Internet aplikacija za razmještanje studenata na pismenom ispitu

Golubović, Josip

Undergraduate thesis / Završni rad

2017

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:609939

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Stručni studij

INTERNET APLIKACIJA ZA RAZMJEŠTANJE STUDENTATA NA PISMENOM ISPITU

Završni rad

Josip Golubović

Osijek, 2017.

SADRŽAJ

1. UVOD 1
1.1 Zadatak završnog rada 1
2. PRIMJENJENE TEHNOLOGIJE 2
2.1 PHP 2
2.2 XAMPP 4
2.3 MySQL 5
3. REALIZACIJA APLIKACIJE 6
3.1 Princip pisanja PHP skripti 6
3.2 Princip rada s bazama 7
3.3 Primjer predmeta i popis grupa iz resursa 10
3.4 Prikaz koda za prostoriju K2-11 14
3.5 Raspored studenata za jednu ispitnu grupu 17
3.6 Raspored studenata za dvije ispitne grupe 19
3.7 Raspored studenata za tri ispitne grupe 20
4. ZAKLJUČAK 22
LITERATURA 23
SAŽETAK 24
ABSTRACT 25
ŽIVOTOPIS 26
PRILOG Izvorni kod aplikacije 27

1. UVOD

Cilj ovog rada je aplikacija koja obavlja posao razmještanja studenata prije pismenog ispita i time olakšava posao profesorima i uštedi vrijeme studentima prije ispita. Prvi korak pri izradi je odabrati jezik s kojim će se izraditi aplikacija. S obzirom da se radi o internet aplikaciji zbog jednostavnosti izrade odabran je skriptni jezik PHP, te besplatan sustav za upravljanje bazom podataka MySQL. MySQL baze su optimizirane, stabilne, imaju dobro definirane module i ekstenzije te podršku od brojnih programskih jezika. Drugi korak je obilazak učionica i izrada skica sjedećih mjesta. U radu je posebna pažnja posvećena standardima koji su određeni prema zahtjevima norme ISO 9001:2008. kojom je fakultet osigurao kvalitetu visokog obrazovanja. U drugom poglavlju opisan je skriptni jezik pomoću kojeg je aplikacija izrađena, te su postavljeni osnovni primjeri koda programskog jezika i baza podataka. Trećim poglavljem završnog rada predstavljena je problematika pisanja izvornog koda te je obuhvaćena realizacija aplikacije.

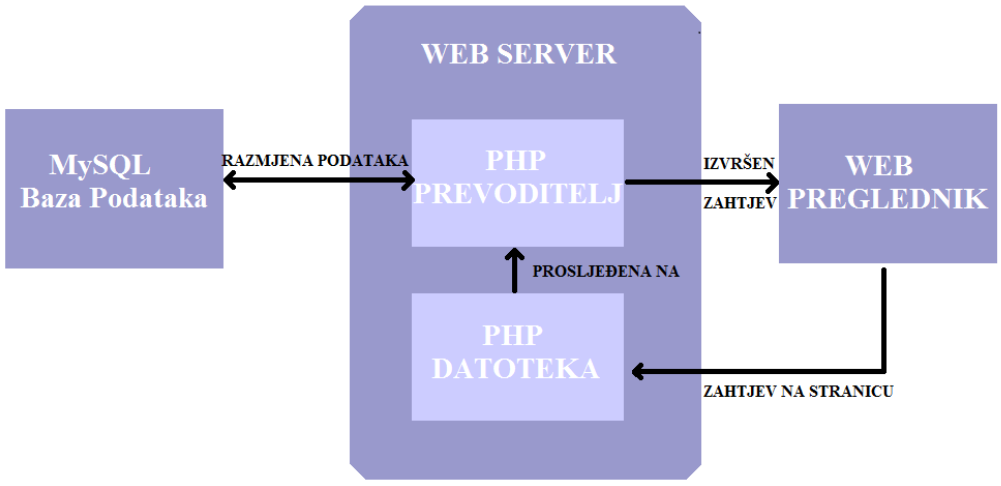
1.1 Zadatak završnog rada

U ovom završnom radu potrebno je napraviti internet aplikaciju za razmještanje sjedećih mjesta za studente na pismenom ispitu. Aplikacija uključuje izradu modela učionice s proizvoljnim brojem stolica i prolaza između njih.

2. PRIMJENJENE TEHNOLOGIJE

2.1 PHP

PHP je skriptni jezik namijenjen za izradu dinamičnog web sadržaja. Izvodi se na strani servera. Zbog svoje sintakse koju je naslijedio od programskog jezika C i zbog svoje jednostavnosti stekao je veliku popularnost. Jezik se ima mogućnost za objektno orijentirano programiranje, te ima mogućnost za čisto-proceduralno programiranje. Prvi alat po nazivu PHP/FI je napisan 1995. godine, a napisao ga je Rasmus Lerdorf za osobne potrebe (eng. Personal Home Page Tools), odakle i skraćenica PHP. Ova verzija je imala samo neke zajedničke stvari sa današnjim PHP-om. Ubrzo nakon službenog objavljivanja verzije PHP/FI zamjenjuje ga PHP 3.0 ime je promijenjeno u PHP bez dodatka „FI“, a i značenje skraćenice se izmijenilo i predstavlja "PHP: preprocesor hipertekst" (PHP: Hypertext Preprocessor). Ova verzija PHP-a predstavlja verziju koja je slična današnjim verzijama PHP-a. Kreirali su ga izraelski programeri Zeev Suraski i Endi Gutmans. Glavna osobina PHP je njegova proširivost. Osnovna i glavna upotreba PHP-a je u programiranju dinamičnih stranica na Internetu. [1]



Slika 2.1. Redoslijed izvršavanja PHP skripte

- klijent (korisnik Interneta koji koristi neki internet pretraživač) zahtjeva PHP stranicu sa servera

- server prosljeđuje zahtjev servisu za web (program web-server na serveru)

- web-server prepoznaje da se traži PHP datoteka

- ne šalje njegov sadržaj klijentu, nego ga izvršava kao program pomoću PHP modula

- izlazni tekst programa (standardni izlaz) se šalje klijentu kao rezultat zahtjeva

- klijent prepoznaje vrstu rezultata (HTML kod, slika, PDF sadržaj, arhiva itd.)

-rezultat se prikazuje klijentu na odgovarajući način.

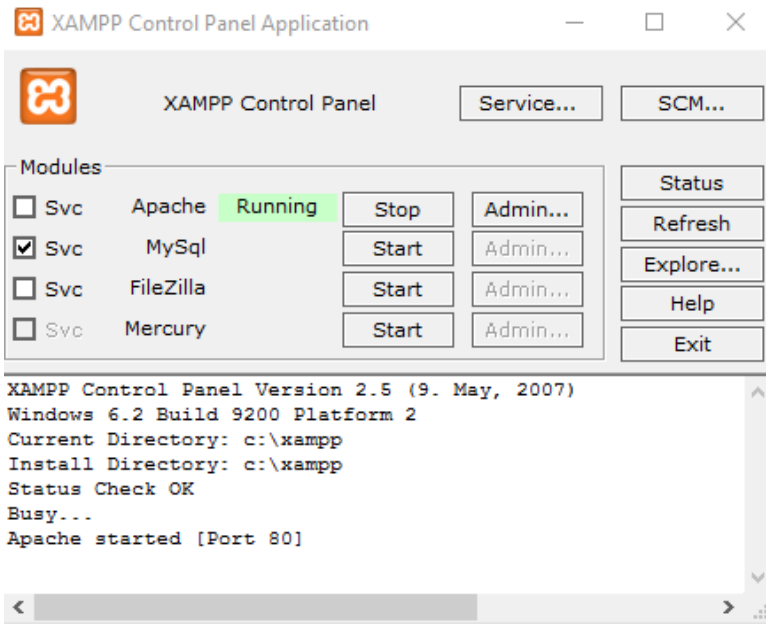
Za rad PHP skripti kao i za rad stranice koja koristi PHP kod, web server mora podržavati PHP. Osim nekih besplatnih, većina komercijalnih servera podržava PHP. Ako želite isprobati PHP skripte na vašem računalu morate skinuti i instalirati program Xampp. Slika 2.2 prikazuje Logo alata i tehnologija koje su korišteni u ovom radu.



