izrade web sjedišta

Za izradu web sjedišta potrebno je prvo izraditi plan izrade. Plan izrade treba sadržavati strukturu web sjedišta koja prikazuje hijerarhijski odnos između web stranica. Prvo nam je potrebna naslovna stranica koja uglavnom nosi naziv index.html i ona se nalazi na prvoj razini strukture. Na drugoj razini nalaze se stranice na koje se dolazi pomoću linkova na naslovnoj stranici. Za izradu web sjedišta također nam je potreban i kratak opis sadržaja svih stranice. Navigacija igra veliku ulogu tako da mora biti lako razumljiva i jednostavna za pronaći. Potrebno je koristiti lako čitljiv font i pogodnu boju teksta i pozadine (odabrati boju pozadine i boju teksta koje su dobar kontrast jedna drugoj). Nakon izrade plana i na kraju izrade i samog web sjedišta, potrebno ga je testirati i ispraviti greške. Bilo bi poželjno da se stranice prikazuju na isti način na svim web preglednicima kao i mobilnim preglednicima.

3.3. Informacijska shema i struktura web sjedišta

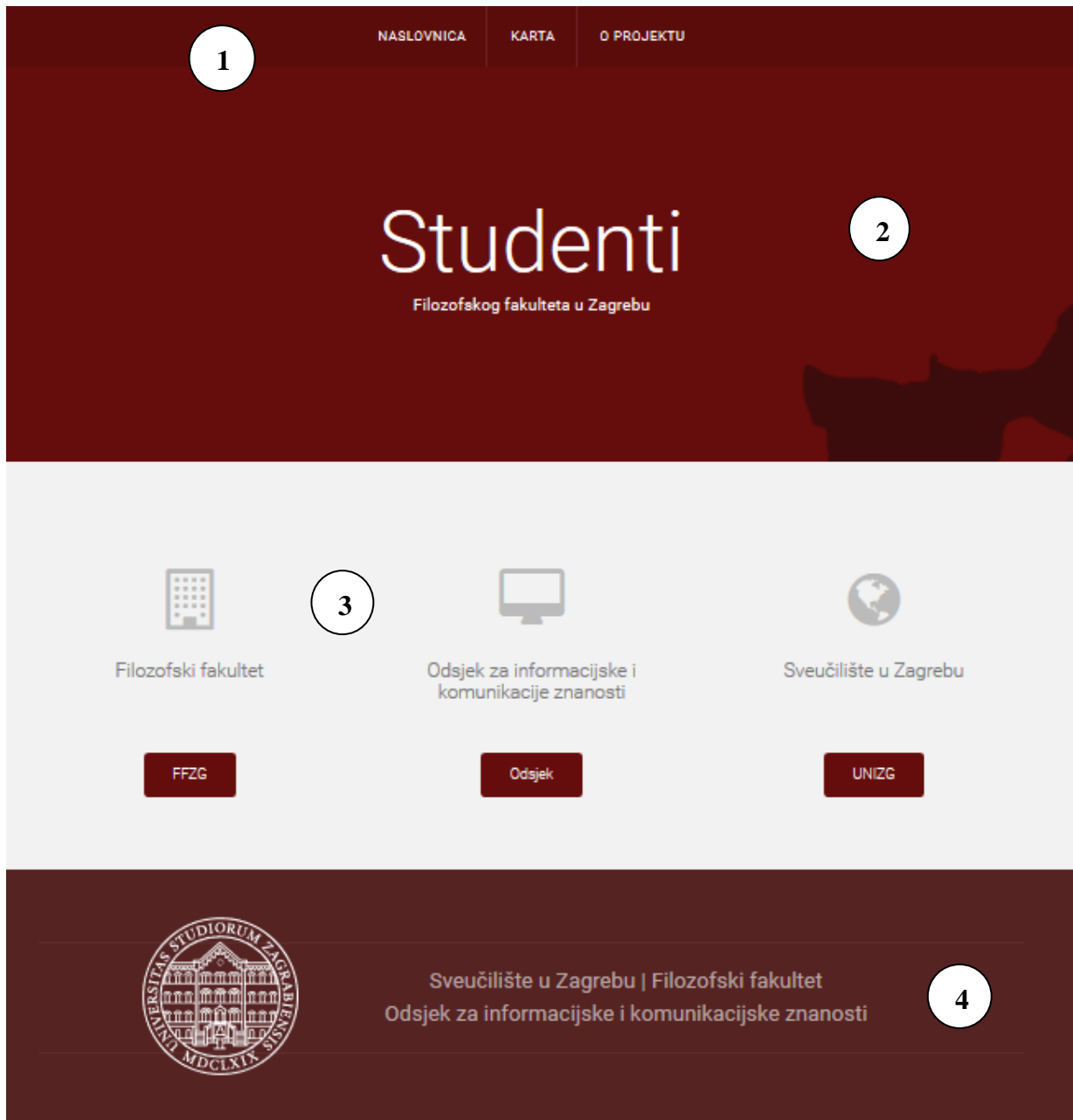
Prije same izgradnje web sjedišta, potrebno je odrediti od kojih će se komponenti ono sastojati. U okviru svakog informacijskog sustava razlikujemo komponente za prikupljanje podataka, obradu, memoriranje i dostavljanje krajnjim korisnicima.

