

## WEB PROGRAMIRANJE – PHP ZADACI

**Zadatak 1** Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o osiguranicima Fonda za zdravstveno osiguranje. Koraci za implementaciju su sledeći:

- a. Kreirati bazu podataka **zadatak1** i u njoj tabelu **osiguranik** izvršavanjem navedenih upita. Tabela **osiguranik** sadrži kolone **ime**, **prezime** i **lbo** (jedinstveni lični broj osiguranika). Kreirati fajl *lib.php* i u njemu sledeće funkcije za rad sa bazom podataka:
  - Funkciju **vratiSveOsiguranike()** koja vraća sve osiguranike iz baze. Podaci se vraćaju u obliku asocijativnog niza.
  - Funkciju **obrisiOsiguranika(\$lbo)** koja briše osiguranika čiji je LBO prosleđen.

```
CREATE TABLE IF NOT EXISTS osiguranik (  
  lbo varchar(15) NOT NULL,  
  ime varchar(20) NOT NULL,  
  prezime varchar(20) NOT NULL  
);
```

```
INSERT INTO osiguranik (lbo, ime, prezime) VALUES  
( '12345', 'Marko', 'Markovic'),  
( '12346', 'Milan', 'Milanovic'),  
( '12347', 'Petar', 'Petrovic'),  
( '12349', 'Janko', 'Jankovic'),  
( '12348', 'Jovan', 'Jovanovic');
```

- b. Na osnovu navedenih funkcija implementirati stranu *index.php* koja prikazuje sve osiguranike iz baze (slika 1). Pored svakog osiguranika postoji *checkbox* kontrola za izbor jednog ili više osiguranika za brisanje. Kada se strana prvi put otvori nijedna *checkbox* kontrola nije selektovana.
- c. Kada se klikne na dugme *Dalje* otvara se strana *pregled.php* kao na slici 2 koja prikazuje samo osiguranike izabrane za brisanje. Na ovoj strani nije moguće menjati podatke. Klik na dugme *Brisanje* briše osiguranika iz baze podataka i ispisuje korisniku poruku o tome da li je brisanje uspelo.
- d. Klik na link *Nazad* vraća korisnika na stranicu *index.php* bez snimanja i pruža mu mogućnost da promeni prethodno izabrane osiguranike. Za osiguranike koje su prethodno bili izabrani *checkbox* kontrole su selektovane.

Ime	Prezime	LBO	Za brisanje
Marko	Markovic	12345	<input type="checkbox"/>
Milan	Milanovic	12346	<input checked="" type="checkbox"/>
Petar	Petrovic	12347	<input type="checkbox"/>
Janko	Jankovic	12349	<input checked="" type="checkbox"/>
Jovan	Jovanovic	12348	<input type="checkbox"/>
Dalje			

Slika 1

Ime	Prezime	LBO
Milan	Milanovic	12346
Janko	Jankovic	12349
Nazad		Brisanje

Slika 2

#### Napomene:

- Za rešavanje zadatka koristiti **smarty** mehanizam šablona. Ukoliko se zadatak radi bez primene smarty-ja gubi se 20 posto poena zadatka.
- Za prenos parametara **koristiti promenljive sesije, a ne hidden polja**.

**Zadatak 2** Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o pacijentima doma zdravlja. Koraci za implementaciju su sledeći:

- Kreirati bazu podataka **zadatak2** i u njoj tabelu **pacijent** izvršavanjem navedenog upita. Tabela **pacijent** sadrži kolone **lbo** (jedinostveni lični broj osiguranika), **ime**, **prezime**, **posao** (naziv radne organizacije, za pacijente koji imaju 19 i više godina ovo polje je prazno) i **skola** (naziv skole koju pacijent pohađa, za pacijente koji imaju manje od 19 godina ovo polje je prazno). Kreirati fajl *lib.php* i u njemu sledeće funkcije za rad sa bazom podataka:

- Funkciju **vratiSvePacijente()** koja vraća sve pacijente iz baze. Podaci se vraćaju u obliku asocijativnog niza.

