

Testiranje: PHP i SQL

1. Podesiti naslov dokumenta. Naslov je **"Testiranje"**.
 - Pod naslovom dokumenta misli se na **<title>** tag.
 - Koristiti **echo** instrukciju.
2. Ispisati **<h1>** element na stranici, sa sledećim sadržajem: **"Testiranje je počelo."**
 - Koristiti **echo** instrukciju.
3. Napiši PHP program koji zamenjuje vrednosti dve promenljive.
 - `x = 5`
 - `y = 10`
4. Napiši PHP program za izračunavanje sume dve celobrojne vrednosti. Ukoliko su dve vrednosti iste, program vraća njihov trostruki zbir.
5. Napiši PHP program koji ispisuje na stranici **true**, ukoliko je jedna od dve celobrojne vrednosti broj 30, ili ako im je suma 30 u suprotnom program ispisuje **false**.
 - ulaz: 30, 0
 - izlaz: true
 - ulaz: 25, 5
 - izlaz: true
 - ulaz: 20, 30
 - izlaz: true
 - ulaz: 20, 25
 - izlaz: false
6. Napisati PHP program koji će proveriti da li je jedna od dve celobrojne vrednosti manja od 0, a druga veća od 100.
7. Napisati PHP program koji će proveriti da li je jedna od dve celobrojne vrednosti u opsegu između 100 i 200. (uključujući 100 i 200) Ukoliko jeste program vraća **true**, a ukoliko nije program vraća **false**.
8. Napisati PHP program koji određuje najveći broj između tri celobrojne vrednosti i ispisati ga na stranicu.
 - Nije dozvoljeno korišćenje bilo koje ugrađene funkcije za određivanje najvećeg broja.
9. Napisati PHP program koji će proveriti koja od dve celobrojne vrednosti je bliža broju 100 i ispisati je. Ukoliko su brojevi jednaki, ispisati 0.
10. Napisati PHP program koji će ispisati na strinici **true** ukoliko su obe celobrojne vrednosti u opsegu između 40 i 50 (uključujući 40 i 50), ili su obe vrednosti u opsegu između 50 i 60 (uključujući 50 i 60), u suprotnom **false**.
11. Napisati PHP program koji će proveriti da li dva nenegativna broja imaju istu poslednju cifru. Ukoliko imaju, ispisati **true** na strinici, u suptornom **false**.

12. Definirati jednu promenljivu koja je tipa **string**. Napisati PHP program koji će:

- Pretvoriti sve karaktere stringa u velika slova.
- Pretvoriti sve karaktere stringa u mala slova.
- Pretvoriti prvi karakter stringa u veliko slovo.
- Pretvoriti prvi karakter svake reči u veliko slovo.

13. Napisati PHP program koji će proveriti da li **string1** sadrži u sebi **string2**. Ukoliko sadrži ispisati: "Reč je prisutna.", u suprotnom "Reč nije prisutna."

- string1 - PHP je najbolji programski jezik i volim da pišem PHP programe.
- string2 - PHP

14. Napisati PHP program koji briše sve 0 sa početka datog stringa.

- string - 000547023.24

15. Napisati PHP program koji će obrisati deo stringa.

- string: PHP je najbolji programski jezik i volim da pišem PHP programe.
- string koji se briše: PHP
- ispisati na stranici tako izmenjeni string

16. Definirati numerički niz. (statički) Vrednosti niza su brojevi od 1 do 5. Ispisati poslednji element niza.

17. Definirati asocijativni niz. (statički) Niz sadrži podatke o vama. Ime, prezime, grad u kome živite na sledeći način.

- Primer:
 - indeks ime, vrednost Petar
 - indeks prezime, vrednost Petrovic
 - indeks grad, vrednost Beograd

Ispisati podatke niza na sledeći način:

Ime: Petar

Prezime: Petrovic

Grad: Beograd

18. Za predefinisane brojeve **n** i **a** napisati kod koji sve brojeve od 1 do **n** koji su deljivi sa **a** smešta u numerički niz i prikazuje ga na stranici.

19. Kreirati numerički niz gde je svaki element novi asocijativni niz sa podacima o zaposlenima. Podaci u nizu su (0) ime i prezime, (1) mesto zaposlenja, (2) lokacija gde živi i (3) zarada. Niz treba da ima bar pet zaposlenih.

- a. Izlistati i prikazati sve zaposlene koji žive u Beogradu.
- b. Ispisati sve zaposlene koji imaju platu veću od 15000.
- c. Pronaći radnika sa najvećom zaradom i ispisati podatke o njemu.
- d. Pronaći srednju zaradu svih radnika.

20. Napisati PHP program koji će prikazati elemente niza u neodređenoj listi. Dati niz sadrži 3 elementa koji su stringovi: zelena, crvena i bela.

21. Napraviti niz koji će sadržati celobrojne vrednosti od 1 do 5. Obrisati bilo koji element niza tako da indeksi niza budu normalizovani.

Primer:

- `array(5) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) [4]=> int(5) }`
- Kada se obriše element sa indeksom [3] rezultat je:
- `array(4) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(5) }`

22. Napisati PHP program koji će na osnovu dva niza napraviti treći niz u kojem će **indeksi** biti vrednosti iz prvog niza, a **vrednosti** će biti vrednosti iz drugog niza.