Slika 2.2. Alati korišteni u izradi aplikacije

2.2 XAMPP

Za rad s PHP skriptama prvo je potrebno preuzeti besplatan program Xampp. Nakon preuzimanja slijedi jednostavna instalacija. Sa Xampp-om smo instalirali platforme kao što su Apache, MySQL, PHP, MyAdmin i još mnoge druge. (Sl 2.3.)



Slika 2.3. Platforma Xampp 1.7.0 sa svojim alatima

Pokretanjem aplikacija Apache i MySQL od računala pravimo virtualni lokalni server preko kojeg se pozivaju i obrađuju PHP skripte. Za pisanje PHP skripti mogu se koristiti razni tekst editori kao što su Sublime, Notepad ++, Visual Studio Code itd Spomenuti tekst editori su otvorenog koda, dakle svako ga može prilagoditi na sebi svojstven način. U ovom radu koristi se Visual Studio Code, iz razloga što ima jako puno proširenja koji mogu pomoći pri pisanju koda, a naravno i dizajnom je jako privlačan i oku ugodan te jednostavan.

2.3 MySQL

SQL (Structured Query Language) je programski jezik koji služi za izradu baze podataka, te svim daljnjim operacijama potrebnim za rad s bazama podataka. U ovom završnom radu je korišten za izradu baze za spremanje podataka s kojima bi kasnije manipulirali pomoću PHP skripti. MySQL sustav izabran je iz razloga što je besplatan, open source sustav za upravljanje bazom. Također, MySQLbaza vrlo je optimizirana i stabilna

Glavne značajke ove baze su:

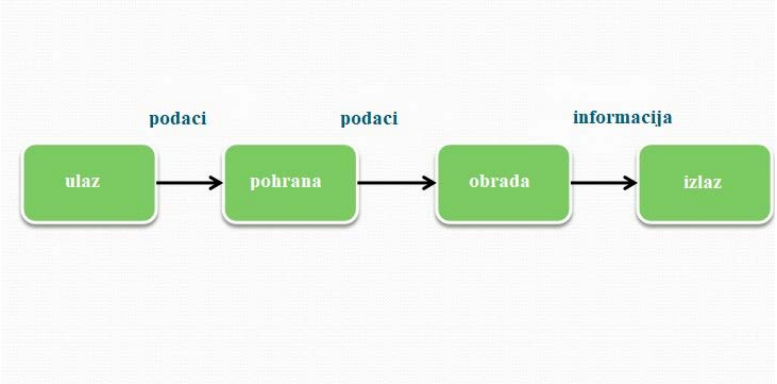
- u slučaju pada sustava ili nestanka električne energije podaci ostaju u bazi
- jedna datoteka za cijelu bazu podataka
- korisničko sučelje je jednostavno
- open source kod

Kod baza podataka potrebno je voditi računa o ulaznim i izlaznim informacijama.

Ulazne informacije u ovoj aplikaciji su:

- prostorije opremljene za pisanje ispita,
- termin pismenog ispita,
- broj sjedećih mjesta pojedine prostorije,
- broj grupa ispita i
- broj studenata koji su prijavili pismeni ispit.

Slika 2.4 prikazuje obradu ulaznih podataka te prikaz informacije na monitoru.



Slika 2.4. Obrada podataka

3. REALIZACIJA APLIKACIJE

3.1 Princip pisanja PHP skripti

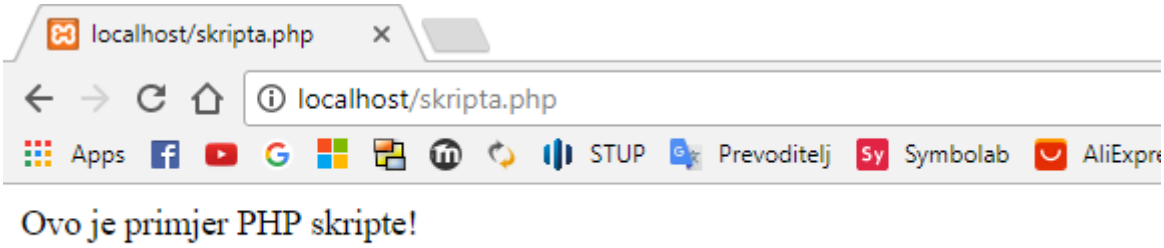
Slika 3.1 prikazuje PHP skriptu koja će na Internet pregledniku ispisati određeni tekst. Prilikom pozivanja napisane skripte mora biti pokrenut program Xampp i njegova aplikacija Apache. Napisana skripta sprema se u datoteku „htdocs“ koja se nalazi u datoteci gdje je instaliran program Xampp. Najčešće je to „C:\xampp\htdocs“. Skripta se pokreće tako da se u Internet preglednik napiše „localhost/skripta.php" (prethodno napisani kod spremljen je pod nazivom „skripta.php“).

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "Ovo je primjer PHP skripte!";
?>

</body>
</html>
```

Slika 3.1. Primjer koda napisanog u tekst editoru Visual Studio Code



Slika 3.2. Primjer PHP skripte na Internet pregledniku

3.2 Princip rada s bazama

```php
<?php

/*Kreiranje baze podataka, koju možemo ubaciti na sva mjesta na kojima nam
zatreba upravljanje sa bazom: */

session_start();

$dbhost = "localhost"; // ovo je server host
$dbname = "aplikacija"; // ime baze
$dbuser = "root"; // korisničko ime
$dbpass = ""; // lozinka

mysql_connect($dbhost, $dbuser, $dbpass) or die("MySQL Error: " .
mysql_error());
mysql_select_db($dbname) or die("MySQL Error: " . mysql_error());

?>
```

Slika 3.3. PHP kod za kreiranje baze

```sql
CREATE TABLE IF NOT EXISTS `student` (
  `student_id` int(11) NOT NULL AUTO_INCREMENT,
  `student_ime ` varchar(15) NOT NULL,
  `student_prezime` varchar(15) NOT NULL,
  `student_JMBAG` varchar(255) NOT NULL,
  PRIMARY KEY (`student_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Slika 3.4. SQL kod za kreiranje tablice

Slika 3.3 prikazuje kod za kreiranje bazu podataka. Naziv baze je „aplikacija“ korisničko ime je root, a lozinka nije postavljena zbog jednostavnijeg testiranja aplikacije. Naravno nakon što aplikacija bude u uporabi korisničko ime i lozinka mogu biti postavljeni zbog sigurnosti baze podataka. Navedeni podaci u kodu napisani su kao varijable zbog jednostavnijeg izmjenjivanja.

Slika 3.4 prikazuje kreiranje jednostavne tablice napisane u SQL programskom jeziku.

Tablica „studenti“ sadrži parametre:

id – koji ne smije biti nula i automatski se povećava za 1

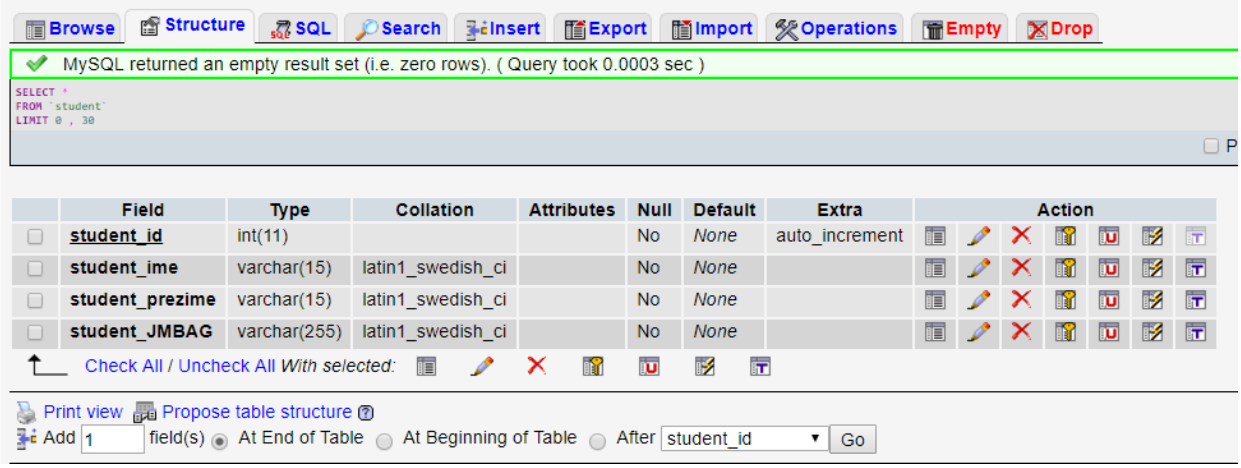
ime – u našem slučaju studenta koji je prijavio ispit