Ovo web sjedište sastoji se od:

- index.html – naslovna stranica koja sadrži kratak opis projekta i poveznice na stranice fakulteta
- karta.html – stranica koja sadrži interaktivnu kartu i polja za odabir pretrage
- oprojektu.html – stranica koja sadrži opis projekta i poveznice na slične projekte, te kontakt podatke
- karta.php – komponenta koja omogućava spajanje na MySQL bazu podataka, generira podatke potrebne za prikaz na karti te ih pohranjuje u XML dokument
- baza.sql – komponenta koja sadrži podatke o mjestima rođenja, godinama upisa i broju studenata
- bazainfo.php - komponenta s podacima potrebnim za spajanje s bazom podataka
- style.css – komponenta zadužena za dizajn web sjedišta

- markers.xml – podatci iz baze podataka u XML datoteci.

Na Slici 1 prikazan je završni izgled naslovne stranice ovog projekta i veza prema ostalim stranicama. Broj 1 prikazuje izbornik s poveznicama na ostale stranice, broj 2 prikazuje naslovnu traku koja sadrži naslov stranice, broj 3 prikazuje tijelo stranice na kojem se nalazi sadržaj, a broj 4 prikazuje podnožje stranice.



Slika 1: Naslovna stranica

4. Projektiranje i realizacija web sjedišta

U ovom dijelu prikazat ću proces izrade interaktivne karte uz pomoć Google Maps-a i MySQL baze podataka, kodiranje stranice u HTML i CSS kodu, te svrhu projekta. Temeljito ću proći kroz kod i pokušati što detaljnije objasniti kako radi, te kako su komponente ovog web sjedišta međusobno povezane.

4.1. Google Maps i PHP/MySQL

Za početak potrebna nam je baza podataka koja u sebi treba sadržavati atribute lat i lng¹. Ti atributi sadržavaju podatke o koordinatama mjesta koje želimo prikazati na karti. S trenutnim mogućnostima zumiranja Google Maps-a, potrebna nam je preciznost od samo 6 znamenki iza decimale. Tip podataka treba biti postavljen na float veličine (10,6), što znači da će se pohraniti 6 znamenki nakon decimale i dodatne 4 znamenke prije decimale. Uz te atribute, tablica bi također trebala sadržavati atribut id postavljen kao primarni ključ². Osim atributa id, dodani su još i atributi mjesto, god, smjer, upisani, BA i MA, koji sadrže podatke o mjestu iz kojeg dolaze studenti, godini upisa ili završetka studija, te broj upisanih studenata, odnosno onih koji su završili preddiplomski studij ili diplomirali (Kodni blok 1).

