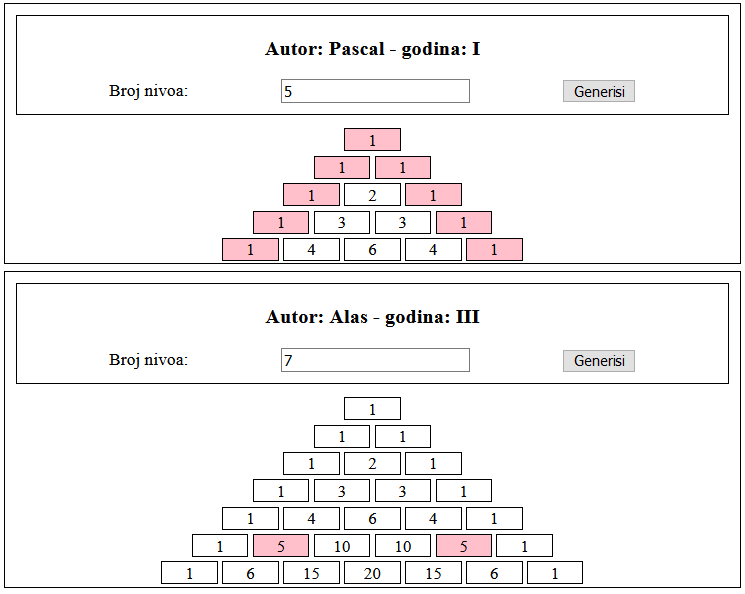
Elektronski fakultet u Nišu - Katedra za računarstvo

# Web programiranje

20.06.2019.

**Zadatak 1.** U JavaScript-u napraviti aplikaciju za generisanje Paskalovog trougla. Svaki Paskalov trougao se sastoji od polja, gde je vrednost svakog polja jednaka levom i desnom prethodniku trenutnog polja. Prikaz Paskalovog trougla organizovati kao na slici. Omogućiti korisniku zadavanje nivoa i generisanje prikaza Paskalovog trougla pomoću forme koja se nalazi u gornjem delu i fiksne je visine od 100 px. Formu i prikaz trougla kreirati koristeći flex dizajn.

Klikom na dugme “Generiši“ neophodno je rekreirati indeksiranu strukturu korišćenu za čuvanje polja i osvežiti prikaz trougla tako da odgovara novoj vrednosti nivoa. Inicijalno prikazati samo formu za unos nivoa, **bez** ikakvog **prikaza** trougla ili polja. Smatrati da će korisnik za nivo uvek uneti vrednost veću od 3.

Nakon što je generisan i prikazan trougao sa svim svojim poljima, klik na bilo koje polje trougla treba da oboji sva polja sa tom vrednošću crvenom bojom.

Za implementaciju zadatka kreirati klase Trougao i Polje sa odgovarajućim atributima i metodama. Prilikom kreiranja objekata ovih klasa proveriti validnost prosleđenih parametara i postaviti podrazumevane vrednosti ukoliko su nevalidne. U glavnom programu kreirati dve instance trougla sa proizvoljnim autorom i godinom studija.

**Napomena:** Nije dozvoljeno dinamičko dodeljivanje i korišćenje ID-jeva u JavaScript kodu. Crtanje svih elemenata uraditi isključivo kroz JavaScript. JavaScript, CSS i HTML odvojiti u zasebne fajlove.

**Napomena**: Dupliranje i „hard-coded“ kod donosi negativne poene.

**Zadatak 2.** Koristeći PHP, AJAX, Fetch i JSON napraviti aplikaciju za rad sa bazom podataka o korisnicima jednog mejl servisa. Koraci za implementaciju su sledeći:

1. Pretpostaviti da postoji baza podataka **jun**,i u njoj tabele **servis** i **korisnik** kreirane izvršenjem navedenih upita.
2. Kreirati klase Servis i Korisnik koje odgovaraju tabelama servis i korisnik i za **svaku** kolonu imaju odgovarajući atribut. Od metoda obe klase imaju konstruktor koji postavlja sve atribute. Klasa Servis ima listu Korisnika kao i metodu za dodavanje Korisnika u listu.
3. Kreirati fajl *lib.php* i u njemu **KLASU** sa metodama odgovornim za rad sa bazom podataka:

- Metodu ***vratiServis()*** koja vraća servis sa svim trenutnim korisnicima iz baze podataka. Podaci se vraćaju preko objekta klase Servis. Možete pretpostaviti da u bazi postoji samo jedan servis.