prezime – također u našem slučaju studenta

JMBAG – jedinstveni matični broj akademskog građanina

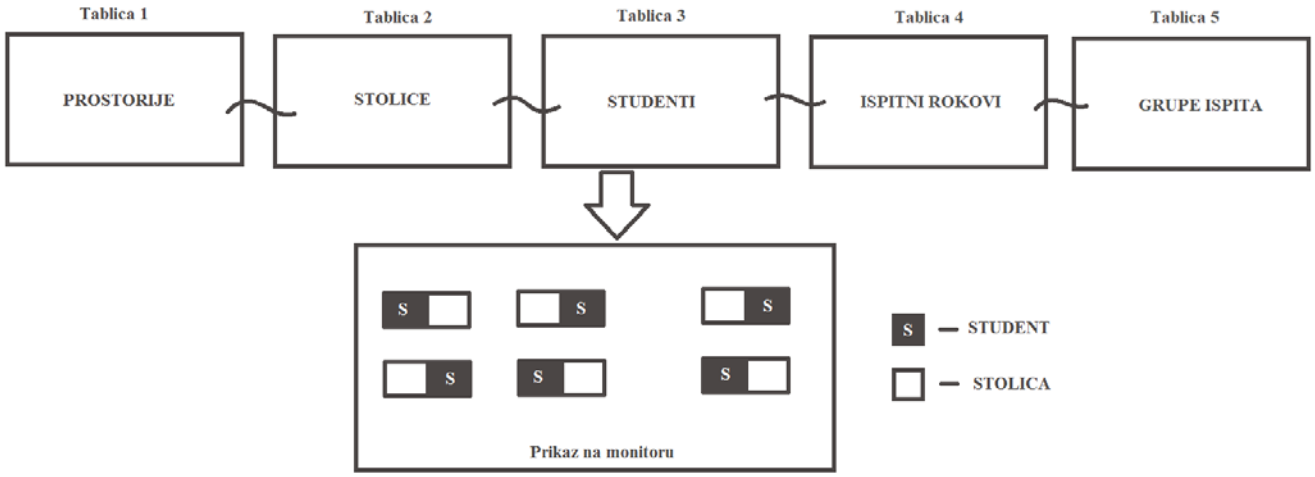
PRIMARY KEY – sam postavio da bude id studenta

Sučelje phpMyadmin možemo vidjeti na slici 3.5.



Slika 3.5. Izgled baza podataka u phpMyadmin sučelju

Navedene informacije aplikacija koristi iz resursa s fakulteta. Studenti koji prijavljuju ispit automatski se upisuju u bazu koja je povezana sa bazom u koju su smještene prostorije odnosno sjedeća mjesta u prostorijama. Sa studentima PHP skripta nastavlja raditi te se oni prikazuju na zaslonu raspoređeni na određena mjesta u ispitnoj dvorani. Dakle rezultat aplikacije prikazuje studente na odgovarajućim mjestima u ispitnoj dvorani .Prva tablica sadržava prostorije koje su opremljene za pisanje ispita, druga tablica broj stolica na koje ćemo poslije razmještati studente. Studente, ispitne rokove i grupe ispita aplikacija će dobiti iz resursa ISVU sustava. Tablice će biti povezane preko jmbag-a studenta i id brojeva prostorija, stolica, kolegija, ispitnih rokova i ispitnih grupa.



Slika 3.6. Princip rada aplikacije

3.3 Primjer predmeta i popis grupa iz resursa

Kada su ispitni rokovi određeni, algoritam traži broj studenata koji pristupaju pismenom ispitu, da bi se mogla odabrati odgovarajuća dvorana za pisanje ispita. Dvorana se bira prema broju sjedećih mjesta i broju ispitnih grupa koje određuje profesor. Slijedi primjer resursa s podacima za predmet inženjerska grafika.

Ova skripta sadrži informacije kao što su ime kolegija, smjer studija, akademska godina, koliko je redovnih koliko vanrednih studenata i grupe studenata.

Aplikacija iz prikazane tablice povlači grupe studenata tj. broj studenata koji pohađaju određeni kolegij.

U kodu obratiti pažnju na grupe tip=1 jer su to predavanja i sadrže sve studente tj. njihov najveći mogući broj koji može izaći na ispit pa prema tome tražimo prostoriju koja ima taj broj sjedećih mjesta za izvođenje ispitnog roka.

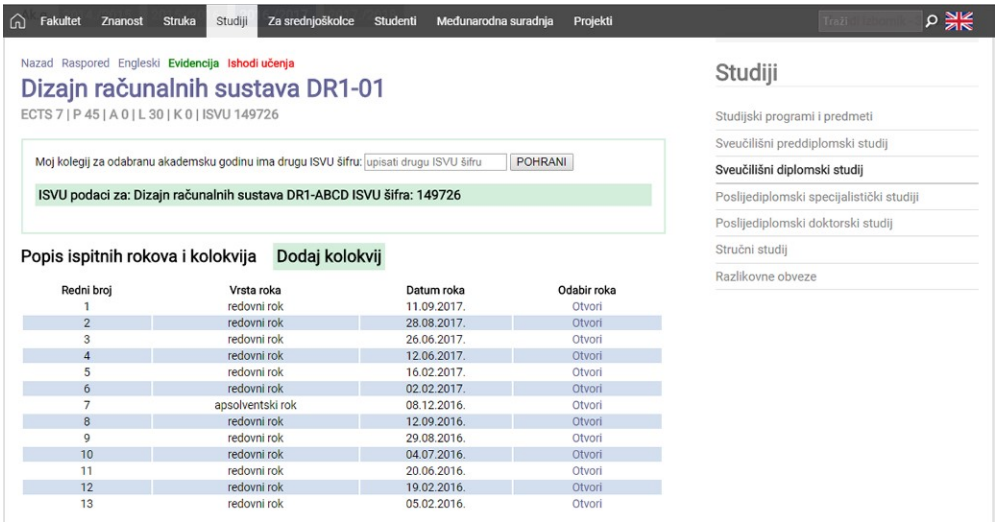
```
<predmet id="1718326" isvuid="37107" loomenid="3536">
 <sifra>P105</sifra>
 <ime>Inženjerska grafika i dokumentiranje</ime>
 <imeen>Engineering Graphics and Documentation</imeen>
 <tip idtip="0">redoviti</tip>
 <studij idstudij="0">Preddiplomski studij</studij>
 <smjer idsmjer="1">Sveučilišni preddiplomski studij elektrotehnike, izb. blok
EE</smjer>
 <akademska_godina>2017./2018.</akademska_godina>
 <nastavna_godina>1</nastavna_godina>
 <semestar>1</semestar>
 <ects>3</ects>
 <brojstudenata>0</brojstudenata>
 <grupe>
 <grupa tip="1" idgrupe="6378">PE</grupa>
 <grupa tip="1" idgrupe="6379">PR</grupa>
 <grupa tip="5" idgrupe="6380">PE-KV1/24</grupa>
 <grupa tip="5" idgrupe="6381">PE-KV2/24</grupa>
 <grupa tip="5" idgrupe="6382">PE-KV3/24</grupa>
 <grupa tip="5" idgrupe="6383">PE-KV4/24</grupa>
 <grupa tip="5" idgrupe="6384">PE-KV5/24</grupa>
 <grupa tip="5" idgrupe="6385">PR-KV1/24</grupa>
 <grupa tip="5" idgrupe="6386">PR-KV2/24</grupa>
 </grupe>
```

