Raspored za studenta

Bašić, Andi

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

RASPORED ZA STUDENTA

Diplomski rad

Andi Bašić

Osijek, 2017.

Sadržaj

1. UVOD

Raspored je osnovni način upravljanja vremenom i sastoji se od popisa vremena s mogućim zadacima, događajima ili radniji koje će se održati ili pak slijeda događaja u kronološkom redu u kojem se neke stvari namjeravaju održati. Proces stvaranja rasporeda je odlučivanje kojim će se redom zadaci izvršavati i kako uložiti resurse između različitih mogućih zadataka da bi ostvarili ciljeve.

Rasporedi mogu obuhvaćati kraće i duže vremenske razdoblje te postoje kratkoročni dnevni i tjedni ili dugoročni mjesečni i godišnji rasporedi. Rasporedi se često izrađuju pomoću kalendara gdje osoba koja pravi raspored može zabilježiti datume i vremena u kojima se planiraju razni događaji. Postoji velik broj različitih vrsta rasporeda, ali za potrebe moga rada bavit ću se rasporedom nastave za studente.

Raspored nastave za studenta je tablica s kordiniranjem četiri glavna elementa elementa: studenti, kolegiji, prostorije i vrijeme izvođenja nastave.[1] Nastava i predavanja na faksu ne održavaju se ciklički, odnosno ne ponavljaju se jednaka predavanja svaki tjedan te posao satničara nije jednostavan. Satničar se mora brinuti o satnici svih kolegija i profesora, mora voditi računa o slobodnim prostorijama na faksu te isto tako o vremenu održavanja nastave. Iako raspored nastave i predavanja postoji na službenoj stranici fakulteta, moj zadatak bio je iz dobivenih resursa generirati i napraviti funkcionalan raspored nastave svih studenata fakulteta.

1.1. Zadatak diplomskog rada

Zadatak diplomskog rada je napraviti programsku izvedbu osobnog rasporeda nastave za svakog pojedinog studenta Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Iz postojećih resursa s podacima potrebno je generirati prikaz rasporeda, tako da se student najprije identificira, a nakon toga dobiva svoj vlastiti raspored nastave i ispita. Odlučio sam se za izradu web aplikacije koristeći se PHP programskim jezikom.

2. PROGRAMSKO OKRUŽENJE PHP

2.1. Uvod

Source kod je datoteka u kojoj se piše izvorni kod programa. Da bi se izvorni kod izveo na računalu, on se prvo mora kompajlirati u *binary* kod te se potom *binary* kod izvodi. Primjerice, u programskom jeziku C, napisani kod kompajliramo u .exe datoteku te pokretanjem .exe datoteke pokrećemo i sam program kojeg je moguće pokrenuti na Windows operacijskom sustavu, dok se primjerice na Linux operacijskom sutavu kod kompajlira u .sh datoteku koja se izvodi kao takva. Za razliku od C i Java kompajlerskih programskih jezika, PHP je interpreter što znači da se kompajlira svakog puta kada se kod izvodi. Izvođenjem kompajlacije, kod se prvo kompajlira u C, nakon toga se C prebacuje u *binary* kod i naposlijetku se izvodi kao takav što je jedan od razloga kritika na račun PHP programskog jezika. No izuzev toga, PHP je odličan za rad s XML datotekama i parsiranjem istih te je za rad s PHP-om potrebno poznavati HTML i CSS.

2.2. Programski jezik PHP

PHP je skriptni jezik na strani poslužitelja izvorno stvoren za izradu dinamičnog web sadržaja te se i danas u najvećoj mjeri koristi zbog toga. Moguće ga je koristiti i kao jezik opće namjene. Slično nekim drugim programskim jezcima (Pearl, Unix), PHP može izvoditi skripte iz naredbene linije. Isto tako jezik omogućava razvoj GUI aplikacije za korištenje na cijeloj platformi.[2] Izvorni tvorac PHP-a je Rasmus Lerdorf, davne 1994. godine. Danas je za PHP-ovu implementaciju zadužen PHP razvojni tim. Originalno je PHP kratica označavala *Personal Home Page* odnosno osobna početna stranica dok danas vrijedi rekurzivni akronim *PHP: Hypertext Preprocessor*. Jezik ima veliku fleksibilnost s time da prikaz rezultata i izlaz programa ne mora nužno biti u HTML formatu. Jezik omogućava generiranje različitih vrsta datoteka kao što su GIF, PNG i JPG slike, Flash animacije i filmovi te PDF datoteke. Podržava rad s bazama podataka uključujući sve veće baze podataka tipa MySQL, Oracle, Sybase, DB2, PostgreSQL te isto tako i novije NoSQL baze podataka kao MongoDB i SQLite. Najnovije istraživanje je pokazalo da se PHP koristi na skoro 80% web lokacija što znači da još uvijek ima široku primjenu i da se uvelike koristi.

2.3. Povijest programskog jezika PHP

1995. godine započeo je razvoj PHP programskog jezika kada je Rasmus Lerdorf napisao par programa u C-u čijim bi radom održavao svoju osobnu početnu stranicu. Kasnije je dodao mogućnost komuniciranja s bazama podataka te radu s web obrascima te je tu implementaciju nazvao *Personal Home Page/Forms Interpreter* ili u prijevodu osobna početna stranica ili tumač formi. Prva verzija izdana je pod nazivom *Personal Home Page Tools version 1.0* te je najveću namjenu imala u izradi jednostavnijih dinamičnih web aplikacija. Prva verzija je već tada imala osnovne funkcionalnosti koje PHP ima od 2013. godine, kao što su oblikovanje obrazaca, mogućnost ugrađivanja HTML-a, sintaksa slična Perlu itd.

1997. godine Andi Gutmans i Zeev Suraski su ponovno napisali *parser* te formirali bazu PHP 3 te isto tako promijenili naziv jezika u rekurzivni akronim *PHP: Hypertext Preprocessor*. Nakon javnih testiranja, službena verzija je objavljena u lipnju 1998. Nakon objave, spomenuti dvojac je preradio jezgru PHP-a te 1999. proizveo Zend Engine.

2004. godine, 13. svibnja izdana je nova PHP 5 verzija pogonjena novim Zend Engine II. PHP 5 uključuje brojne nove mogućnosti i poboljšanja kao što je poboljšana podrška za objektno orijentirano programiranje, *PHP Data Objects* (PDO) proširenja kojima je definirano dosljedno sučelje za pristup bazama podataka te brojna poboljšanja performansi. 2008. godine, PHP 5 je postao jedina stabilna verzija u razvoju. Tijekom vremena, PHP prevoditelji su postali dostupni na većini postojećih 32 i 64-bitnih operacijskih sustava.

2005. godine Andrei Zmievski je pokrenuo projekt kako bi se uvela podrška Unicoda u PHP ugrađivanjem biblioteke za internacionalne znakove i predstavljajući tekstualnih nizova kao UTF-16. Zbog određenih problema, projekt su odgodili do 2009. kada je verzija PHP 5.4 objavljena s mnogim ne Unicode značajkama vraćenih iz PHP 6 verzije.