- Funkciju **dodajPacijenta(\$lbo, \$ime, \$prezime, \$posao, \$skola)** koja dodaje novog pacijenta u bazu.

```
CREATE TABLE IF NOT EXISTS pacijent (
  lbo varchar(15) NOT NULL,
  ime varchar(20) NOT NULL,
  prezime varchar(20) NOT NULL,
  posao varchar(30) NOT NULL,
  skola varchar(30) NOT NULL
);
```

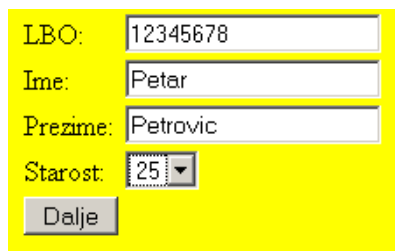
- Implementirati stranu *index.php* koja omogućava unos LBO, imena, prezimena i starosti osiguranika (slika 1). Starost se bira korišćenjem *select* kontrole koja treba da ima ponuđene

opcije od 0 do 100 godina. Preporučeni način za popunjavanje ove kontrole je kroz petlju u tpl fajlu za ovu stranicu.

c. Kada se klikne na dugme *Dalje* otvara se strana *next.php* kao na slici 2 koja može da ima 2 različita izgleda. Ako je prethodno izabrano da pacijent ima 19 i više godina na ovoj strani se zadaje njegova radna organizacija (slika 2 gore). Ako je prethodno izabrano da pacijent ima manje od 19 godina na ovoj strani se upisuju podaci o njegovoj školi (slika 2 dole).

d. Klik na dugme *Snimi* dodaje novog pacijenta u bazu podataka i ispisuje korisniku poruku o tome da li je upis uspeo.

e. Klik na link *Nazad* vraća korisnika na stranicu *index.php* bez snimanja i prikazuje prethodno zadate podatke koje je sada moguće promeniti.



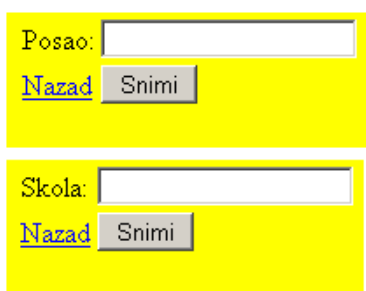
LBO:

Ime:

Prezime:

Starost:

Slika 1



Posao:

[Nazad](#)

Skola:

[Nazad](#)

Slika 2

#### Napomene:

- Za rešavanje zadatka koristiti **smarty** mehanizam šablona. Ukoliko se zadatak radi bez primene smarty-ja gubi se 20 posto poena zadatka.
- Za prenos parametara **koristiti hidden polja, a ne promenljive sesije**.

**Zadatak 3** Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o komponentama u prodavnici računarske opreme. Koraci za implementaciju su sledeći:

- a. Kreirati bazu podataka **zadatak3** i u njoj tabelu **komponenta** izvršavanjem navedenih upita. Tabela **komponenta** sadrži kolone **sifra** (jedinствена šifra proizvoda), **naziv**, i **status**. Status može uzeti jednu od sledeće 3 vrednosti: **dostupno** – proizvod postoji u prodavnici, **nedostupno** – proizvod ne postoji u prodavnici i **naruceno** – proizvod naručen od dobavljača. Kreirati fajl *lib.php* i u njemu sledeće funkcije za rad sa bazom podataka:

- Funkciju ***vraTiSveKomponente()*** koja vraća sve komponente iz baze. Podaci se vraćaju u obliku asocijativnog niza.