Slika 3.7. Primjer grupe iz resursa za kolegij inženjerska grafika

```
<?xml version="1.0" encoding="Windows-1250"?>
<grupaStudenata idgrupe="6378">
<brojGrupe>1</brojGrupe>
<ime>DEB</ime>
<opis></opis>
<predmet isvuid="37107" semestar="1"> Inženjerska grafika i
dokumentiranje/predmet>
<studenti brojStudenata="7">
<student>
<jmbag>0165043258</jmbag>
<ime>Ime 1. studenta</ime>
<prezime>Arambašić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054302</jmbag>
<ime> Ime 2. studenta </ime>
<prezime>Arić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0246046127</jmbag>
<ime> Ime 3. studenta </ime>
<prezime>Bančić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054440</jmbag>
<ime> Ime 4. studenta </ime>
<prezime>Čabrajac</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165051251</jmbag>
<ime> Ime 5. studenta </ime>
<prezime>Čurčić</prezime>
<redovni>1</redovni>
</student>
<student>
```

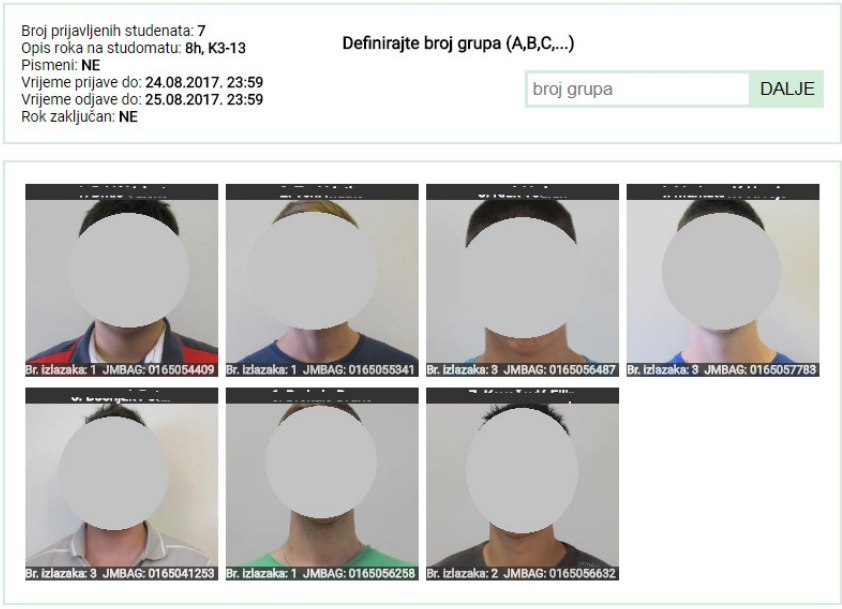
Slika 3.8. Prikaz dio php skripte „studenati“ iz grupe kolegija inženjerska grafika

Slika 3.9. Prikazuje sučelje koje sadrži datum rokova pisanih ispita 2016/2017 godine. Kada su određeni datumi rokova, slijedi upis studena u ISVU sustav tj. bazu, iz kojeg će se poslije studenti dodjeljivati određenim ispitnim dvoranama gdje će se pisati ispiti.



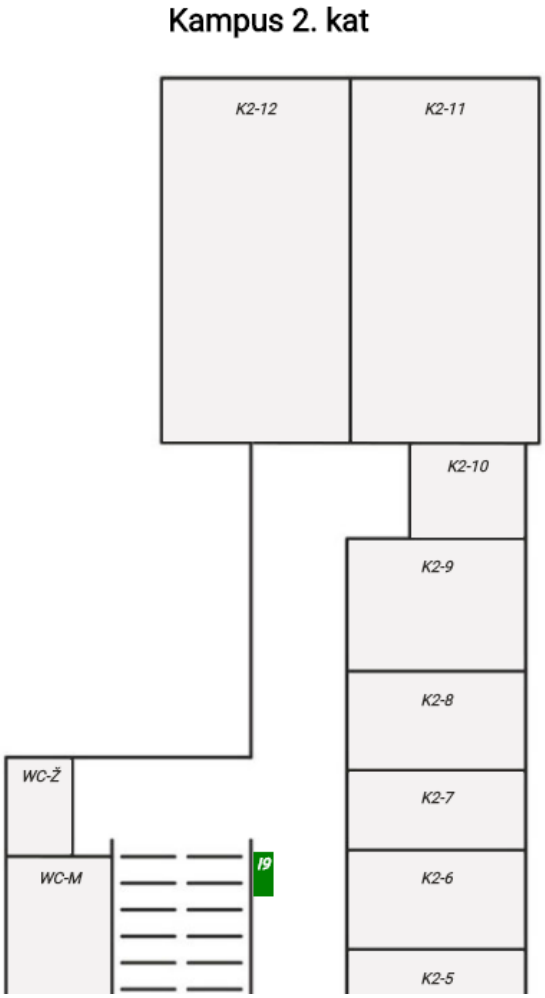
Slika 3.9. Prikaz sučelja s importom ISVU rokova

Slika 3.10. prikazuje sučelje koje sadrži informacije o studentima koji su prijavili ispit iz pojedinog kolegija za određeni rok. Sadrži informacije kao što su broj prijavljenih studenata, broj prostorija u kojoj se piše ispit i broj ispitnih grupa. Ovdje profesori određuju koliko ispitnih grupa će biti na pojedinim kolegijima. Ta informacija potrebna je aplikacija za razmještaj studenata.

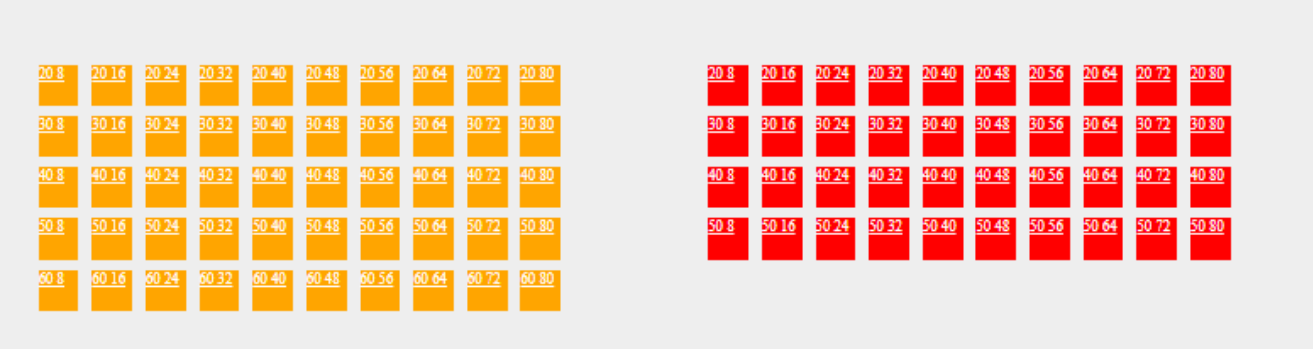


Slika 3.10. Prikazuje informacije iz resursa o odabranom roku pismenog ispita

Ideja završnog rada je algoritam koji raspoređuje studente na sjedeća mjesta. Glavni problem koji je potrebno riješiti je raspored studenata u odnosu na jednu, dvije ili tri grupe ispita. Važno je paziti da studenti budu raspoređeni tako da im je onemogućeno potencijalno prepisivanje. Prije toga potrebno je skicirati ispitne dvorane i odrediti broj sjedećih mjesta pojedine dvorane, zatim napisati algoritam koji generira sjedeća mjesta dvorane te naposljetku razmjestiti studente na sjedeća s obzirom na ispitnu grupu. Algoritam je napisan u PHP skriptnim jezikom koji je predstavljen u prethodnim poglavljima. Sljedeća poglavlja obuhvaćaju pojedine dijelove PHP skripte i slike koje prikazuju kako to izgleda na ekranu. Slika 3.11. prikazuje primjer skripte na stranici Fakulteta Elektrotehnike, računarstva i informacijskih tehnologija.