Tjekom 2014. i 2015. godine, nova glavna PHP verzija se razvila koja je bila numerirana PHP 7. Unicode 6 projekt nikada nije bio izdan. Glavni tvorci bili su Dmitrij Stogoc, Xinchen Hui i Nikita Popov, čiji je cilj bio optimizirati performanse jezika prerađujući Zend Engine za upotrebu kompaktnijih podatkovnih struktura s poboljšanom lokacijom predmemorije zadržavajući gotovo potpunu kompatibilnost jezika.[3] Mjerenja na WordPress platformi su pokazala gotovo stopostotno povećanje performansi. Zbog značajnijih promjena nastao je Zend Engine 3. PHP 7 je predstavio nove jezične značajke, uključujući povratni tip deklaracije za funkcije te podršku za skalarne vrste podataka: *integer*, *float*, *string* i *boolean*.

2.4. Instalacija PHP-a

Za samu instalaciju PHP-a na računalu, potrebno je imati predinstaliranu Visual C runtime. PHP 5.5 i 5.6 zahtjevaju instaliranu VC CRT 11, dok PHP 7.0 zahjeva instaliranu VC CRT 14.

Dvostrukim klikom na HTML datoteku otvorila bi se datoteka u web pretraživaču, no dvostrukim klikom na PHP datoteku, ona bi se vjerojatno otvorila u nekom od tekstualnih uređivača. Razlog tomu je što PHP datoteke prvo moraju biti procesuirane kroz web server prije slanja njegove izlazne informacije na web pretraživač.

Iz gore navedenog razloga, datoteke s .php ekstenzijom moraju biti smještene unutar web datoteke web servera kojim pristupamo preko URL-a u web pretraživaču. Ako imamo instaliran web server na računalu, korjenskom folderu pristupamo preko *http://localhost* u web pretraživaču. Ako primjerice datoteku nazovemo *pozdrav.php* i smjestimo ju unutar web direktorija, toj datoteci preko web pretraživača pristupamo preko *http://localhost/pozdrav.php*.

Web direktorij može biti drugačiji ukoliko je stranica smještena na internetu ili ako je korjenski direktorij web servera s računala postaljen na drugoj lokaciji. U slučaju korištenja XAMPP programa za instalaciju Apache web servera, direktorij na računalu je *htdocs* koji se nalazi unutar korijenskog direktorija XAMPP-a.

Postoje brojni načini instaliranja i pokretanja PHP koda no najkorišteniji način je instalacijom web servera.

2.4.1. XAMPP

XAMPP je besplatna i jednostavna multi-platforma s čijom instalacijom dolazi Apache, MySQL, PHP i brojne druge aplikacije korisne za razvijanje ili testiranje dinamičkih web stranica.

Najjednostavniji način za pokretanje PHP programa je korištenjem programskog paketa XAMPP. Postoje brojni slični programi koji mogu pokretati PHP programe, ali XAMPP je najpopularniji te ga je moguće pokrenuti na svim većim platformama. Instalacija platrofme obuhvaća sve potrebne pakete za rad s PHP-om: Apache, PHP, MySQL.[4]

Koraci za pokretanje prvog PHP koda pomoću XAMPP-a:

1. Potrebno je instalirati XAMPP kojeg je moguće preuzeti sa službene stranice.

2. Otvoriti bilo koji uređivač teksta. Primjerice Sublime Text.

3. U uređivaču teksta napisati PHP kod koji ćemo kasnije pokrenuti preko web pretraživača.

```
<?php
echo 'Ovo je moj prvi PHP program pokrenut preko XAMPP-a! <br />'?>
?>
```

4. Datoteku je potrebno spremiti unutar XAMPP instalacijskog direktorija u korijenskom web direktoriju.

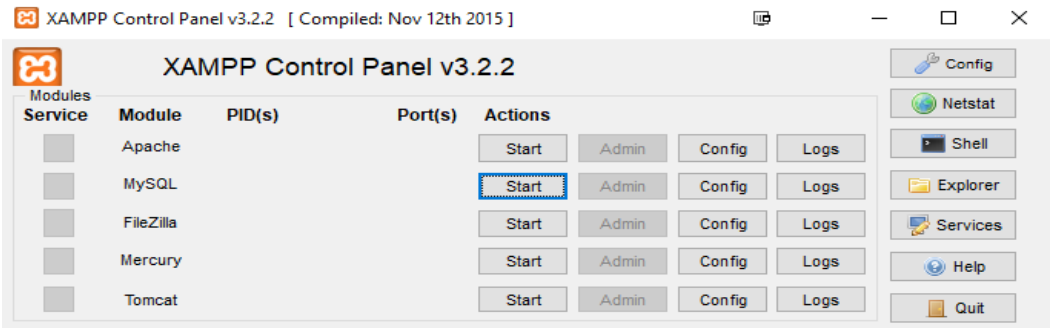
   Zadani instalacijski direktorij u Windowsima je C:\xampp.

   Zadani korijenski web direktorij je *htdocs*. Sve datoteke s ekstenzijom php koje želimo pokrenuti pomoću XAMPP-a, moraju se nalaziti u tom direktoriju.

5. Napokon spremanja datoteke u zadani folder, potrebno ju je imenovati s ekstenzijom php, primjerice *prvi_kod.php*. Treba pripaziti kada spremamo datoteku da nema .txt ekstenzije na kraju jer neki uređivači teksta automatski dodaju .txt nastavak te bi tako naziv naše datoteke bio *prvi_kod.php.txt* umjesto *prvi_kod.php*.

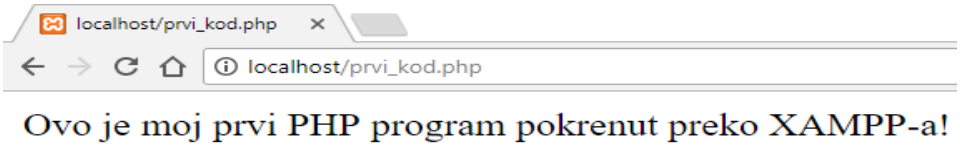
6. Nakon toga, potrebno je pokrenuti XAMPP.

7. Otvaranjem prozorčića vidimo brojne module kao što su Apache, MySQL, FileZilla, Mercury i Tomcat te je za naše potrebe pokretanja prvog php programa potrebno samo pokrenuti Apache server klikom na *Start*.



Slika 2.4.1. Kontrolna ploča

8. Nakon pokretanja Apache servera potrebno je u web pretraživač upisati *http:/localhost/prvi_kod.php*. Ako smo sve korake ispravno napravili, u web pretaživaču bi trebalo ispisati „Ovo je moj prvi PHP program pokrenut preko XAMPP-a!“



Slika 2.4.2. Pokrenuti program

2.5.Sintaksa PHP-a

Velik utjecaj na PHP programski jezik imali su programski jezici C i Perl. Kad se započinje rad u nekom programskom jeziku, ponajprije je potrebno naučiti leksičku strukturu samoga jezika. Leksička struktura je zapravo skup normi i pravila koje se moraju poštovati kako bi napisani kod bio ispravan i valjan. PHP interpreteri jedino izvršavaju kod koji se nalazi između *<?php ?>* oznaka koje predstavljaju početak i kraj PHP dijela koda. Varijable u PHP-u razlikuju mala i velika slova dok ključne riječi i ugrađene konstrukcije ne razlikuju velika slova. Kod jednostavnijih izraza koristi se točka-zarez koji označava kraj naredbe. Bjeline nisu bitne što znači da iskaze možemo raširiti preko više redova ili pak neki veći izraz napisati u jednom redu. Postoji više načina zapisa komentara, ali najprihvatljiviji je način uporabom dvije kose crte iza koje upisujemo komentar ili uporabom crte-zvjezdice za komentiranje više linija koda. Sve varijable počinju znakom dolara te razlikuju velika i mala slova dok funkcije ne razlikuju.