- Funkciju ***promeniStatus(\$sifra, \$status)*** koja menja kolonu status u vrsti sa zadatom šifrom.

```
CREATE TABLE IF NOT EXISTS komponenta (  
  sifra varchar(20) NOT NULL,  
  naziv varchar(50) NOT NULL,  
  status varchar(11) NOT NULL  
);
```

```
INSERT INTO komponenta (sifra, naziv, status)  
VALUES  
( '1234567', 'Gigabyte GA-M68MT-S2', 'dostupno'),  
( '1234568', 'MSI 760GM-P23', 'nedostupno'),  
( '1234569', 'Intel Core i5-2300', 'naruceno'),  
( '1234570', 'AMD Athlon II X4 Quad-Core', 'dostupno'),  
( '1234571', 'GeForce GT520 Asus', 'nedostupno'),  
( '1234572', 'AMD Radeon 6670 Sapphire', 'dostupno');
```

b. Na osnovu kreiranih funkcija implementirati stranu *index.php* koja prikazuje sve komponente iz baze (slika 1). Za prikaz i promenu statusa koristi se *radio* kontrola.

c. Kada se klikne na dugme *Dalje* otvara se strana *pregled.php* kao na slici 2 koja prikazuje sve podatke kao i *index.php*. Podaci se za sada ne snimaju u bazi, već se čuvaju korišćenjem promenljive sesije. Na ovoj strani nije moguće menjati podatke.

d. Klik na dugme *Prosledi* izvršava *update* svih slogova u bazi i ispisuje korisniku poruku o tome da li je operacija uspeła. Klik na link *Nazad* vraća korisnika na stranicu *index.php* bez snimanja, prikazuje ranije izabrane statuse (ne statuse iz baze) i pruža mogućnost za promenu prethodno izabranih statusa.

Sifra	Naziv	Status
1234567	Gigabyte GA-M68MT-S2	<input checked="" type="radio"/> Dostupno <input type="radio"/> Nedostupno <input type="radio"/> Naruceno
1234568	MSI 760GM-P23	<input type="radio"/> Dostupno <input checked="" type="radio"/> Nedostupno <input type="radio"/> Naruceno
1234569	Intel Core i5-2300	<input type="radio"/> Dostupno <input type="radio"/> Nedostupno <input checked="" type="radio"/> Naruceno
1234570	AMD Athlon II X4 Quad-Core	<input checked="" type="radio"/> Dostupno <input type="radio"/> Nedostupno <input type="radio"/> Naruceno
1234571	GeForce GT520 Asus	<input type="radio"/> Dostupno <input checked="" type="radio"/> Nedostupno <input type="radio"/> Naruceno
1234572	AMD Radeon 6670 Sapphire	<input type="radio"/> Dostupno <input type="radio"/> Nedostupno <input checked="" type="radio"/> Naruceno
<input type="button" value="Dalje"/>		

Slika 1

Sifra	Naziv	Status
1234567	Gigabyte GA-M68MT-S2	dostupno
1234568	MSI 760GM-P23	nedostupno
1234569	Intel Core i5-2300	naruceno
1234570	AMD Athlon II X4 Quad-Core	dostupno
1234571	GeForce GT520 Asus	nedostupno
1234572	AMD Radeon 6670 Sapphire	naruceno
<a href="#">Nazad</a>		<a href="#">Prosledi</a>

Slika 2

#### Napomene:

- Za rešavanje zadatka koristiti **smarty** mehanizam šablona. Ukoliko se zadatak radi bez primene smarty-ja gubi se 20 posto poena zadatka.
- Za prenos parametara **koristiti promenljive sesije, a ne hidden polja**.

**Zadatak 4** Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o radnicima. Koraci za implementaciju su sledeći:

- Kreirati bazu podataka **zadatak4** i u njoj tabelu **radnik** izvršavanjem navedenih upita. Tabela **radnik** sadrži kolone **ime**, **prezime** i **jmbg** (jedinstveni matični broj građana). JMBG tretirati kao ključ u tabeli koji nije moguće menjati. Kreirati fajl **lib.php** i u njemu sledeće funkcije za rad sa bazom podataka:

- Funkciju **vratiSveRadnike()** koja vraća sve radnike iz baze. Podaci se vraćaju u obliku asocijativnog niza.