Slika 3.11. Prikaz prostorija 2 kata na FERIT-u (kamps)



Slika 3.12. Prikaz prostorije K2-11 pomoću php skripte

Slika 3.12 prikazuje tlocrt prostorije K2-11 koja se nalazi na drugom katu, te broj sjedećih mjesta u navedenoj prostoriju. Kako je vidljivo na slici svako sjedeće mjesto ima svoju vrijednost koja se upisuje u bazu podataka. Iz baze podataka gdje se nalaze studenti koji su prijavili ispit preko svog vlastitog id broja se slažu na id broj stolice. Nakon obilaska prostorija i izrade skica rasporeda sjedećih mjesta u pojedinoj prostoriji, potrebno je osmisliti algoritam koji smješta studente na njihova mjesta prema zahtjevima norme ISO 9001:2008.

3.4 Prikaz koda za prostoriju K2-11

```
<?
$dimenzija=8;
//pojašnjenje BOJA => BROJ REDAKA, BROJ STUPACA, ŠIRINA SKUPINE
$skupine=array(
'orange'=>array(5,10,50),
'red'=>array(4,10,50),
);
//-------------------DIV SOBA
echo "<div style='float:left;width:50%;border:2px solid #ccc;height:100%;
background:#eee;'>";


foreach($skupine as $boja=>$podaci){ $odmakY=10;

$x=$podaci[0]; $y=$podaci[1]; $sirina=$podaci[2]; $koeficijent=50/$sirina;

$sirinaStolice=6*$koeficijent; $visinaStolice=$dimenzija;
```

```
//-------------------DIV SKUPINA
echo "<div style='float:left;width:{$sirina}%;height:40%;'>
<div style='position:relative;float:left;width:100%;height:100%;'>";


for($i=1;$i<=$x;$i++){ $odmakY+=$dimenzija+2;

for($j=1;$j<=$y;$j++) { $odmakX+=$dimenzija*$koeficijent;
echo "<div
style='float:left;width:$sirinaStolice%;height:$visinaStolice%;position:absolu
te;
background:$boja;left:{$odmakX}%;top:{$odmakY}%;'>
<a style='color:#fff;font-size:50%;' href='?soba=$_GET[soba]&stolica={$redak}-
{$stolica}'>$odmakY $odmakX</a>
</div>";
                      }
$odmakX=0;
                    }

if($_GET[stolica]=="$redak-$stolica"){echo "<form method='post'>";
echo "<input name='stolica' value='$redak-$stolica'>";
echo "</form>";}

//-------------------DIV SKUPINA KRAJ
echo "</div></div>";


}
//-------------------DIV SOBA KRAJ
echo "</div>";

?>
```

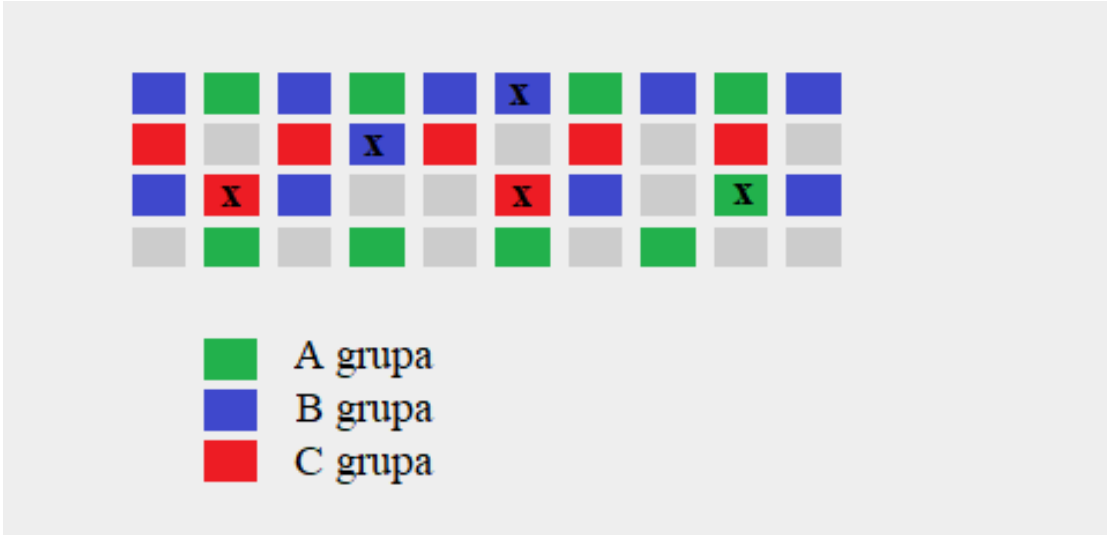
Slika 3.13 Kod za algoritam prostorije K2-11

Narančaste kockice predstavljaju sjedeća mjesta desno od katedre, a crvena lijevo. Mjesta su generirana u skupinama, tako da ih je lakše prilagoditi obliku ispitne dvorane. U array retku „'orange'=>array(5,10,50)“, prvi parametar označava broj redaka u toj skupini, drugi broj određuje broj stupaca, a treći broj je omjer u kojim će podaci biti prikazani na zaslonu. Za prostoriju K2-11 korištene su dvije skupine. Prva skupina označena je narančastom bojom i ima 5 redova i u svakom 10 sjedećih mjesta. Druga skupina označena je crvenom bojom i sastoji se od 4 reda u kojem su također 10 sjedećih mjesta. Kako to izgleda na monitoru prikazano je na slici 3.12.

Takvim prikazom sjedećih mjesta moguće je napisati općeniti algoritam koji će vrijediti za sve ispitne dvorane. U daljnjem pisanju koda definirana je varijabla „$BrojStudenataNaRoku“ pomoću koje se upisuje željeni broj studenata koji pristupaju pismenom ispitu, i varijabla „$BrojGrupa“ gdje odabiremo u koliko ispitnih grupa će biti raspoređeni studenti. Također PHP kodu dodane su dvije for petlje koje će pomoću dva brojača smještati studente na generirana sjedeća mjesta.

```
for($i=1;$i<=$x;$i++){ $odmakY+=$dimenzija+2; $Brojac1++;

for($j=1;$j<=$y;$j++) { $odmakX+=$dimenzija*$koeficijent; $Brojac2++;
```



Slika 3.14 Pogrešno raspoređene ispitne grupe

Na slici 3.14. vidimo pogrešno raspoređene ispitne grupe, tj. potencijalno je omogućeno prepisivanje među studentima A i B grupa (x označava studente koji nisu dovoljno udaljeni od svoje ispitne grupe). Ovaj problem rješavamo algoritmom koji će po određenim standardima rasporediti studente uz što manju mogućnost potencijalnog prepisivanja. Algoritam će također ispisati broj studenata koje nije uspio rasporediti ako je dvorana premala ili broj slobodnih mjesta nakon raspoređivanja tako da je moguće na vrijeme reagirati i prije ispita optimalno rasporediti studente u odgovarajuće dvorane. Na gore prikazanom primjeru vidljivo je puno nepopunjenih mjesta. Možda se ovih 18 studenata moglo smjestiti u manju dvoranu, a trenutnu ostaviti za potencijalno veći broj studenata. U sljedećim poglavljima objašnjen je raspored studenata u dvorani K2-11 s obzirom na jednu, dvije ili tri ispitne grupe.