Ključna riječ u programskom jeziku je riječ kojima se definira neka od temeljnih funkcionalnosti. U tablici ispod nalaze se sve ključne riječi u PHP jeziku.

__CLASS__	echo	insteadof
__DIR__	else	interface
__FILE__	elseif	isset()
__FUNCTION__	empty()	list()
__LINE__	enddeclare	namespace
__METHOD__	endfor	new
__NAMESPACE__	endforeach	or
__TRAIT__	endif	print
__halt_compiler()__	endswitch	private
abstract	endwhile	protected
and	eval()	public
array()	exit()	require
as	extends	require_once
break	final	return
callable	finally	static
case	for	switch
catch	foreach	throw
class	function	trait
clone	global	try
const	goto	unset()
continue	if	use
declare	implements	var
default	include	while
die()	include_once	xor
do	instanceof	yield

Tab 2.5.1. Ključne riječi

PHP ima podršku za osam tipova podataka od kojih su četiri skaralni tipovi: brojevi s pokretnim zarezom, nizovi znakova, logičke vrijednosti i cjelobrojne vrijednosti, dva su složeni tipovi podataka: objekti i polja i dva su posebni tipovi: NULL vrijednost i resurs.

Polja su posebne varijable koje mogu sadržavati više od jedne vrijednosti odjednom. Vrijednosti možemo identificirati pomoću imena za identifikaciju ili prema poziciji te samim time postoje dvije vrste nizova: indeksni nizovi i asocijativni. Postoji dva načina kreiranja indeksnog niza: Prvi je taj da se indeks dodjeljuje automatski počevši od nula:

```
$automobili = array("BMW", "Mercedes", "Mazda");
```

Drugi način je da indeks dodijelimo ručno:

```
$automobili[0]  = "BMW";
$automobili[1]  = "Mercedes";
$automobili[2]  = "Mazda";
```

Asocijativni nizovi su polja koja upotrebljavaju imenovane ključeve koje se dodjeljuju polju. Isto tako postoji dva načina kreiranja asocijativnog niza:

```
$godine = array("Marko" => "23", "Janko" => "33", "Ivanko" => "43");
```

```
$godine['Marko'] = "23";
$godine['Janko'] = "33";
$godine['Ivanko'] = "43";
```

Od verzije PHP 3 uvedena je funkcionalnost objektno-orijentiranog programiranja. Nova funkcionalnost je omogućila dodavanje apstrakcije te je programerima olakšalo kreiranje kompliciranijih zadataka uz lakše održavanje te mogućnosti ponovne upotrebe koda. U novijim verzijama dodatno je unaprijeđena funkcionalnost te su u PHP 5 verziji predstavljene *private* i *protected* varijable članova i metoda uz apstraktne klase i metode. Također je uveden standard za deklariranje konstruktora i destruktora, slično onome iz C++ objektno-orijentiranog jezika. Dodan je i interfejs i mogućnost višestruke implementacije interfejsa. Osnovna svojstva OOP-a su: učahurivanje, apstrakcija, nasljeđivanje i višeobličje.

Klasa u objektno orijentiranom programiranju predstavlja predložak za stvaranje objekta. Ona je struktura koja sadrži funkcije i varijable te se u programskom jeziku PHP definira sa ključnom riječi *class*. Pomoću ključne riječi *new* se stvaraju objekti koji su instance klase, a varijablama i funkcijama objekta pristupa se pomoću znaka ->.

3. XML I PARSIRANJE

3.1. XML

XML je kratica za *Extensible Markup Language*, odnosno jezik označavanja podataka koji kroz uporabu oznaka definira set pravila za kodiranje određenih dokumenata u formatu razumljivim i čitljivijim i stroju i čovjeku.[5] Format oznaka XML jezika je vrlo sličan formatu HTML jezika. XML svojim formatom omogućava puno lakše parsiranje dokumenta što znači da imamo mogućnost izdvajanja svih mogućih podataka iz određenih XML resursa ili datoteka, koje kasnije možemo prikazivati na različite načine. Danas je XML poprilično rasprostranjen te se koristi u različite svrhe: pohranu, razmjenu i izdvajanje podataka, baze podataka, daljinsko povezivanje procedura, razmjenu informacija između računalnih sistema te se isto tako koristi se u različitim područjima: medicini, izdavačkim i narudžbenim sistemima itd. Za standardizaciju XML jezika se brine World Wide Web Consortium te je prva verzija objavljena 1998. godine.

Stotine vrsta formata dokumenata razvijeno je koristeći XML sintaksu uključujući RSS, SOAP, SVG, Atom i XHTML. XML bazirani formati su postali zadani formati za mnoge programe kao što su Microsoft Office (Office Open XML), LibreOffice (OpenDocument), Appleov iWork i OpenOffice.org. Aplikacije za Microsoftov .NET Framework koriste XML datoteke za konfiguracije, dok Apple koristi XML pri implementaciji registara.

XML format sastoji se od većeg broja podataka, entiteta i elemenata.

```
<vozilo serijski_broj=“3213-132225-634“>
    <naslov> Automobili </naslov>
    <automobili>
        <automobil> BMW </automobil>
        <automobil> Mercedes </automobil>
   </automobili>
</vozilo>
```

Početak XML dokumenta mora sadržavati prolog u kojoj mora biti definirana verzija XML.

```
<?xml version=“1.0“ ?>
```

U samom prologu moguće je definirati i kodnu stranicu koja će se koristiti u dokumentu u slučaju korištenja znakova i slova neengleske abecede.

```
<?xml version=“1.0“  ecnoding=“UTF-8“?>
```

XML standard pisanja dokumenata zahtjeva da svaka otvorena oznaka mora biti i zatvorena te isto tako svaki XML element mora imati početnu i završnu oznaku.

```
<automobil> BMW </automobil>
```

Oznake smiju biti ugnježđene, ali moraju biti pravilno formatirane i ne smije dolaziti do preklapanja oznaka.

Neispravno:

```
<automobil><proizvođač> BMW <automobil></proizvođač>
```

Ispravno:

```
<automobil><proizvođač> BMW </proizvođač><automobil>
```

Jedan od standarda XML dokumenata je da treba postojati jedan korijenski element na prvoj razini u datoteci koji uokviruje ostali sadržaj u njemu.

```
<automobil>
   <proizvođač> BMW </proizvođač>
   <godište> 1993 </godište>
   <prijeđeni_kilometri> 200 000 </prijeđeni_kilometri>
   <broj_vrata> 5 <broj_vrata>
</automobil>
```

Moj diplomski rad uvelike je zasnovan na radu s dvije XML datoteke iz kojih je moj zadatak dohvat svih potrebnih stavki za generiranja prikaza rasporeda.