- Funkciju **izmeniRadnika(\$lbo, \$ime, \$prezime)** koja radniku čiji je JMBG prosleđen menja ime i prezime.

```
CREATE TABLE IF NOT EXISTS radnik (
  jmbg varchar(13) NOT NULL,
  ime varchar(20) NOT NULL,
  prezime varchar(20) NOT NULL
);
```

```
INSERT INTO radnik (jmbg, ime, prezime) VALUES
('1234567890123', 'Marko', 'Markovic'),
('2234567890123', 'Milan', 'Milanovic'),
('3234567890123', 'Petar', 'Petrovic'),
('4234567890123', 'Janko', 'Jankovic'),
('5234567890123', 'Jovan', 'Jovanovic');
```

- Na osnovu navedenih funkcija implementirati stranu **index.php** koja prikazuje sve radnike iz baze (slika 1). Pored svakog radnika postoji **radio** kontrola za izbor tačno jednog radnika za izmenu. Kada se strana prvi put otvori nijedna **radio** kontrola nije selektovana.

c. Kada se klikne na dugme *Izmeni* otvara se strana *izmena.php* kao na slici 2 koja prikazuje samo radnika izabranog za izmenu. Na ovoj strani je moguće menjati ime i prezime. Strana sadrži i 2 *submit* dugmeta *Odustani* i *Prosledi* u okviru iste forme.

d. Klik na dugme *Prosledi* otvara stranicu *index.php* gde se vrši snimanje izmena, a zatim prikazuje lista radnika kao na slici 1.

e. Klik na dugme *Odustani* otvara stranicu *index.php* bez snimanja i prikazuje prikazuje listu radnika kao na slici 1 **gde je selektovana radio kontrola** pored radnika koji je prethodno bio označen za menjanje.

Ime	Prezime	JMBG	Za izmenu
Marko	Markovic	1234567890123	<input type="radio"/>
Milan	Milanovic	2234567890123	<input type="radio"/>
Petar	Petrovic	3234567890123	<input checked="" type="radio"/>
Janko	Jankovic	4234567890123	<input type="radio"/>
Jovan	Jovanovic	5234567890123	<input type="radio"/>
<input type="button" value="Izmeni"/>			

Slika 1

Ime:	<input type="text" value="Petar"/>
Prezime:	<input type="text" value="Petrovic"/>
<input type="button" value="Odustani"/>	<input type="button" value="Prosledi"/>

Slika 2

#### Napomene:

- Za rešavanje zadatka koristiti **smarty** mehanizam šablona. Ukoliko se zadatak radi bez primene smarty-ja gubi se 20 posto poena zadatka.
- Za prenos parametara **koristiti hidden polja, a ne promenljive sesije**.

#### Zadatak 5

##### Važne napomene:

- Za rešavanje zadatka koristiti **smarty** mehanizam šablona. Ukoliko se zadatak radi bez primene smarty-ja gubi se 20% poena zadatka.
- Za prenos parametara koristiti promenljive sesije, a ne hidden polja ili JavaScript metode.

Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o studentima. Koraci za implementaciju su sledeći:

- a. Kreirati bazu podataka **zadatak5** i u njoj tabelu **student** izvršavanjem navedenih upita. Tabela **student** sadrži kolone **indeks**, **ime**, **prezime**, **godina** i **status** (finansiranje iz budžeta – FIB ili samofinansiranje – SAF). Indeks tretirati kao ključ u tabeli koji nije moguće menjati. Kreirati fajl *lib.php* i u njemu sledeće funkcije za rad sa bazom podataka:

- Funkciju ***vрати\_sve\_studente()*** koja vraća sve studente iz baze. Podaci se vraćaju u obliku asocijativnog niza.