3.5 Raspored studenata za jednu ispitnu grupu

Za ispit koji je predviđen u ispitnoj dvorani K2-11 na prvom roku prijavljeno je 76 studenata. Ako profesor odredi samo jednu ispitnu grupu, algoritam smješta studente u svaki red s jednim praznim mjestom između studenata te praznim mjestom za red iza studenta. Tako svaki student ima dovoljno mjesta za neometan rad koji je određen prema zahtjevima norme ISO 9001:2008, ali mu je omogućeno potencijalno prepisivanje jer nije dovoljno udaljen od kolege. Ako se u navedenu prostoriji pokuša smjestiti studente na način da im se onemogući potencijalno prepisivanje, od 90 sjedećih mjesta bit će iskorišteno 25 mjesta. Takav algoritam ne stvara optimalnu iskorištenost prostora. Sljedeći algoritam smješta studente tako da sjede dovoljno udaljeno u istom redu, s time da postoji potencijalna opasnost prepisivanja od studenta iz prethodnog ili sljedećeg reda. Logika algoritma je sljedeća: U neparnom redu popuni neparna mjesta a u parnom parna. Dio algoritma koji odrađuje napisanu logiku može se vidjeti ispod:

```
if($BrojGrupa==1){

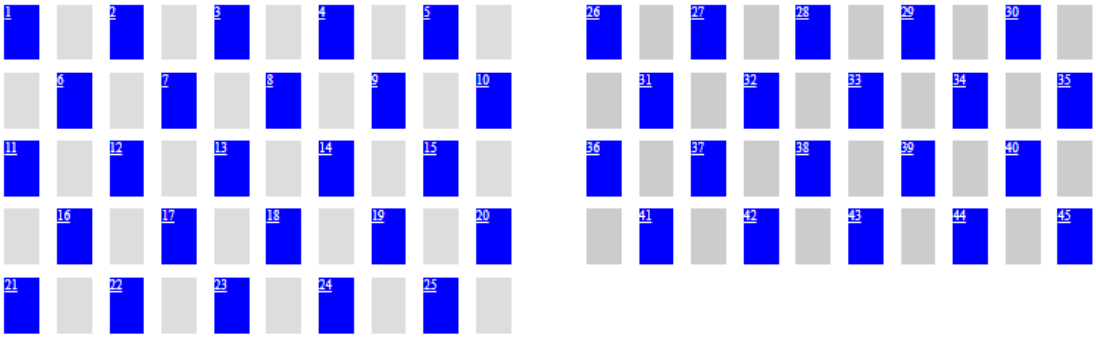
if($neparnired and $neparnosjedalo) {$StudentSjediTu++;}

elseif( $parnired and $parnosjedalo ){$StudentSjediTu++;} }
```

Navedeni dio koda ispisuje studenta kao „broj“ te ga smješta na zadanu poziciju. Kako bi mu ispitna grupa bila dodijeljena napisana je sljedeća petlja:

```
if($BrojGrupa==1){ $bojaGrupe="background:blue;"; }
```

Plavom bojom označena je A grupa. Prolaskom kroz ovu petlju, svi studenti se označavaju jednom grupom kako je prikazano na slici ispod. Jasno se vidi da algoritam ispisuje i broj sjedećih mjesta za odabranu ispitnu grupu i s time olakšava planiranje rasporeda studenata po ispitnim dvoranama.



Slika 3.15 Razmještaj studenata za jednu ispitnu grupu

Kako u ovoj dvorani za jednu ispitnu grupu prema zadanim normama stane samo 45 studenata od željenih 76, potrebno je naći rješenje za ostale studente. Problem je moguće riješiti na nekoliko načina. Prvi način bio bi smještanje traženog broja studenata pomoću dvije ispitne grupe u istu dvoranu. Drugi način je pronalazak veće dvorane gdje je moguće smjestili sve studente u slučaju da je dvorana slobodna. Ako su sve veće dvorane u danom trenutku zauzete, potrebno je pronaći dvoranu za ostatak studenata koji bi u isto vrijeme pisali isti ispit u drugoj ispitnoj dvorani.

3.6 Raspored studenata za dvije ispitne grupe

Isti problem koji je naveden u prethodnom poglavlju rješava se tako da se isti broj studenata raspoređuje u istu dvoranu, ali u dvije ispitne grupe. Algoritam raspoređuje studente u svaki drugi red jedan pokraj drugoga. Tako je potencijalno prepisivanje onemogućeno čak i ako se student okrene (kolega koji piše njegovu grupu ispita udaljen je za jedno sjedeće mjesto). Logika algoritma je sljedeća: U neparnom redu popuni neparna mjesta A grupom a parna B grupom, a parni red ostavi prazan.

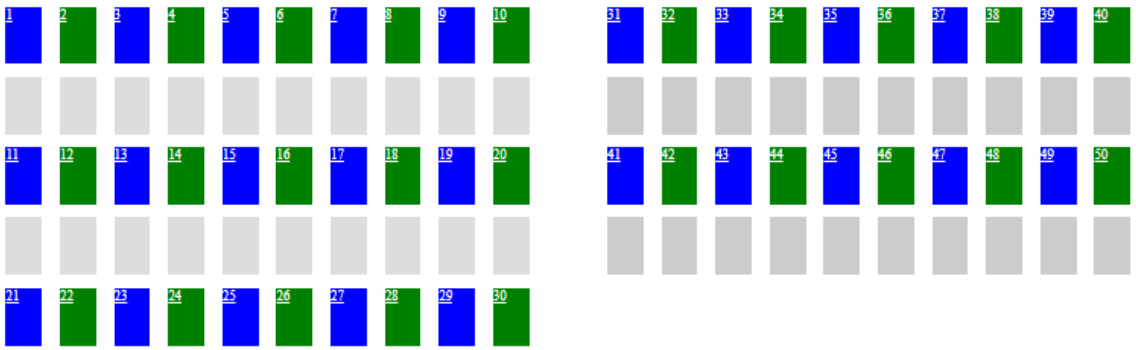
```
if($BrojGrupa==2) {

if($neparnired and $neparnosjedalo ){$StudentSjediTu++;}

if($neparnired and $parnosjedalo) {$StudentSjediTu++;} }

if($BrojGrupa==2){

if ($neparnosjedalo){$bojaGrupe="background:blue;";}else{$bojaGrupe="background:green;
";} }
```



Slika 3.16 Razmještaj studenata za dvije ispitne grupe

Algoritam ispisuje da se 26 studenata nije uspjelo rasporediti. S dvije grupe uspijevamo smjestiti 5 studenata više nego s jednom. U ovom primjeru studenti su kvalitetnije smješteni jer im je smanjena prilika za potencijalno prepisivanje. Problem nije u potpunosti riješen jer je i dalje potrebna još jedna ispitna dvorana za preostalih 50 studenata.

3.7 Raspored studenata za tri ispitne grupe

76 studenata razmješta se u istoj prostoriji sa 90 sjedećih mjesta prema zadanim normama za 3 ispitne grupe. Algoritam smješta C grupu u red koji je bio prazan u prethodnom primjeru. Dakle, popunjava parne retke na parnim mjestima. Studenti iste grupe odvojeni su praznim mjestom ili studentom koji ima suprotnu ispitnu grupu. Kvaliteta sjedenja ostaje ista kao kod prethodnog primjera. Broj studenata koji se ne može rasporediti se smanjio sa 26 na 6, tj. za 20 studenata s obzirom na raspored sjedenja za dvije grupe (ili za 39 studenata s obzirom na raspored sjedenja u istoj prostoriji za jednu grupu). Logika algoritma kaže: U neparnom redu popuni neparna mjesta A grupom, a parna B grupom te u parni red na parno mjesto smjesti C grupu i u zadnji red na parno mjesto smjesti grupu iz prethodnog neparnog retka.

```
if($BrojGrupa==3 ) {

if($neparnired and $neparnosjedalo or $parnosjedalo or $Brojac1==1 or
$Brojac1==$BrojRedovaSuma and $BrojRedovaSuma%2 ){$StudentSjediTu++;}

elseif( $parnired and $parnosjedalo ){$StudentSjediTu++;}

}
```

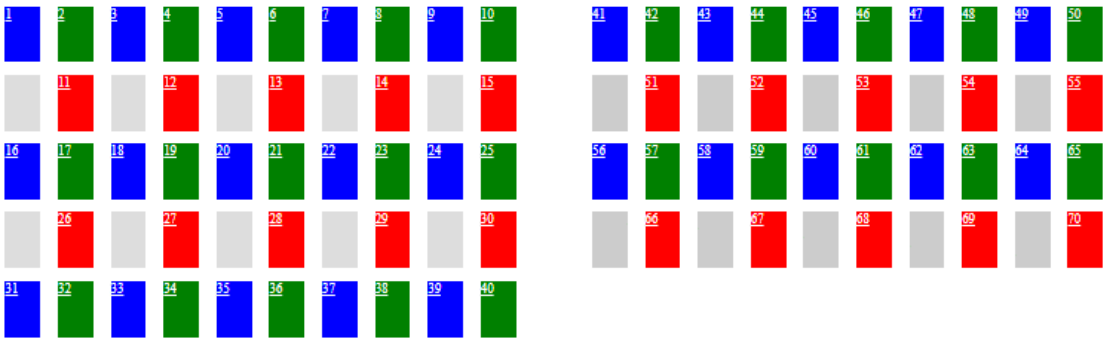
Odabrana boja za C grupu je crvena. Algoritam sadrži uvjete za sve 3 grupe koji, kada se obave, ne rezultiraju najboljim smještajem studenata. Potrebno je u gore prikazanom kodu, u uvjetnoj grani koja ispituje koju ispitnu grupu smo odabrali, staviti još neke parametre kako bi ispitna dvorana bila maksimalno popunjena. U uvjetnoj grani potrebno je ispitati koji je redak paran, a koji neparan da bi se moglo odrediti koju grupu staviti u zadnji red ispitne dvorane te tako smjestili još više studenata, a da se i dalje ne odskače od propisane norme.