Prva datoteka prikazuje formatiran prikaz rasporeda sati za zadani datum. Korijenski element je raspored u formatu YY-MM-DD s atributima tjedna, tip tjedna i redni broj tjedna. Idući element je stavka rasporeda koja ima svoj identifikacijski broj te se u njemu nalaze elementi smjer, predmet, nastavnik, vrsta nastave, početak, kraj, prostorija, grupa studenta, dodatni opis, odrađeni broj sati, planirani broj sati. Element smjer ima samo jedan atribut i to je identifikacijski broj smjera. Atributi predmeta su mrkveid, isuid, semestar, vrsta i sifra. Nastavnik ima atribute mat_broj, domaci, aai što zapravo predstavlja email nastavnika. Vrsta nastave ima samo jedan atribut tip, kao element prostorija čiji je atribut idprostorije te grupa studenata s atributom idgrupe. Na sljedećoj dijelu koda je prikazan već spomenuti opis svih elemenata XML dokumenta te pripadajuće stavke koje se nalaze u rasporedu za označeni datum.

http://mrkve.etfos.hr/api/raspored/index.php?date=2017-4-4

```
<?xml version="1.0" encoding="Windows-1250"?>
<raspored datum="2017-4-4" tjedan="14" tiptjedna="1" rednibrojtjedna="6">
  <stavkaRasporeda idblok="4634">
    <smjer idsmjer="2">Sveučilišni preddiplomski studij računarstva</smjer>
    <predmet mrkveid="1617090" isvuid="37115" semestar="2" vrsta="0"
sifra="P205">Programiranje II</predmet>
    <nastavnik mat_broj="210" domaci="1" aai="nenadic@etfos.hr">NENADIĆ
KREŠIMIR</nastavnik>
    <vrstanastave tip="1">PR - predavanja</vrstanastave>
    <pocetak>08:00</pocetak>
    <kraj>11:15</kraj>
    <prostorija idprostorije="10">K2-1</prostorija>
    <grupastudenata idgrupe="4075">PR</grupastudenata>
    <dodatniopis></dodatniopis>
    <odradjeno>26</odradjeno>
    <planirano>30</planirano>
  </stavkaRasporeda>
       .
       .
       .
```

Druga datoteka prikazuje formatiran prikaz studentskih grupa koje imaju svoj poseban identifikacijski broj. Unutar korijenskog elementa grupa nalaze se elementi za broj grupe, ime, opis te predmet koji u atributima ima poseban identifikacijski broj te kao drugi atribut ima broj semestra u kojemu se predmet nalazi. Idući element je studenti čiji skup čime svi studenti određene grupe. Student za svoje elemente ima jmbg, ime, prezime, oznaku redovni dok studenti kao element ima atribut ukupnog broja studenata grupe. Svaka grupa ima svoj poseban identifikacijski broj koji je označen u URL stranice i kojeg možemo mijenjati kako bi dobili prikaz druge grupe studenata. Tim putem možemo dobiti prikaz svih grupa koje su formatirane s obzirom na neke zajedničke karakteristike određenih studenta, tipa studij, smjer, godina, predmet itd. Na idućoj stranici naveden je dio koda u kojemu se nalazi prikaz podataka za grupu pod identifikacijskim brojem 4075.

http://mrkve.etfos.hr/api/raspored/grupa.php?idgrupe=4075

```
<?xml version="1.0" encoding="Windows-1250"?>
<grupaStudenata idgrupe="4075">
  <brojGrupe>1</brojGrupe>
  <ime>PR</ime>
  <opis></opis>
  <predmet isvuid="37115" semestar="2">Programiranje II</predmet>
  <studenti brojStudenata="135">
    <student>
      <jmbag>0165068474</jmbag>
      <ime>Filip</ime>
      <prezime>Anđelović</prezime>
      <redovni>1</redovni>
    </student>
    <student>
      <jmbag>0165070084</jmbag>
      <ime>Sara</ime>
      <prezime>Aščić</prezime>
      <redovni>1</redovni>
    </student>
      .
      .
      .
```

3.2.PARSIRANJE

Parisranje je čin razdvajanja informacija u svoje sastavne dijelove. U računarstvu, parser je program, dio koda ili API koji se može referencirati unutar vlastitih programa koji analizira datoteke za prepoznavanje dijelova komponenti. Sve aplikacije koje čitaju neki ulaz imaju nekakvu vrstu parsera. Primjerice, Microsof Word sadrži parser koji se pokreće kada se otvori .doc datoteka te samim time provjerava može li identificirati sve skrivene kodove. Zapisivanje strukture se vrši pomoću *Document Type Definition* ili definicije tipa dokumenata i sheme koje se koriste za ovjeravanje dokumenata.

XML aplikacije sadrže parser koji čita XML i identificira funkcije svakog dijela dokumenta te tu informaciju čini dostupnom u memoriji za ostatak programa. Tijekom čitanja XML datoteke, parser provjerava sintaksu (šiljaste zagrade, podudarajuće odgovarajuće znakove itd.) za dobro formatiranu datoteku ili izvještava o mogućim problemima. PHP u sebi ima tri parsera: SimpleXML, biblioteka temeljena na DOM-u i biblioteka temeljena na Expat C-u.

3.2.1. SimpleXML

SimpleXML je dodatak koji omogućava lako manipuliranje XML podacima te pruža jednostavan način dohvatka imena, atributa i tekstualnog dijela elemenata uz obavezno poznavanje strukture XML dokumenta.[6] Od PHP 5 verzije, SimpleXML funkcije su dio PHP jezgre pa nije potrebno instalirati nikakve dodatke za korištenje funkcija. Glavna funkcija za korištenje ovim dodatkom je *simplexml_load_file()* koja XML datoteku generira u objekt. Brojači unutar objekta omogućavaju pristup elementima u svakom čvoru. Samim time, možemo pristupiti svim elementima pomoću numeričkih indeksa te atributima preko nenumeričkih indeksa.[7] Moguća je i daljnja pretvorba niza znakova na bilo koju vrijednost kako bi ekstraktirali tekstualni podatak. Za razliku od DOM parsera, SimpleXML nije memorijski učinkovita te nema mogućnost generiranja dokumenta no vrlo je jednostavna za korištenje što je prikazano na idućem primjeru.