- Funkciju ***izmeni\_studenta(\$indeks, \$ime, \$prezime, \$godina, \$status)*** koja studentu čiji je indeks prosleđen menja ime, prezime, godinu i status.

```
CREATE TABLE IF NOT EXISTS student (  
  indeks varchar(15) NOT NULL,  
  ime varchar(20) NOT NULL,  
  prezime varchar(20) NOT NULL,  
  godina varchar(1) NOT NULL,  
  status varchar(3) NOT NULL  
);
```

```
INSERT INTO student (indeks, ime, prezime, godina,  
status) VALUES  
( '12345', 'Marko', 'Markovic', '1', 'FIB'),  
( '12346', 'Milan', 'Milanovic', '2', 'FIB'),  
( '12347', 'Petar', 'Petrovic', '2', 'SAF'),  
( '12349', 'Janko', 'Jankovic', '3', 'SAF'),  
( '12348', 'Jovan', 'Jovanovic', '4', 'FIB');
```

- b. Na osnovu navedenih funkcija implementirati stranu *index.php* koja prikazuje sve studente iz baze (slika 1). Pored svakog studenta postoji *checkbox* kontrola za izbor jednog ili više studenata za izmenu. Kada se strana prvi put otvori nijedna *checkbox* kontrola nije selektovana.

c. Kada se klikne na dugme *Izmeni* otvara se strana *izmena.php* kao na slici 2 koja prikazuje samo studente izabrane za izmenu. Na ovoj strani je moguće menjati ime, prezime, godinu i status. **Ime** i **prezime** se zadaju *input* kontrolom tipa *text*. **Godina** se zadaju pomoću *select* kontrole. **Status** se zadaje *input* kontrolom tipa *radio*. Strana sadrži i 2 *submit* dugmeta „Odustani“ i „Prosledi“ u okviru iste forme.

d. Klik na dugme „Prosledi“ otvara stranicu *index.php* gde se vrši snimanje izmena, a zatim prikazuje lista studenata kao na slici 1.

e. Klik na dugme „Odustani“ otvara stranicu *index.php* bez snimanja i prikazuje listu studenata kao na slici 1 **gde su selektovane checkbox kontrole** pored studenata koji su prethodno bili označeni za izmenu.

Indeks	Ime	Prezime	Godina	Status	Za izmenu
12345	Marko	Markovic	1	FIB	<input checked="" type="checkbox"/>
12346	Milan	Milanovic	2	FIB	<input type="checkbox"/>
12347	Petar	Petrovic	2	SAF	<input checked="" type="checkbox"/>
12349	Janko	Jankovic	3	SAF	<input type="checkbox"/>
12348	Jovan	Jovanovic	4	FIB	<input type="checkbox"/>

Slika 1

Indeks:	12345
Ime:	<input type="text" value="Marko"/>
Prezime:	<input type="text" value="Markovic"/>
Godina:	<input type="text" value="1"/>
Status:	FIB <input checked="" type="radio"/> SAF <input type="radio"/>
Indeks:	12347
Ime:	<input type="text" value="Petar"/>
Prezime:	<input type="text" value="Petrovic"/>
Godina:	<input type="text" value="2"/>
Status:	FIB <input type="radio"/> SAF <input checked="" type="radio"/>
<input type="button" value="Odustani"/> <input type="button" value="Prosledi"/>	

Slika 2

## Zadatak 6

Važne napomene:

- Za rešavanje zadatka koristiti *smarty* mehanizam šablona. Ukoliko se zadatak radi bez primene *smarty*-ja gubi se 20% poena zadatka.
- Za prenos parametara koristiti hidden polja, a ne promenljive sesije ili JavaScript metode.