¹ Skraćeno od latitude i longitude (zemljopisna širina i dužina). Podatci o koordinatama pojedinih mjesta preuzeti su s web stranice www.distancesfrom.com.

² Svaka tablica u bazi podataka mora imati primarni ključ, odnosno polje ili skup polja s jedinstvenom vrijednošću za svaki zapis spremljen u tablicu. Primarni ključ može se koristiti za prepoznavanje pojedinog zapisa i kao referencu na njega.

```
CREATE TABLE `markers` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `mjesto` VARCHAR( 60 ) COLLATE utf8_unicode_ci NOT NULL,  
  `god` VARCHAR( 80 ) NOT NULL,  
  `smjer` VARCHAR( 80 ) NOT NULL,  
  `lat` FLOAT( 10, 6 ) NOT NULL,  
  `lng` FLOAT( 10, 6 ) NOT NULL,  
  `upisani` CHAR( 80 ),  
  `BA` CHAR( 80 ),  
  `MA` CHAR( 80 )  
  
  ) ENGINE = MYISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Kodni blok 1: Kreiranje tablice u SQL-u

Nakon što je tablica kreirana, potrebno ju je popuniti podacima. Podatke ćemo dodati pomoću naredbe INSERT i zatim navesti vrijednosti pravim poretom, što znači da će u tablici u stupcu mjesto biti pohranjeni nazivi gradova, u stupcu god godine itd. (Kodni blok 2).

```
INSERT INTO `markers` (`mjesto`, `god`, `smjer`, `lat`, `lng`, `upisani`,  
  `BA`, `MA`)  
VALUES ('Zagreb', '2013.', 'info', '45.8144400', '15.9779800', '2', '28',  
  '27');
```

Kodni blok 2: Naredna INSERT

Sad kada je kreirana tablica i popunjena podacima, potrebne su neke PHP izjave za izvoz podataka tablice u XML formatu koji će karta dohvatiti putem JavaScript poziva. Prvo treba pripremiti novi XML dokument i kreirati 'roditeljski čvor' (eng. *parent node*)³. Nakon toga potrebno je povezati se s bazom podataka i izvršiti naredbu SELECT * (odaberi sve) iz kreirane tablice kako bi mogli iterirati kroz podatke (Kodni blok 3).

³ čvor koji se nalazi na prvoj višoj hijerarhijskoj razini u odnosu na drugi čvor

```
// Pokretanje XML dokumenta i kreiranje roditeljskog čvora
$dom = new DOMDocument("1.0");
$node = $dom->createElement("markers");
$parnode = $dom->appendChild($node);

// Spajanje na MySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) { die('Not connected : ' . mysql_error());}

// Postavljanje aktivne MySQL baze podataka
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
die ('Can\'t use db : ' . mysql_error());
}
```

Kodni blok 3: Spajanje na MySQL bazu podataka

Sljedeći korak je kreiranje karte. Za kreiranje karte, kao i za njezino učitavanje prilikom pokretanja stranice, potrebno je dodati kod prikazan u Kodnom bloku 4. Unutar div elementa dodana je karta širine 820px i visine 600px, a pomoću elementa `<body onload="load()">`, karta će se učitati prilikom pokretanja stranice.

```
<body onload="load()">
    <div id="map" style="width: 500px; height: 300px"></div>
</body>
```

Kodni blok 4: Dodavanje karte u HTML

Za učitavanje XML datoteke na stranici, može se iskoristiti XMLHttpRequest objekt koji omogućuje preuzimanje datoteke koja se nalazi na istoj domeni kao i tražena web stranica, te je osnova "AJAX" programiranja. U ovom slučaju, definirana je funkcija za učitavanje datoteke,

funkcija `downloadUrl()` i ona uzima dva parametra. Prvi je `url` koji određuje put do XML dokumenta ili PHP-a koji generira taj dokument, ovisno o tome želimo li da se XML dinamički ažurira kada se baza podataka mijenja. Drugi je `callback` koji označava funkciju koja je pozvana kada je XML vraćen JavaScript-u (Kodni blok 5).