Primjer Automobili.xml datoteke:

```
<?xml version=1.0“ ?>
<projadni_centar>
    <vozilo>
        <proizvođac>
            <automobil> BMW </automobil>
            <automobil> Mercedes </automobil>
        </proizvođac>
        <broj_vrata> 4 <broj_ vrata>
        <komentar> </komentar>
    </vozilo>
    <vozilo>
        <proizvođac>
            <automobil> Audi </automobil>
            <automobil> Ford </automobil>
        </proizvođac>
        <broj_ vrata> 5 <broj_ vrata>
        <komentar> </komentar>
    </vozilo>
</prodajni_centar>
```

Proizvođače vozila iz gore navedene datoteke dobili bi preko:

```
$document = simplexml_load_file(“Automobili.xml“);

foreach ($document->vozilo as $vozilo){
    echo $vozilo->proizvodac . “\r\n\“;
 }
```

Pozivanjem metode *children()* na objektu može se proći kroz čvorove potomaka zadanog čvora. Na isti način na objektu može se koristiti metodu *attributes()* za iteriranje kroz atribute čvora:

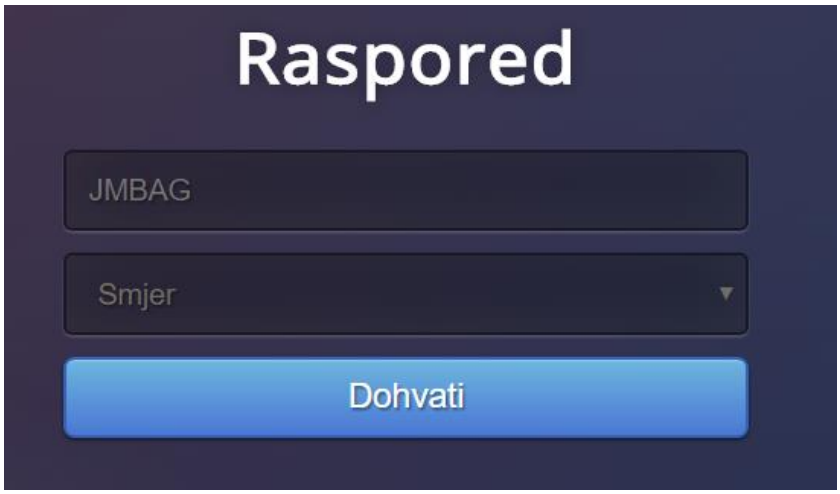
```
$document = simplexml_load_file(“Automobili.xml“);

foreach ($document->prodajni_centar->children() as $node){
    foreach ($node->attributes() as $attribute) {
        echo “{$attribute}\n“;
    }
}
```

4. RJEŠAVANJE DIPLOMSKOG ZADATKA

4.1. PROGRAMSKA IZVEDBA

Ideja za rješavanje diplomskog zadatka bila je da pomoću dobivenih resursa, odnosno dvije URL stranice koje su formatirane u XML obliku, programski napravim funkcionalan raspored nastave i ispita za svakog studenta Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Za rad sam odabrao PHP programski jezik. Iako nisam imao nekog prevelikog predznanja u spomenutom programskom jeziku, znao sam da ima dobru podršku za rad i parsiranje XML datoteka.

Rad sam podijelio u dvije glavne datoteke. Prva datoteka je *index.php*, koja je zapravo stranica za prijavljivanje u kojoj student mora upisati svoj JMBAG te u padajućem izborniku odabrati smjer studija na kojem se nalazi. Nakon upisa i odabira, treba stisnuti tipku dohvati koja nas vodi iduću stranicu na kojoj se prikazuje traženi raspored. Također postoje datoteke koje omogućavaju izvođenje bootstrapa koji je korišten kako bi se tablica ispravno prikazivala na različitim rezolucijama i veličinama ekrana te isto tako i zasebna css datoteka koja služi za stiliziranje određenih dijelova aplikacije, ponajprije početne stranice. Predložak css stranice je preuzet te je dodatno preuređen za potrebe prikaza aplikacije.



Slika 4.1.1 Izgled indeksne stranice

```
<form method="post" action="dohvat.php">
      <input type="text" name="u" placeholder="JMBAG" required="required" />

        <select required name="p">
            <option value='' disabled selected hidden> Smjer <option>
            <option value='1'> PE (Elektrotehnika)</option>
            <option value='2'> PR (Računarstvo)</option>
            <option value='7'> SR (Informatika)</option>
            <option value='8'> SA (Automatika)</option>
            <option value='9'> SE (Elektrotehnika)</option>
            <option value='31'> DEA (Elektroenergetika)</option>
            <option value='32'> DEB </option>
            <option value='33'> DEC </option>
            <option value='34'> DKA (Komunikacije i informatika)</option>
            <option value='35'> DKB </option>
            <option value='36'> DRA (Računarstvo)</option>
            <option value='37'> DRB </option>
            <option value='38'> DRC </option>
            <option value='39'> DRD </option>
        </select>
        <input type="submit" value="Dohvati" class="btn btn-primary btn-block
btn-large"> </input>
    </form>
```

Indeksna stranica je vrlo jednostavna te u zaglavlju poziva css datoteka te određene datoteke potrebne za izvođenje bootstrapa. Napravljena je forma s metodom *post* koja pokreće *dohvat.php* datoteku. U formi se nalazi jedan *input* tipa tekst za JMBAG na koji je stavljen *placeholder* te atribut *name* s vrijednosti u. Napravljen je padajući izbornik tipa *select* u kojemu se nalaze mogući smjerovi koje student mora odabrati. Dodana je i tipka tipa *submit* s nazivom dohvati, te upisom studentovog JMBAG-a i odabirom njegovog smjera potrebno je pričekati da se stavke rasporeda dohvate te prikažu na zaslonu.

Iduća datoteka je *dohvat.php* datoteka je zapravo najbitnija datoteka te se u njoj nalazi cijela funkcionalnost aplikacije i zapravo omogućava njeno izvođenje.

Na početku stranice dodane su dvije funkcije

```
    error_reporting(0);
    set_time_limit(0);
```

Prva je korištena zbog toga što su se na zaslonu izbacivale nepotrebne obavijesti, a druga je korištena kako bi se izbjegla fatalna greška koja se normalno generira u web pretraživaču nakon trideset sekundi čekanja izvršenja skripte zbog velike količine podataka koje je potrebno obraditi.

```
$jmbag = $_POST['u'];
  if (isset($_GET['u'])) $jmbag = $_GET['u'];

$smjer = $_POST['p'];
  if (isset($_GET['p'])) $smjer = $_GET['p'];
```

Pomoću $_POST metode dohvaćamo parametre JMBAG-a i smjera s indeksne stranice, te s obzirom da klikom na link za tjedan prije i tjedan poslije učitavamo ponovno istu stranicu, samo s različitim datumom, moramo napraviti provjeru odnosno postaviti spomenute parametre pomoću $_GET metode, kako bi se znalo za koji smjer i za kojeg točno studenta se traže predavanja idućeg ili prethodnog tjedna. Vrijednosti JMBAG-a i smjera spremamo u istoimene varijable. Isto tako ako sami definiramo parametre u URL-u stranice i postavimo vrijednost smjera i datuma, raspored bi se trebao prikazati jer $_GET metoda dohvaća parametre zadane u URL-u stranice.

```
if (isset($_GET['d1'])) $d1 = $_GET['d1'];
else $d1 = date('Y-m-d');

$dut = date('w', strtotime($d1));
if($dut == 0) $dut=7;
$pon = strtotime($d1)-($dut-1)*24*3600;
```