if($BrojGrupa==3){

if($parnired==0){$bojaGrupe="background:blue;";} //grupaA

if($parnired==0 and $parnosjedalo) {$bojaGrupe="background:green;";} //grupaB

if($neparnired==0 and $parnosjedalo) {$bojaGrupe="background:red;";} //grupaC }}



Slika 3.17 Razmještaj studenata za tri ispitne grupe

Dakle, kada se zna konačan broj prijavljenih studenata za određeni rok, prije ispita potrebno je odrediti koliko se ispitnih grupa koristiti za određeni kolegij. Tako bi se napokon precizno odabrala ispitna dvorana u kojoj se raspoređuju studenti koji neometano pristupaju i završavaju svoje pismene ispite. Nakon raspoređivanja u tri ispitne grupe, u ovom primjeru ostaje 6 studenata koji nisu smješteni u zadanu ispitnu dvoranu. Time ostaje opcija izabrati veću ispitnu dvoranu ili rasporediti studente u dvije manje ispitne dvorane.

4. ZAKLJUČAK

Cilj završnog rada bilo je automatizirati razmještanje studenata i time olakšati posao profesorima te skratiti vrijeme čekanja ispita studentima. U ovom radu, predložen je način na koji se to može jednostavno napraviti s alatima i jezicima upoznatim na Fakultetu elektrotehnike, računalstva i informacijskih tehnologija. U prethodnim poglavljima, prikazan je primjer razmještaja studenata na jednoj prostoriji. Zbog svoje strukture algoritma, kod se može vrlo jednostavno prilagoditi za bilo koju ispitnu dvoranu te jednako dobro rasporediti studente. Iako je aplikacija jednostavna, ona sa sobom povlači još puno inovativnih projekata, kao na primjer: sve prostorije, tj. stolice na fakultetu, bit će digitalno zabilježene i jednostavno će biti moguće provjeriti koja je najoptimalnija za određeni broj studenata. Profesori unaprijed mogu unijeti svoje ispite u bazu, gdje bi studentima pomoću aplikacije za razmještanje bila dodijeljena određena grupa ispita. S obzirom na grupe, bio bi napravljen razmještaj. Ovaj način razmještanja ne mora biti nužno samo za ovu svrhu. Može, primjerice, služiti u nekom skladištu gdje određeni proizvodi ne smiju biti jedni pokraj drugih i slično. Aplikacija svakako ima puno mjesta za poboljšanje i nadogradnju te ukupni napredak.

LITERATURA

[1.] Kevin Tatroe, Peter MacIntyre i Rasmus Lerdorf: Programiranje PHP, (2006)

[2.] Sustav upravljanja kvalitetom ISO 9001:2008

[3.] Doyle, Matt. Object-Oriented PHP (2011)

[4.] Nikolić, B. HTML Programiranje, (2013)

[5.] PHP priručnik http://php.net/manual/en/tutorial.php (7-9-2017)

[6.] Uvod u PHP http://php.com.hr/77 (8-8-2017)

[7.] PHP tečaj http://www.netakademija.hr/sto-je-php/ (30-7-2017)

[8.] Općenito o MySQL https://www.mysql.com/ (13-8-2017)

SAŽETAK

Naslov: Internet aplikacija za razmještanje studenata na pismenom ispitu

U ovom završnom radu napravljen je algoritam za razmještanje studenata na pismenom ispitu pomoći skriptnog jezika PHP i MySQL baze podataka. Prijavljivanjem na ispit, student automatski dobiva svoje mjesto u predavaonici. Aplikacija se sastoji od baze studenata i baze prostorija u kojoj su naznačena sjedeća mjesta. Algoritam razmješta studente na mjesta u ispitnu dvoranu s obzirom na tri ispitne grupe. Svrha aplikacije je skratiti vrijeme raspoređivanja studenata prije ispita.

Ključne riječi: Internet aplikacija, PHP, MySQL, programiranje.

ABSTRACT

Title: A student-deployment application.

In this final paper, an application for deploying students was written to the script language PHP and MySQL database**.** By signing up for the exam, the student automatically gains his place in the lecture hall. The application consists of the student base and the base where the sitting places are indicated. The application deploys students into places in the lecture hall. The purpose of the application is to shorten the time of scheduling the students before the exam.

Keywords: Internet applications, PHP, MySQL, programming

ŽIVOTOPIS

Josip Golubović rođen je 22.veljače 1995. godine u Vinkovcima. Odrastao i živio u Đurićima s roditeljima, mlađom sestrom i starijim bratom. Školovanje započeo u Osnovnoj Školi „Ivan Filipović“ u Račinovcima. Srednjoškolsko obrazovanje je nastavio u Tehničkoj Školi Nikole Tesle u Vukovaru, gdje je 2013. godine maturirao i stekao zvanje Tehničar za računalstvo. U srednjoj školi polagao je razne tečaje vezane za informatiku. Najznačajniji je HP-ov tečaj koji je položio sa odličnim uspjehom. Nakon završene srednje upisao je Elektrotehnički fakultet u Osijeku, smjer informatika na stručnom studiju.