```
function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };
};
```

Kodni blok 5: Kreiranje funkcije `downloadUrl()`

Nakon što je kreirana funkcija, ona se može pozvati iz napisanog koda, pomoću PHP dokumenta i `callback` funkcije. `Callback` funkcija pronalazi sve elemente iz XML-a i iterira kroz njih, pronalazi te elemente i u ovom slučaju vraća attribute mjesto, godina, smjer, upisani, BA, MA i attribute `lat` i `lng`. Zatim kreira oznaku koju dodaje na kartu i info prozor s opisom nakon što korisnik klikne na oznaku (Kodni blok 6).


```
downloadUrl("karta.php?year=" + selectedYear, function(data) {
    var xml = data.responseXML;
    var markers = xml.documentElement.getElementsByTagName("upis");
    var markers = xml.documentElement.getElementsByTagName("pred");
    var markers = xml.documentElement.getElementsByTagName("dipl");

    for (var i = 0; i < markers.length; i++) {
        var name = markers[i].getAttribute("name");
        var god = markers[i].getAttribute("god");
        var smjer = markers[i].getAttribute("smjer");
        var upisani = markers[i].getAttribute("upisani");
        var BA = markers[i].getAttribute("BA");
        var MA = markers[i].getAttribute("MA");

        var point = new google.maps.LatLng(
            parseFloat(markers[i].getAttribute("lat")),
            parseFloat(markers[i].getAttribute("lng")));
```

Kodni blok 6: Pozivanje funkcije `downloadUrl()`

Za kreiranje oznaka, korišten je Google Maps JavaScript API objekt `Circle` pomoću kojeg je moguće odrediti boju, debljinu rubova itd. Veličina krugova i njihov položaj na karti određen je s dva dodatna svojstva, a to su `center` koji određuje gdje će se oznaka nalaziti pomoću `google.maps.LatLng` i `radius` koji određuje veličinu kruga. U ovom projektu, veličina krugova određena je brojem studenata, a položaj krugova određen je mjestom iz kojeg dolaze (Kodni blok 7).