Idući zadatak je bio izračunati i pronaći način definiranja početnog dana u tjednu odnosno ponedjeljaka. Na isti, već opisani način provjeravamo je li zadan datum u parametru d1 i ako je, njega se sprema u istoimenu varijablu. U slučaju da nije zadan, dohvaća se trenutni datum pomoću funkcije *date* u formatu *YY-MM-DD*. Varijablu d1 šaljemo u funkciju *strtotime* koja pretvara engleski zapis vremena i datuma u vrijednost Unix vremena, te pomoću parametra w koji označava brojčanu reprezentaciju dana u tjednu izdvajamo dane i spremamo u varijablu $dut. U uvjetu otklanjamo mogućnost vrijednosti nule te ju postavljamo na vrijednost sedam koja označava nedjelju jer želimo dobiti brojčane vrijednosti od jedan do šest jer nedjelju moramo ukloniti iz rasporeda. U varijablu $pon spremamo izračun datuma koji se izračunava tako da u funkciju *strtotime* spremamo varijablu $d1 koja predstavlja trenutni datum te se ona oduzima od brojčane oznake dana u tjednu umanjenu za jedan i pomnoženu s konstantom koja predstavlja broj sekundi u jednom danu. Na taj način dobijemo datum ponedjeljka i njega spremamo u varijablu $pon.

```
    $brojac = array();
    for($dan=1;$dan<=6;++$dan) $brojac[$dan]=0;

    $prikaz = array();
    $pocetni_dan = $pon;
```

Definiramo polje za varijablu brojač u polje i u *for* petlji inicijaliziramo brojač za svaki dan jednak nuli. Isto tako definiramo polje za varijablu $prikaz u polje, te varijablu $pocetni_dan spremamo u varijablu $pon.

```
  for($dan=1;$dan<=6;++$dan)
    {
        $datum = date('Y-m-d', $pocetni_dan);
        $pocetni_dan += 24*3600;

        $xmlobjekt =
simplexml_load_file("http://mrkve.etfos.hr/api/raspored/index.php?date=$datum
");
```

Glavna *for* petlja koja kreće od prvog dana ponedjeljka sve do šestog dana subote i kroz svaku iteraciju petlje povećava vrijednost dana za jedan. Na početku je definiran datum u kojemu predajemo varijablu početnog dana da svakog puta dohvaća podatke od ponedjeljka. Kroz iteraciju isto tako početni dan uvećavamo za jedan dan te u objekt $xmlobjekt pretvaramo XML dokument s varijablom datuma na kraju koju dohvaća prije u kodu.

```
foreach($xmlobjekt->stavkaRasporeda as $sr)
        {
            $brgr = $sr->grupastudenata['idgrupe'];

            if($brgr == '') continue;
            if($sr->smjer['idsmjer'] != $smjer) continue;

            $xmlobjekt2 =
simplexml_load_file("http://mrkve.etfos.hr/api/raspored/grupa.php?idgrupe=$br
gr");

            $slusa = false;
            foreach($xmlobjekt2->studenti->student as $st)
                {
                if($st->jmbag == $jmbag) $slusa = true;
                }
            if($slusa == false) continue;

            $prikaz[$dan][$brojac[$dan]] = $sr;
            $brojac[$dan]++;
        }
    }
```

Dostavljeni XML dokument ima puno stavki, pa je zadatak proći kroz svaku stavku i prvo provjeriti postoji li broj grupe, te ako ga nema nastavlja se dalje s kodom. Nadalje se provjerava smjer te ako je različit od smjera koji je poslan preko indeksne stranice, također nastavljamo dalje s kodom jer se ne pronalazi ništa. U $xmlobjekt2 učitavamo datoteku koja u URL-u ima zadan *id* grupe. Definiramo varijablu $slusa i stavljamo ju u početku na *false* te kroz petlju

tražimo studente odnosno podudaranje jmbaga studenta koji se nalazi u određenoj grupi. Ako je on pronađen, znači da student sluša taj predmet, a ako ga nema, kod se nastavlja dalje. Kad smo provjerili sve, ako je rezultat *false*, znači da student nije u toj grupi. Ako je student pronađen u grupi, onda ga dodajemo u varijablu prikaz koja se ponaša kao matrica te zasebno spremamo stavke za sve dane te brojači služe kako bi se te stavke za svaki dan prebrojale. Petlje se odrađuju za sve dane u tjednu osim nedjelje i vrijednosti se spremaju u varijablu $prikaz koja će kasnije služiti za prikazivanje svih spremljenih stavki rasporeda.

```
$lijevo = date('Y-m-d', $pon - 7*24*3600);
$desno = date('Y-m-d', $pon + 7*24*3600);
```

Definirane varijable koje će se kasnije koristiti u linkovima za prikaz idućeg i prethodnog tjedna. Konstanta 7*24*3600 predstavlja vremensko razdoblje od tjedan dana.

```
echo '<div class="bs-example">
    <div class="table-responsive">
    <table class="table">
    <tr>
        <td width="50%" colspan="3" style="text-align:left; border-top:none">
<a href="dohvat.php?u='.$jmbag.'&p='.$smjer.'&d1='.$lijevo.'"><img
style="width:20px;" src="strelical.png"></a>
        </td>
        <td colspan="3" style="text-align:right; border-top:none"> <a
href="dohvat.php?u='.$jmbag.'&p='.$smjer.'&d1='.$desno.'"><img
style="width:20px;" src="strelicar.png"></a>
        </td>
    </tr>
    </table>
```

Definirane su klase za tablicu te u kutevima prvog gornjeg reda tablice smještamo linkove za lijevo i za desno čijim se klikom dohvaća raspored prethodnog ili idućeg tjedna.

```
    <table class="table table-bordered">
    <tr>
       <th width="16%"> ponedjeljak<br>'. date('d.m.Y.', $pon) .' </th>
       <th width="16%"> utorak<br>'. date('d.m.Y.', $pon+24*3600) .' </th>
       <th width="16%"> srijeda<br>'. date('d.m.Y.', $pon+2*24*3600) .' </th>
       <th width="16%"> četvrtak<br>'. date('d.m.Y.', $pon+3*24*3600) .'</th>
       <th width="16%"> petak<br>'. date('d.m.Y.', $pon+4*24*3600) .' </th>
       <th width="16%"> subota<br>'. date('d.m.Y.', $pon+5*24*3600) .' </th>
    </tr>';
```

U idućem dijelu koda, u tabličnom redu dodajemo tablična zaglavlja za svaki dan u tjednu te ponedjeljak prikazujemo u formatu datum, mjesec, godina i dohvaćamo varijablu $pon u kojoj imamo izračun datuma ponedjeljka i njega prikazujemo. Za izračun ostalih dana u tjednu dodajemo konstante ovisno o danu koji se prikazuje.

```
for($dan=1;$dan<=6;++$dan)
  {
   echo "<td>";
   for($st=0; $st<$brojac[$dan];++$st)
     {
     $pr=$prikaz[$dan][$st];
     echo $pr->pocetak ."-" . $pr->kraj . "<br>";
     echo $pr->predmet . " " . $pr->odradjeno . "/" . $pr->planirano ."<br>";
     echo $pr->prostorija . "<br>";
     if ( $pr->vrstanastave['tip'] == 1) echo "PR"  . "<br>" . "<br>";
     if ( $pr->vrstanastave['tip'] == 3) echo "AV" . "<br>" . "<br>" ;
     if ( $pr->vrstanastave['tip'] == 4) echo "LV" . "<br>" . "<br>" ;
     if ( $pr->vrstanastave['tip'] == 5) echo "KV" . "<br>" . "<br>" ;
     if ( $pr->vrstanastave['tip'] == 7) echo "Ispit" . "<br>" . "<br>" ;
     }
    echo "</td>";
   }
```

Na kraju imamo dio koda za ispis pomoću *for* petlje koja prolazi kroz svih šest dana te pomoću brojača koji mora biti matričnog tipa kako bi prikazao stavke za sve dane i napravio zbroj koliko ih uopće i ima. Ako je u brojaču nula stavki, prikaz se neće ni jednom izvršiti i u tablici će za taj određeni dan biti prazan kvadratić.