PRILOG Izvorni kod aplikacije

```
<?


echo "<form method='post' action='2-
31.php'style='float:left;width:50%;height:5%;margin-left:5%;'>
<input type='$BrojStudenataNaRoku'  name='broj1' placeholder='Upisite broj
studenata.'>
<input type='$BrojGrupa'  name='broj2' placeholder='Upisite broj grupa.'>

<input type='submit'>
</form>";
;

$dimenzija=9;
//pojašnjenje BOJA => BROJ REDAKA, BROJ STUPACA, ŠIRINA SKUPINE
$struktura_ucionice=array(
'#ddd'=>array(6,10,50),
'#D3D3D3'=>array(7,10,50),
'#ccc'=>array(7,22,100),
//'green'=>array(5,22,100),
//'grey'=>array(6,10,50),
);



//-------------------DIV SOBA
echo "<div style='float:left;width:90%;border:2px solid #ccc;height:90%;
margin-left:5%;
background:#eee;'>";
$BrojPraznihMjesta=1;

foreach($struktura_ucionice as $boja=>$podaci){ $odmakY=0;  $Brojac1='';



$x=$podaci[0]; $y=$podaci[1]; $sirina=$podaci[2]; $koeficijent=50/$sirina;


$sirinaStolice=6*$koeficijent; $visinaStolice=$dimenzija;


//-------------------DIV SKUPINA
echo "<div style='float:left;width:{$sirina}%;height:40%;margin-
left:0,4%;margin-top:2%;'>
<div style='position:relative;float:left;width:90%;height:90%;'>";
```

```
for($i=1;$i<=$x;$i++){ $odmakY+=$dimenzija+2; $Brojac1++;

for($j=1;$j<=$y;$j++) { $odmakX+=$dimenzija*$koeficijent; $Brojac2++;

$neparnosjedalo=($Brojac2)%2;
$parnosjedalo=($Brojac2+1)%2;
$parnired=($Brojac1+1)%2;
$neparnired=($Brojac1)%2;

$BrojStudenataNaRoku = $_POST['broj1'];
$BrojGrupa = $_POST['broj2'];


//raspored stolica
if ($BrojStudenataNaRoku==''){$StudentSjediTu++;}

//raspored sjedenja za jednu grupe
if($BrojGrupa==1){
if($neparnired and $neparnosjedalo) {$StudentSjediTu++;}
elseif( $parnired and $parnosjedalo ){$StudentSjediTu++;}


}

//raspored sjedenja za dvije grupe
if($BrojGrupa==2) {
if($neparnired and $neparnosjedalo ){$StudentSjediTu++;}
if($neparnired and $parnosjedalo) {$StudentSjediTu++;} }

//raspored sjedenja tri grupe
if($BrojGrupa==3 ) {
if($neparnired and $neparnosjedalo or $parnosjedalo or $Brojac1==1 or
$Brojac1==$BrojRedovaSuma and $BrojRedovaSuma%2 ){$StudentSjediTu++;}
elseif( $parnired and $parnosjedalo ){$StudentSjediTu++;}
}


$SumaStudenata+=$StudentSjediTu;


//boja grupa
if($StudentSjediTu!=''){
if($BrojGrupa==1){ $bojaGrupe="background:blue;"; }
if($SumaStudenata>$BrojStudenataNaRoku){$bojaGrupe="background:gray";}
```

```
if($BrojGrupa==2){if
($neparnosjedalo){$bojaGrupe="background:blue;";}else{$bojaGrupe="background:g
reen; ";}
if($SumaStudenata>$BrojStudenataNaRoku){$bojaGrupe="background:gray";}
}

if($BrojGrupa==3){
if($parnired==0){$bojaGrupe="background:blue;";} //grupaA
if($parnired==0 and $parnosjedalo) {$bojaGrupe="background:green;";} //grupaB
if($neparnired==0 and $parnosjedalo) {$bojaGrupe="background:red;";} //grupaC
if($SumaStudenata>$BrojStudenataNaRoku){$bojaGrupe="background:gray";}

}
}

echo "<div
style='float:left;width:$sirinaStolice%;height:$visinaStolice%;position:absolu
te;
background:$boja;left:{$odmakX}%;top:{$odmakY}%;$bojaGrupe'>
<a style='color:#fff;font-size:50%;' href='?soba=$_GET[soba]&stolica={$redak}-
{$stolica}'>";
if($StudentSjediTu){
if($SumaStudenata<=$BrojStudenataNaRoku){ echo "$SumaStudenata";}
elseif($SumaStudenata>$BrojStudenataNaRoku){ echo
"$BrojPraznihMjesta";$BrojPraznihMjesta++;}
}
echo "</a></div>";




$StudentSjediTu=$bojaGrupe='';
                        }


$odmakX=0;
                    }


if($_GET[stolica]=="$redak-$stolica"){echo "<form method='post'>";
echo "<input name='stolica' value='$redak-$stolica'>";
echo "</form>";}

//-------------------DIV SKUPINA KRAJ
echo "</div></div>";

}
$NemaMjesta=$BrojStudenataNaRoku-$SumaStudenata;
```

```php
if ($BrojStudenataNaRoku<$SumaStudenata)
{
   echo "Broj slobodnih mjesta: ";
   echo abs($NemaMjesta) ;}
else {
echo "$NemaMjesta studenata ne moze se rasporediti.<br/>";}


echo "<br/>Broj sjedecih mjesta: $SumaStudenata.<br/>";

//-------------------DIV SOBA KRAJ
echo "</div>";

?>
```

Ispitni termini SQL

```sql
CREATE TABLE IF NOT EXISTS `ispiti` (
`id` int(11) NOT NULL,
  `predmet` varchar(50) NOT NULL,
  `rok` varchar(50) NOT NULL,
  `prostorija` varchar(5) NOT NULL,
  `kolokvij` varchar(20) NOT NULL,
  `grupa` varchar(10) NOT NULL,
  `datum` varchar(20) NOT NULL,
  `opis` blob NOT NULL,
  `tekst` longblob NOT NULL,
  `autor` varchar(100) NOT NULL,
  `brojstranica` varchar(10) NOT NULL
);
```

Tabica za prostorije SQL (FERIT)

```sql
CREATE TABLE IF NOT EXISTS `prostor` (
  `id` int(11) NOT NULL,
  `lokacija` varchar(20) NOT NULL,
  `kat` varchar(3) NOT NULL,
  `naziv` varchar(30) NOT NULL,
  `opis` varchar(500) NOT NULL,
  `broj_mjesta` varchar(5) NOT NULL,
  `koordinate` varchar(100) NOT NULL,
  `zavod` varchar(10) NOT NULL,
  `zauzece` varchar(5) NOT NULL
);
```

Primjer grupe studenta iz resursa

```
<?xml version="1.0" encoding="Windows-1250"?>
<grupaStudenata idgrupe="3678">
<brojGrupe>1</brojGrupe>
<ime>DEB</ime>
<opis></opis>
<predmet isvuid="149737" semestar="1">Energetska učinkovitost</predmet>
<studenti brojStudenata="34">
<student>
<jmbag>0165043258</jmbag>
<ime>Ivan</ime>
<prezime>Arambašić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054302</jmbag>
<ime>Matej</ime>
<prezime>Arić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0246046127</jmbag>
<ime>Ivan</ime>
<prezime>Bančić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054440</jmbag>
<ime>Tomislav</ime>
<prezime>Čabrajac</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165051251</jmbag>
<ime>Ivan</ime>
<prezime>Čurčić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054461</jmbag>
<ime>Martina</ime>
<prezime>Ćurić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054498</jmbag>
```

```
<ime>Stjepan</ime>
<prezime>Duspara</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054568</jmbag>
<ime>Kristijan</ime>
<prezime>Hanžek</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054589</jmbag>
<ime>Petar</ime>
<prezime>Huis</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165051379</jmbag>
<ime>Dario</ime>
<prezime>Ivančević</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054594</jmbag>
<ime>Filip</ime>
<prezime>Jakupec</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054601</jmbag>
<ime>Darko</ime>
<prezime>Jakus</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165051428</jmbag>
<ime>Ivana</ime>
<prezime>Josipović</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165052446</jmbag>
<ime>Mihael</ime>
<prezime>Kaučić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054690</jmbag>
<ime>Josip</ime>
```

```
<prezime>Klinc</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054729</jmbag>
<ime>Marija</ime>
<prezime>Kovačević</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054760</jmbag>
<ime>Damir</ime>
<prezime>Krancpiler</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054776</jmbag>
<ime>Mario</ime>
<prezime>Krešo</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0314002596</jmbag>
<ime>Marcel</ime>
<prezime>Leder</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054888</jmbag>
<ime>Josip</ime>
<prezime>Maskaljević</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055016</jmbag>
<ime>Ivan</ime>
<prezime>Nikolić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055037</jmbag>
<ime>Nikolina</ime>
<prezime>Orlović</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055058</jmbag>
<ime>Hrvoje</ime>
<prezime>Palčić</prezime>
```

```
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055149</jmbag>
<ime>Tomislav</ime>
<prezime>Pirić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055203</jmbag>
<ime>Nedjeljko</ime>
<prezime>Rado</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0036468464</jmbag>
<ime>Filip</ime>
<prezime>Relić</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165051678</jmbag>
<ime>Danijel</ime>
<prezime>Šalković</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165055411</jmbag>
<ime>Antonela</ime>
<prezime>Vidović</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165054045</jmbag>
<ime>Nikolina</ime>
<prezime>Zovko</prezime>
<redovni>1</redovni>
</student>
<student>
<jmbag>0165022894</jmbag>
<ime>Ivica</ime>
<prezime>Čabraja</prezime>
<redovni>3</redovni>
</student>
<student>
<jmbag>0149011968</jmbag>
<ime>Damir</ime>
<prezime>Hanzer</prezime>
<redovni>3</redovni>
```

```
</student>
<student>
<jmbag>0165020538</jmbag>
<ime>Ivan</ime>
<prezime>Ličanin</prezime>
<redovni>3</redovni>
</student>
<student>
<jmbag>0165020522</jmbag>
<ime>Ivan</ime>
<prezime>Pućo</prezime>
<redovni>3</redovni>
</student>
<student>
<jmbag>0165042133</jmbag>
<ime>Robert</ime>
<prezime>Štrangar</prezime>
<redovni>3</redovni>
</student>
</studenti>
</grupaStudenata>
```