```
if(selectedType == "upisani"){
    var upis = new google.maps.Circle({
        strokeColor: '#990000',
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: '#990000',
        fillOpacity: 0.5,
        map: map,
        center: point,
        position: point,
        radius: Math.sqrt(upisani) * 4000
    });
    bindInfoWindow(upis, map, infoWindow, infoupisani);
}
```

Kodni blok 7: Kreiranje oznaka na karti pomoću objekta Circle

Za kraj, još je potrebno kreirati info prozor s opisom. Info prozor kreiran je pomoću funkcije `bindInfoWindow` koja u sebi sadrži `google.maps.event.addListener`. Event Listener omogućuje korisniku da nakon što klikne na određenu oznaku na karti, dobije opis u obliku pop-up prozorčića s porukom koja se može sastojati od teksta ili slike. Nakon kreiranja funkcije `bindInfoWindow` (Kodni blok 8), potrebno je još samo dodati tekst koji želimo da se nalazi u info prozoru (Kodni blok 9).

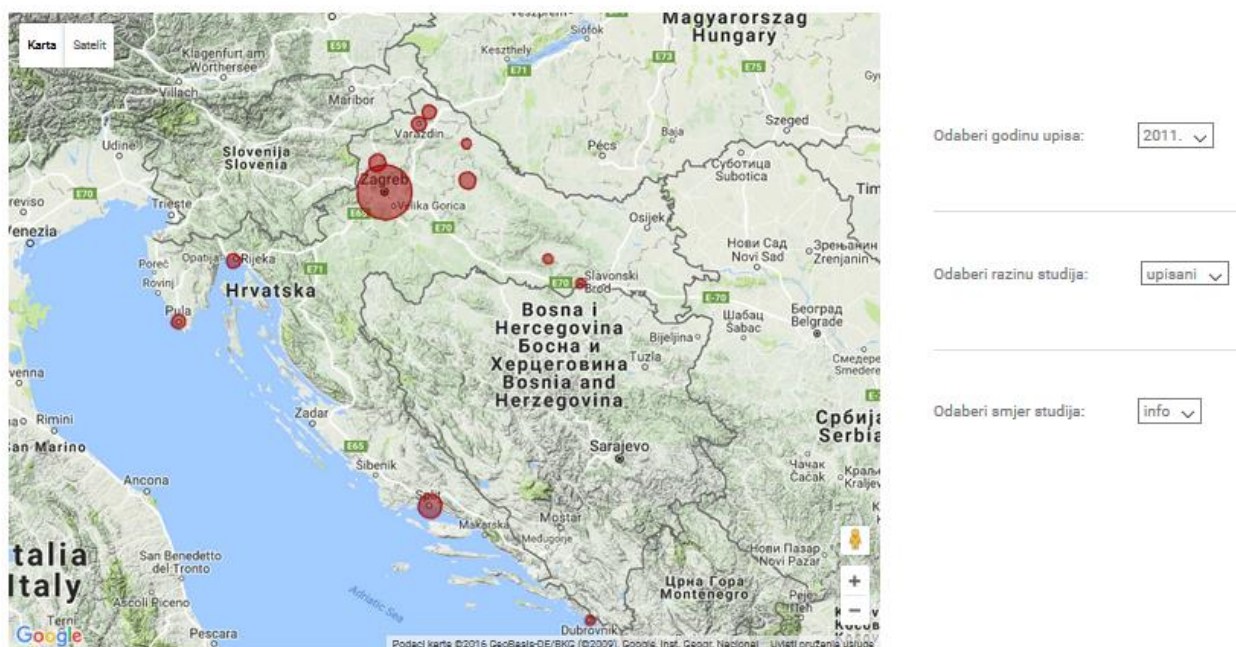
```
function bindInfoWindow(upis, map, infoWindow, infoupisani) {
    google.maps.event.addListener(upis, 'click', function() {
        infoWindow.setContent(infoupisani);
        infoWindow.open(map, upis);
    });
}
```

Kodni blok 8: Kreiranje funkcije bindInfoWindow()

```
var infoupisani = '<b>' + name + '</b>' + '<br/>' + 'Godina: ' + god +
'<br/>' + 'Smjer: ' + smjer + '<br/>' + 'Upisani: ' + upisani;
```

Kodni blok 9: Dodavanje teksta unutar info prozora

Nakon što su dodane sve komponente koje sam navela, dobit ćemo gotovu kartu prikazanu na Slici 2. Tu se može vidjeti da nakon što je odabrana 2011. godina i upisani studenti, na karti će se prikazati oznake na mjestima iz kojih ti studenti dolaze. Također, oznake koje su prikazane krugovima, određene su brojem upisanih studenata, što znači da najviše studenata koji su se upisali na Odsjek za informacijske i komunikacijske znanosti na Filozofskom fakultetu u Zagrebu 2011. godine, dolaze iz Zagreba. Osim toga na ovom primjeru možemo vidjeti da 2011. godine nije bilo upisanih studenata iz Osijeka i Zadra i njihove okolice, što možda možemo objasniti time što i Osijek i Zadar također imaju studij informacijskih znanosti na svojim sveučilištima.



Slika 2: Završni izgled karte

4.2.HTML struktura web sjedišta

Na naslovnoj stranici projekta, kao i na ostalim stranicama ovog web sjedišta, potrebno je definirati navigacijsku traku. U ovom slučaju, ona se nalazi u gornjem rubu stranice (Slika 1) i postavljena je horizontalno što je definirano CSS kodom koji ću objasniti kasnije.

Element `nav` koristi se za postavljanje navigacijskih linkova i namijenjen je većim blokovima linkova. Uz `nav` element, ovdje su još korišteni elementi `ul`, `li` i `a`. Elementi `ul` i `li` su elementi liste, a zajedno se koriste za izradu neuređene liste. Element `a` koristi se za definiranje linkova i dolazi s atributom `href` koji određuje destinaciju linka (Kodni blok 10).