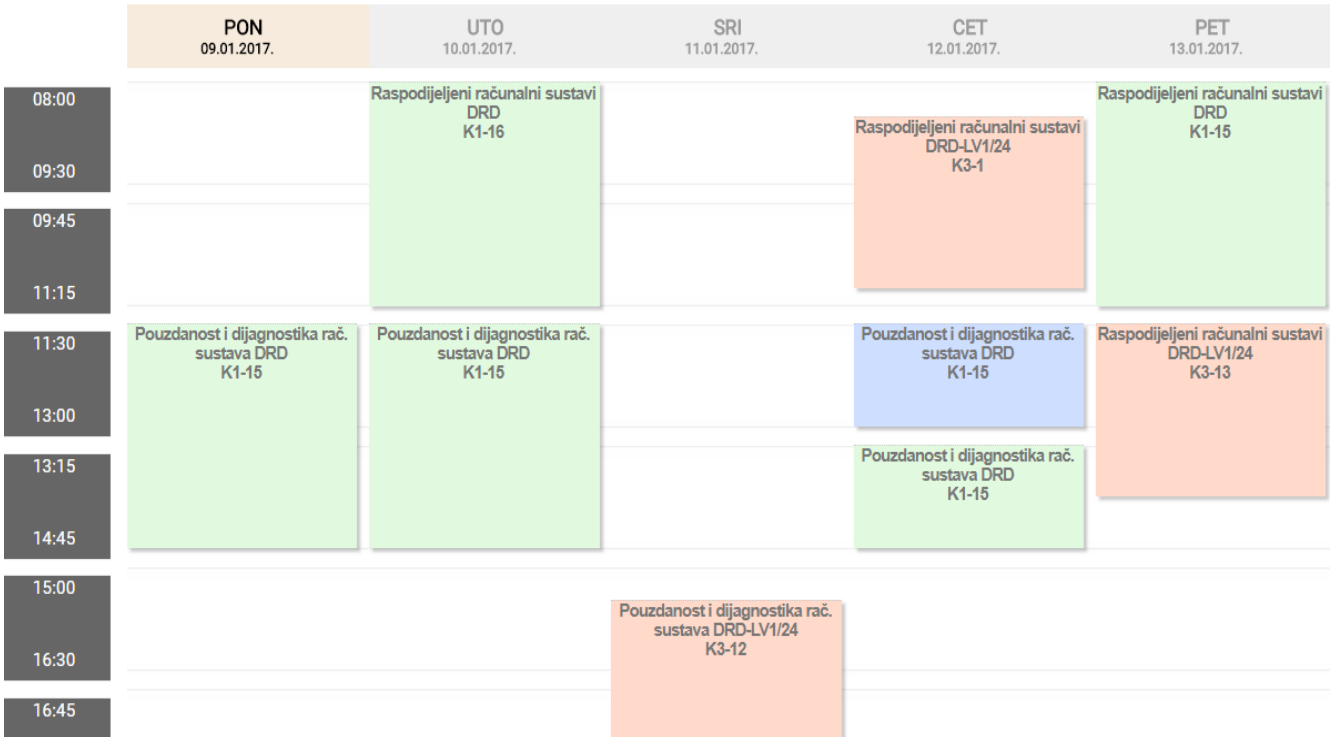
Varijabla $prikaz je zapravo matrica u kojoj se prikazuje sve stavke rasporeda za svaki dan. Iz nje možemo dohvaćati i prikazivati razne podatke koje se nalaze u jednom od dva glavna XML objekta kao što su: početak predavanja, završetak, prostorija, profesor itd. U tablici je dodan ispis vrste predavanja te broj odrađenih sati i ukupni broj sati.

4.2. USPOREDBA RASPOREDA

Glavna razlika između službenog rasporeda koji se nalazi na stranicama fakulteta i moje verzije rasporeda je u vremenu dohvaćanja stavki rasporeda i prikazu istih. Zbog dohvaćanja podataka iz dvije XML datoteke koje sadržavaju velik broj informacija te zbog višebrojnih iteracija nad tim podacima, vrijeme dohvaćanja i prikaza stavki rasporeda može potrajati i do pola minute. To je glavni nedostatak aplikacije kojeg sam u suradnji s profesorom pokušao riješiti, te je pokušaj detaljnije opisan u idućem poglavlju. S obzirom da je moja web aplikacija minimalistički uređena, daljnji napredak vidim i u tome. Velika prednost moje verzije rasporeda je u tome što se studentu prikazuje jedino raspored kolegija koje student sluša, odnosno u rasporedu se ne prikazuju nebitne stavke za studenta te kolegiji koje student ne pohađa. Na idućim slikama vidimo usporedbu oba rasporeda.

ponedjeljak 09.01.2017.	utorak 10.01.2017.	srijeda 11.01.2017.	četvrtak 12.01.2017.	petak 13.01.2017.
11:30-14:45 Pouzdanost i dijagnostika rač. sustava 40/45 K1-15 PR	08:00-11:15 Raspodijeljeni računalni sustavi 40/45 K1-16 PR 11:30-14:45 Pouzdanost i dijagnostika rač. sustava 44/45 K1-15 PR	15:30-18:00 Pouzdanost i dijagnostika rač. sustava 12/15 K3-12 LV	08:30-11:00 Raspodijeljeni računalni sustavi 9/15 K3-1 LV 11:30-13:00 Pouzdanost i dijagnostika rač. sustava 14/15 K1-15 AV 13:15-14:45 Pouzdanost i dijagnostika rač. sustava 46/45 K1-15 PR	08:00-11:15 Raspodijeljeni računalni sustavi 44/45 K1-15 PR 11:30-14:00 Raspodijeljeni računalni sustavi 12/15 K3-13 LV

Slika 4.2.1 Moja verzija rasporeda



Slika 4.2.2 Službeni raspored

4.3. UBRZAVANJE KODA

Zbog velike količine podataka i velikog broja iteracija nad podacima XML datoteka, vrijeme potrebno da se dohvate stavke rasporeda i prikažu u tablici varira između dvadeset i trideset sekundi što predstavlja jedan od najvećih nedostataka rasporeda. Uz pomoć profesora, koji je stvorio novi resurs iz postojećih podataka, parsirao je podatke tako da stvori novu XML datoteku koja će služiti da zadavanjem odgovarajućeg datuma i JMBAG-a studenta, ispiše identifikacijske brojeve svih predmeta koje student ima u tekućoj akademskoj godini.

```
http://mrkve.etfos.hr/api/studentpredmet/index.php?date=2017-09-
11&jmbag=0165056200

1617633 1617074 1617078 1617319 1617002 1617001 1617621
```