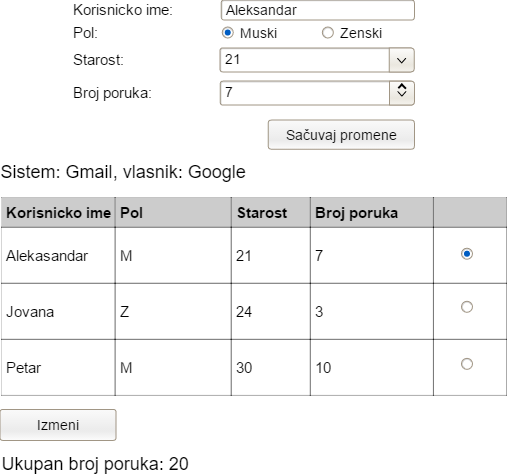
- Metodu ***vratiKorisnika($idKorisnika)*** koja vraća podatke o korisniku čiji je id prosleđen.

- Metodu ***azurirajKorisnika(Korisnik $Korisnik)*** koja ažurira podatke o korisniku u bazi podataka.

1. Kreirati HTML stranu *index.html* koja inicijalno prikazuje podatke o sistemu i sve njegove korisnike, **bez** prikaza forme za izmenu (slika). Pored prikaza podataka o svakom klijentu stoji radio dugme za označavanje radi ažuriranja.

|  |  |
| --- | --- |
| CREATE DATABASE jun;  USE jun;  CREATE TABLE sistem (  id INT NOT NULL AUTO\_INCREMENT,  naziv varchar(30) NOT NULL,  vlasnik varchar(30) NOT NULL,  PRIMARY KEY(id)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  CREATE TABLE korisnik (  id INT NOT NULL AUTO\_INCREMENT,  korisnicko\_ime varchar(20) NOT NULL,  pol varchar(1) NOT NULL,  starost INT NOT NULL, | broj\_poruka INT NOT NULL,  sistem\_id INT NOT NULL,  PRIMARY KEY(id),  FOREIGN KEY (sistem\_id) REFERENCES sistem (id)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  INSERT INTO sistem (naziv, vlasnik) VALUES  ('Gmail', 'Google');  INSERT INTO korisnik (korisnicko\_ime, pol, starost,  broj\_poruka, sistem\_id) VALUES  ('Aleksandar', 'M', 21, 7, 1),  ('Jovana', 'Z', 24, 3, 1),  ('Petar', 'M', 30, 10, 1); |

1. Klikom na dugme „Izmeni“ se kroz AJAX/Fetch pribavljaju podaci o odgovarajućem korisniku i na osnovu vrednosti njegovih atributa, **popunjava** se i prikazuje forma za izmenu. Forma se nalazi u gornjem delu i omogućava ažuriranje postojećeg korisnika. Za pol koristiti radio dugmiće, za broj poruka numeric, a za broj godina select kontrolu (opseg vrednosti od 7 do 107).
2. Klik na dugme „Sačuvaj promene“ kroz AJAX/Fetch snima izmene u bazi podataka, ponovo učitava sve podatke iz baze, pakuje ih u JSON format, vraća ih klijentu i prikazuje ažurirane podatke u tabeli, a formu za izmenu je neophodno sakriti.
3. Ispod podataka o klijentima prikazati ukupan broj svih poruka.
4. **Sva komunikacija između klijenta i servera se obavlja bez prosleđivanja (submit) forme tj. stranica index.html se ne osvežava.**



**Napomena: Nije dozvoljeno korišćenje grafitne olovke.**

Predmetni asistenti