```
<nav id="nav">
  <ul>
    <li class="active"><a href="index.html">Naslovnica</a></li>
    <li><a href="karta.html">Karta</a></li>
    <li><a href="oprojektu.html">O projektu</a></li>
  </ul>
</nav>
```

Kodni blok 10: Navigacijska traka

Na stranici koja sadrži kartu, potrebno je definirati i elemente pomoću kojih možemo izabrati što želimo da bude prikazano na karti. To je određeno pomoću elementa `select` koji kreira padajući izbornik. U ovom slučaju, kreirana su tri takva izbornika, jedan za odabir godine, drugi za odabir upisanih, BA ili MA studenata i treći za odabir smjera (Kodni blok 11).

```
<p>Odaberi godinu upisa:</p>
  <select id="byYear" onchange="load();">
    <option selected>2009.</option>
    <option>2010.</option>
    <option>2011.</option>
    <option>2012.</option>
    <option>2013.</option>
    <option>2014.</option>
    <option>2015.</option>
    <option>2016.</option>
  </select>
```

Kodni blok 11: Definiranje padajućeg izbornika

Stranica koja sadrži podatke o projektu, sastoji se od lijevog izbornika i teksta na desnoj strani (Slika 3).

Korisni linkovi

[Google Maps APIs](#)

[W3Schools](#)

[Google Maps Tutorial](#)

[DistancesFrom](#)

Kontakt

MENTOR:

dr.sc. Kristina Kocijan, doc
[krkocijan@ffzg.hr]

VODITELJ PROJEKTA:

Renata Waller
[rwaller@ffzg.hr]

O projektu

**Primjer informacijskog sustava u mrežnom okruženju:
upisani/diplomirani studenti informacijskih i komunikacijskih znanosti
Filozofskog fakulteta u Zagrebu na interaktivnoj karti Hrvatske**

Ideja ovog projekta bila je izraditi interaktivnu kartu koja sadrži podatke o tome koje godine je koliko studenata upisalo, završilo preddiplomski ili diplomiralo na Filozofskom fakultetu u Zagrebu. Trenutno, baza sadrži podatke od 2009. do 2016. godine, odnosno od uvođenja bolonjskog procesa do danas. Uz to, za sada je moguće vidjeti broj studenata samo sa odsjeka informacijskih i komunikacijskih znanosti, no pošto je projekt još u izradi, u budućnosti planiram proširiti bazu i za ostale odsjeka.

Svrha ovog projekta bila bi pomoći budućim studentima pri odabiru željenog studija tako što bi mogli vidjeti na kojim smjerovima ima manje, odnosno više diplomiranih studenata što bi im moglo pomoći pri zapošljavanju. Isto tako, ovaj projekt mogao bi dobro doći i budućim poslodavcima koji će moći planirati zapošljavanje nakon što imaju podatke koliko studenata je te godine završilo neku studijsku grupu iz njihovog grada. Osim toga, projekt može poslužiti i za usporedbu upisnih kvota na određenim smjerovima i broju studenata koji upisuju te smjerove i završavaju ih.

Google Maps API (Application Programming Interface) je sučelje za programiranje primarno namijenjeno integraciji u web stranice i prilagodbi vlastitim potrebama. Dostupan je za Android, iOS, web preglednike i putem HTTP web servisa. Za širu upotrebu Google Maps tehnologije, Google je razvio posebno programsko korisničko okruženje, Google Maps API, koji mogu koristiti druge aplikacije. Korisničko okruženje zasnovano na XML-u (eng. eXensible Markup Language) dodatno omogućava korisnicima da prilagode karte i da, uz pomoć položajnih podataka, kreiraju interaktivne karte za potrebe svojih aplikacija.

Slika 3: Izbornik i sadržaj stranice o projektu

Lijevi izbornik sadrži korisne poveznice i kontakt podatke. On se sastoji od HTML elemenata `div`, `header` i `section`. Element `div` definira odjeljak u HTML dokumentu i koristi se za grupiranje blokova koji se onda mogu uređivati CSS-om. Element `section` koristi se za definiranje sekcija kao što su zaglavlje, podnožje, poglavlje ili bilo koji drugi odjeljak dokumenta. Element `header` sadrži uvodni sadržaj ili skup navigacijskih poveznica i uglavnom se sastoji od jednog ili više heading elemenata (Kodni blok 12).