Korišćenjem PHP jezika napraviti aplikaciju za rad sa podacima o osiguranicima fonda za zdravstveno osiguranje. Koraci za implementaciju su sledeći:

a. Kreirati bazu podataka **zadatak6** i u njoj tabelu **osiguranik** izvršavanjem navedenih upita. Tabela **osiguranik** sadrži kolone **lbo** (lični broj osiguranika), **ime**, **prezime**, **pol** (skraćena oznaka M ili Z) i **datum\_rodjenja** (u formatu dd.mm.yyyy.). Kreirati fajl *lib.php* i u njemu sledeće funkcije za rad sa bazom podataka:

- Funkciju ***vрати\_sve\_osiguranike()*** koja vraća sve osiguranike iz baze podataka. Podaci se vraćaju u obliku asocijativnog niza.

- Funkciju ***dodaj\_osiguranika(\$lbo, \$ime, \$prezime, \$pol, \$datum\_rodjenja)*** koja dodaje novog osiguranika.

```
CREATE TABLE IF NOT EXISTS osiguranik (
  lbo varchar(15) NOT NULL,
  ime varchar(20) NOT NULL,
  prezime varchar(20) NOT NULL,
  pol varchar(1) NOT NULL,
  datum_rodjenja varchar(11) NOT NULL
);
```

```
INSERT INTO osiguranik (lbo, ime, prezime, pol,
datum_rodjenja) VALUES
('123456', 'Marko', 'Markovic', 'M', '10.10.1961.'),
('123457', 'Milan', 'Milanovic', 'M', '15.11.1971.'),
('123458', 'Petra', 'Petrovic', 'Z', '20.12.1981.'),
('123459', 'Janko', 'Jankovic', 'M', '25.10.1991.'),
('123450', 'Jovana', 'Jovanovic', 'Z', '30.11.2001.');
```



- b. Na osnovu kreiranih funkcija implementirati stranu *index.php* koja prikazuje sve osiguranike iz baze (slika 1). Ispod liste svih osiguranika nalaze se kontrole za unos podataka o novom osiguraniku. **LBO**, **ime** i **prezime** se zadaju *input* kontrolom tipa *text*. **Pol** se zadaje *input* kontrolom tipa *radio*. **Datum\_rodjenja** se zadaju pomoću 3 *select* kontrole. *Select* kontrole omogućavaju da se unesu vrednosti od 1 do 12 za dane, od 1 do 31 za mesece i od 1900 do 2100 za godinu. Te vrednosti izgenerisati petljom u šablonu. Dugme „Dodaj“ je tipa *submit*.
- c. Kada se klikne na dugme „Dodaj“ na strani *index.php* otvara se strana *pregled.php* kao na slici 2 koja prikazuje podatke o novom osiguraniku. Na ovoj strani nije moguće menjati podatke.
- d. Klik na dugme „Prosledi“ na strani *prosledi.php* vraća korisnika na stranu *index.php*, dodaje novog osiguranika u bazu podataka i prikazuje ponovo sadržaj kao na slici 1. Ukoliko dodavanje nije uspelo korisniku se prikazuje poruka o tome.
- e. Klik na dugme „Nazad“ vraća korisnika na stranicu *index.php* bez snimanja, a **polja za unos sadrže sve prethodno zadate podatke**.

LBO	Ime	Prezime	Pol	Datum rodjenja
123456	Marko	Markovic	M	10.10.1961.
123457	Milan	Milanovic	M	15.11.1971.
123458	Petra	Petrovic	Z	20.12.1981.
123459	Janko	Jankovic	M	25.10.1991.

---

LBO:   
 Ime:   
 Prezime:   
 Pol: M ☐ Z ☒  
 Dan rodjenja:   
 Mesec rodjenja:   
 Godina rodjenja:

Slika 1

LBO: 123450  
 Ime: Jovana  
 Prezime: Jovanovic  
 Pol: Z  
 Datum rodjenja: 30.11.2001

Slika 2