Ispis *id-ja* svih predmeta zadanog studentovog JMBAG-a te datuma.

```
$xmlsp =
simplexml_load_file("http://mrkve.etfos.hr/api/studentpredmet/index.php?date=
$datum&jmbag=$jmbag");

$sp=array();

$i=0;
foreach ($xmlsp as $predmet => $pr) {
    $sp[$i] = (string)$pr;
    $i++;
}
```

Smjestili smo predmete u novi objekt $xmlsp te smo definirali polje u koje ćemo spremati pronađene predmete te se pomoću brojača ispisuju predmeti koje student sluša.

```
if (!in_array((string)$sr->predmet['mrkveid'], $sp)) continue;
```

Kad se sve stavke rasporeda dohvate, provjeravamo ako predmet nije u ovome skupu onda se ni ne moraju dohvaćati grupe te se program nastavlja dalje s izvođenjem.

Unatoč dodanom kodu vrijeme dohvata i prikaza rasporeda nije se značajno promijenilo pa sam se za krajnju verziju rada odlučio ostaviti onu verziju bez dijela koda s pokušajem ubrzavanja aplikacije.

5. ZAKLJUČAK

Zadatak diplomskog rada bio je napraviti programsku izvedbu rasporeda nastave za sve studente Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Zadatak je napravljen u PHP programskom jeziku uz korištenje HTML-a, CSS-a te bootstrap okvira.

Cilj rada bio je iz danih resursa, odnosno dvije XML datoteke, obraditi i parsirati podatke te generirati stavke rasporeda i prikazati ih u tablici. Na početku rada spomenut je pojam rasporeda te da je raspored osnovni način upravljanja vremenom. Studentski raspored je tablica s koordiniranjem četiri elementa: studenti, profesori, prostorije i vrijeme izvođenja nastave. Napravljen je uvod u programske jezike te sam najveću pažnju posvetio PHP programskom jeziku jer sam pomoću njega uspješno napravio diplomski rad. Spomenuta je povijest programskog jezika te je objašnjen način instalacije PHP-a na računalo uz pomoć XAMPP platforme čijom instalacijom dolazi Apache, MySQL, PHP i brojne druge aplikacije korisne za razvijanje dinamičkih web stranica. Također je objašnjen način instalacije same platforme. Spomenuta je glavna sintaksa PHP jezika te su u tablici navedene sve ključne riječi. Objašnjeni su nizovi i neki glavni pojmovi vezani za sam jezik te je ukratko spomenuta mogućnost objektno orijentiranog programiranja. Spomenute su sve tri mogućnosti parsiranja XML-a pomoću PHP-a te je objašnjen SimpleXML način parsiranja koji je korišten u izradi programskog rješenja. Navedeni su primjeri i objašnjen je način parsiranja podataka iz datoteka. Na kraju, detaljno je opisana cijela programska izvedba diplomskog zadatka te su priložene slike izgleda aplikacije. Navedena je usporedba moje izvedbe rasporeda s rasporeda koji se nalazi na službenim stranicama fakulteta. Na samom kraju rada, objašnjen je i pokušaj ubrzavanja dohvata svih potrebnih stavki te njihov prikaz.

Aplikacija je u potpunosti funkcionalna te su implementirani svi elementi koje raspored mora sadržavati. Uz profesorovu pomoć aplikacija je uspješno napravljena te je namijenjena svim studentima Fakulteta elektrotehnike, računarstva i informacijskih tehnologija. Svi studenti uz upis svog JMBAG-a te odabirom svoga smjera mogu pristupiti vlastitom rasporedu. Na aplikaciji postoje moguće brojne nadogradnje u budućnosti. Glavni nedostatak rasporeda je već spomenuto vrijeme potrebno da se dohvate sve stavke koje može potrajati i do dvadeset pet sekundi. Sama aplikacija je napravljena minimalističkom uporabom dizajnom u čemu također vidim moguću nadogradnju i poboljšanje u nekim budućim verzijama aplikacije.

LITERATURA

[1] https://en.wikipedia.org/wiki/School_timetable [10.6.2017]

[2] R. Lerdorf, K. Tatroe, P. MacIntyre, Programiranje PHP, Dobar Plan, 2015, Zagreb

[3] https://en.wikipedia.org/wiki/PHP [21.6.2017]

[4] https://en.wikipedia.org/wiki/XAMPP [15.7.2017]

[5] https://en.wikipedia.org/wiki/XML [23.7.2017]

[6] https://www.w3schools.com/php/php_ref_simplexml.asp [5.8.2017]

[7] R. Lerdorf, K. Tatroe, P. MacIntyre, Programiranje PHP, Dobar Plan, 2015, Zagreb, 284 str

SAŽETAK

Cilj ovoga rada bio je napraviti web aplikaciju u PHP programskom jeziku namijenjenu studentima Fakulteta elektrotehnike, računarstva i informacijskih tehnologija. Rasporedu se pristupa upisom studentovog JMBAG-a te odabirom smjera na kojem se nalazi. Na početku rada napravljen je uvod u programske jezike s najvećim osloncem na PHP. Objašnjena je XAMPP platforma te koraci instalacije. Navedena je osnovna sintaksa PHP-a te spomenute tri mogućnosti parsiranja XML dokumenta te je opisan SimpleXML dodatak koji je korišten u izradi rada. Na kraju, detaljno je opisana programska izvedba same web aplikacije.

Ključne riječi: raspored, PHP, XAMPP, parsiranje, XML, SimpleXML, web aplikacjia

STUDENT SCHEDULE

ABSTRACT

The main task of this paper was to make web application in PHP programming language mainly intended for students of the Faculty of Electrical Engineering, Computing and Information Technology in Osijek. The schedule can be accessed by entering the student JMBAG number and by selecting the course in which the student is enrolled. At the beginning of the paper, an introduction to the programming languages with the greatest support for PHP was made. The XAMPP platform and installation steps were explained. The basic syntax of PHP was mentioned and three possibilities of parsing an XML document were mentioned and so was the SimpleXML extension which was used in work.

Key words: schedule, PHP, XAMPP, parsing, XML, SimpleXML, web application

ŽIVOTOPIS

Andi Bašić rođen je 31.10.1993 godine u Novoj Gradiški. U istom gradu odrasta i živi te se 2008. godine upisuje u Opću gimnaziju. Srednju školu završava s odličnim uspjehom te 20012. godine upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku. 2015. godine završava preddiplomski studij te upisuje diplomski studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku - smjer Informacijske i podatkovne znanosti. Od jezika vrlo dobro poznaje engleski te je upoznat s osnovama njemačkog jezika. Praksu je odradio u Edunovi te najviše zanimanja pronalazi u PHP programskom jeziku.

PRILOZI

Cijeli programski kod priložen je na CD-u.