```
<div id="sidebar">
  <section>
    <header>
      <h2>Korisni linkovi</h2>
    </header>
    <div class="row">
      <section>
        <ul class="default">
          <li><a href="link">Google Maps APIs</a></li>
          <li><a href="link">W3Schools</a></li>
          <li><a href="link">Google Maps Tutorial</a></li>
          <li><a href="link">DistancesFrom</a></li>
        </ul>
      </section>
    </div>
  </section>
</div>
```

Kodni blok 12: Lijevi izbornik na stranici o projektu

4.3.CSS struktura web sjedišta

U ovom projektu korištena je vanjska CSS datoteka nazvana `style.css`. Korištenje vanjskog CSS-a uvelike olakšava posao jer mijenjanjem samo jedne CSS datoteke možemo utjecati na izgled cijelog web sjedišta, pa tako i u ovom slučaju gdje sve stranice web sjedišta imaju identično

zaglavlje i podnožje. Ako koristimo vanjski CSS, svaka stranica u elementu `head` mora imati poveznicu na tu CSS datoteku unutar `link` elementa (Kodni blok 13).

```
<head>
  <link rel="stylesheet" href="css/style.css" />
</head>
```

Kodni blok 13: Poveznica na vanjsku CSS datoteku

Da bi se iz vanjske CSS datoteke moglo pozvati određeni element unutar HTML datoteke koriste se selektori. Za svaki element u HTML dokumentu treba dodati atribute `id` ili `class` kojima dajemo imena po želji radi lakšeg snalaženja.

U uređivanju navigacijske trake korišten je `nav` element određen svojstvom `position: absolute;` koji u ovom slučaju za poziciju koristi tijelo dokumenta i pomiče se kako se pomiče i sama stranica. Uz postavljanje pozicije `nav` elementa određeni su i elementi `ul` i `li` unutar tog `nav` elementa. Svojstvo `margin` definira prostor oko elemenata, dok svojstvo `padding` definira prostor oko sadržaja elementa. Najbitnije svojstvo što se tiče navigacijske trake je `display` koji definira kako će se elementi prikazati. U ovom slučaju korišteno je svojstvo `display: inline-block;` koji elemente liste postavlja u horizontalni položaj (Kodni blok 14).

```
#nav-wrapper
{
    background: rgba(0,0,0,.1);
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
}

#nav > ul
{
    margin: 0;
    padding: 0;
    text-align: center;
}

#nav > ul > li
{
    display: inline-block;
    border-right: 1px solid;
    border-color: rgba(255,255,255,.1);
}
```

Kodni blok 14: CSS svojstva kojima je uređena navigacijska traka

Pozadinska slika postavljena je uz pomoć svojstva `background`. Pošto je prema zadanim postavkama određeno da se slika ponavlja tako da prekrije cijeli element, potrebno je `background` svojstvu dodati `no-repeat`. Uz to imamo i svojstvo `background-size: cover;` koje određuje da slika bude proširena što je više moguće, tako da područje pozadine bude u potpunosti prekriveno. Za razliku od navigacijske trake koja se pomiče zajedno sa stranicom, pozadinska slika određena je svojstvom `position: fixed;` koje ne dozvoljava slici da se pomiče, već ostaje na istom mjestu (Kodni blok 15).


```
#header
{
    position: relative;
    background: url(../images/header.png) no-repeat bottom center;
    background-attachment: fixed;
    background-size: cover;
    text-align: center;
}
```

Kodni blok 15: CSS svojstva kojima je uređena pozadinska slika

5. Zaključak

U ovom radu opisala sam postupak nastajanja i dizajniranja informacijskog sustava u mrežnom okruženju od ideje do završnog proizvoda.

Pošto se sam rad temelji na izradi konkretnog primjera web sjedišta, detaljno sam objasnila svaki dio njegove izrade. Za uvod sam opisala osnovne pojmove koji su bili potrebni u izradi projekta, kao i kako funkcionira projektiranje i realizacija web sjedišta, te objasnila njihovu ulogu u izradi projekta. Zatim sam objasnila zašto sam se odlučila za ovakav projekt, kao i njegovu ideju i svrhu, te planove za budućnost. Kao što sam već napomenula, projekt je još u testnoj fazi i planiram ga proširiti s izborom za sve odsjeke kao i izborom smjerova diplomskog studija. Nakon što sam dala plan izrade i informacijsku strukturu svog web sjedišta, u trećem dijelu rada, detaljno sam objasnila kodiranje stranice u HTML i CSS kodu, kao i programiranje u klijentskim i poslužiteljskim skriptnim jezicima i modeliranje baze podataka.

Najveći dio rada posvetila sam izradi interaktivne karte pomoću Google Maps JavaScript API-ja. Krenula sam od izrade i popunjavanja baze podataka, nakon čega sam objasnila kako se pomoću PHP izjava spojiti na MySQL server i pokrenuti XML dokument koji mi je bio potreban u izradi karte. Nakon toga sam detaljno opisala funkcije korištene u izradi oznaka i info prozora na karti napisanih u JavaScript jeziku.

Ovaj projekt može poslužiti za usporedbu upisnih kvota i diplomiranih studenata na određenim smjerovima, ali i usporedbu upisnih kvota ostalih fakulteta u Hrvatskoj. Mislim da bi ovaj projekt mogao pomoći budućim studentima pri odabiru željenog studija tako što bi mogli vidjeti na kojim smjerovima ima manje, odnosno više diplomiranih studenata što bi im moglo pomoći pri zapošljavanju. Isto tako, ovaj projekt mogao bi dobro doći i budućim poslodavcima koji će moći planirati zapošljavanje nakon što imaju podatke koliko studenata je te godine završilo neku studijsku grupu iz njihovog grada.

6. Literatura

1. CARNet, „Društveni softver: Web 2.0“, <http://www.carnet.hr/tematski/drustvenisoftver/pojmovnik>, (posjećeno 12.8.2016.), 2009.
2. Crowther, R., „Hello! HTML5 & CSS3“, Manning Publications Co, Shelter Island, NY 11964, 2013. Str: 233-424.
3. distancesfrom.com, „DistancesFrom“, <http://www.distancesfrom.com/Latitude-Longitude.aspx>, (posjećeno 10.7.2016.), 2016.
4. Google Developers, „Google Maps APIs“, <https://developers.google.com/maps/>, (posjećeno 2.8.2016.)
5. Henick, B., „HTML & CSS: The Good Parts“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2010. Str: 7-24.
6. Nixon, R., „Learning PHP, MySQL, and JavaScript“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2009. Str: 337-419.
7. O'Reilly, „What is Web 2.0?“, <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>, (posjećeno 12.8.2016.), 2005.
8. Radovan, M., „Projektiranje informacijskih sustava“, Informator, Zagreb, 1991.
9. Schmitt, C., Simpson K., „HTML5 Cookbook“, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2012. Str: 1-27.
10. Tuđman M., Boras D., Dovedan Z., „Uvod u informacijske znanosti“, [http://www.ffzg.unizg.hr/infoz/dzs/text/Uvod u informacijske znanosti/index.html](http://www.ffzg.unizg.hr/infoz/dzs/text/Uvod%20u%20informacijske%20znanosti/index.html), (posjećeno 12.8.2016.), Zagreb, 1991.
11. W3Schools, „W3Schools“, <http://www.w3schools.com>, (posjećeno: 12.8.2016.)