- Prvi niz: `"field1" => "first", "field2" => "second", "field3" => "third"`
- Drugi niz: `"field1value" => "dinosaur", "field2value" => "pig", "field3value" => "platypus"`
- Izlaz treba biti: `"first" => "dinosaur", "second" => "pig", "third" => "platypus"`

23. Napisati PHP program koji će kreirati niz čije će vrednosti biti brojevi iz datog stringa.

- String: `"1 - 2 - 3 - 4 - 5"`
- Kreirati niz u kojem će biti samo parni brojevi i ispisati ga na stranicu.
- Kreirati niz u kojem će biti samo neparni brojevi i ispisati ga na stranicu.

24. Napisati kod koji deli predefinisani tekst po belom karakteru a zatim formira asocijativni niz gde je trenutna reč ključ sledećoj. Poslednja reč nije ključ.

Primer: `Lorem ipsum dolorem` [„Lorem“ => „ipsum“, „ipsum“ => „dolorem“]

25. Napisati PHP program koji će ispisati sve podatke o studentima iz datog JSON-a. Koristiti **foreach** za ispis podataka.

- JSON podatak:

```
[
    {
        "name" : "John Garg",
        "age" : "15",
        "school" : "Ahlcon Public school"
    },
    {
        "name" : "Smith Soy",
        "age" : "16",
        "school" : "St. Marie school"
    },
    {
        "name" : "Charle Rena",
        "age" : "16",
        "school" : "St. Columba school"
    }
]
```

26. Napraviti formu koja će imati jedno polje “input” i submit dugme. Koristite metodu “GET”.

- U polje “input” se unose brojevi odvojeni **razmakom** (space).
- Na submit forme, brojevi se sortiraju po rastućem poretку i upisuju nazad u polje “input” odvojeni zarezom.
- **Napomena:** kod tako sortiranih brojeva i odvojenih zarezom, posle poslednjeg broja zarez ne treba da postoji.

27. Napraviti formu koja će imati jedno polje “input” i submit dugme. Koristite metodu “POST”.

- U polje “input” unose se proizvoljna imena odvojena **razmakom** (space).
- Na submit forme, ispisuju se imena koja imaju više od 5 karaktera.
 - Imena se ispisuju u okviru neodređene liste.
 - Imena koja imaju više od 5 karaktera ispisuju se zelenom bojom.
 - Imena koja imaju manje od 5 karaktera ispisuju se crvenom bojom.
 - Imena koja imaju 5 karaktera ispisuju se sivom bojom.

28. Napisati PHP kod koji će da omogućiti upload fajla.

- Fajl mora imati ekstenziju .png.
- Fajl ne sme biti veći od 1MB.

29. Napraviti dve stranice: index.php i cookie.php. Na stranici **index.php** napravi formu sa dva select polja i submit dugmetom. U prvom selectu “korisnik” može da odabere boju pozadine, a u drugom selectu “korisnik” može da odabere boju slova na stranici. Klikom na submit dugme kreiraju se dva kolačica (oba kolačica traju 1 dan) i korisnik se redirektuje na stranicu cookie.php gde je podešena odabrana boja pozadine i boja slova na stranici. Stranica cookie.php sadrži dva <h4> taga i link ka index.php stranici, na kojoj takođe trebaju biti primenjene odabrane boje. U prvom <h4> tagu treba ispisati koju pozadinu je korisnik odabrao, a u drugom <h4> tagu treba ispisati koju boju slova je korisnik odabrao.

30. Napraviti funkciju koja će proveriti koji su brojevi prosti od 2 do n (n je ulazni parametar funkciji). Proste brojeve upisati u niz koji će funkcija vratiti kao rezultat. Na stranici se ispisuju elementi novokreiranog niza.

31. Napraviti funkciju koja će ispitati da li je string **palindrom**. Funkciji se prosledjuje string. Funkcija ispisuje rezultat na stranici. Ukoliko je string palindrom: “Palindrom”, ukoliko nije: “Nije palindrom”.

32. Kreirati bazu podataka i tabelu **users**. Tabela **users** sadrži sledeće kolone: id, username, password, role. Upisati u tabelu **users** 4 korisnika kroz PHP kod. Prikazati koliko je korisnika upisano uz pomoć **mysqli** funkcije i takođe ispisati **id** korisnika koji je poslednji upisan u bazu uz pomoć **mysqli** funkcije.

1. Marko, marko1, administrator
2. Janko, janko2, urednik
3. Stevan, stevan3, pretplatnik

- Prikazati sve korisnike na stranici. (prikazati sve kolone iz tabele)
 - Prikazati broj redova koji su isčitani iz baze uz pomoć **mysqli** funkcije.
- Promeniti lozinku korisniku Janko kroz PHP kod.
- Obrisati sve korisnike kroz PHP kod.

33. Iskoristiti tabelu users iz prethodnog zadatka, koja je sada prazna.

Kreirati sledece stranice:

- index.php
- login.php
- admin.php
- change.php
- delete.php
- functions.php
 - u ovom fajlu skladištimo sve potrebne funkcije
 - uključiti functions.php fajl svuda gde je potrebno uz pomoć PHP instrukcije

index.php

- navigacija treba da se prikaže kao neodređena lista
- kreirati funkciju u okviru functions.php fajla **render_header()**
 - funkcija nema ulazne parametre
 - ukoliko je korisnik ulogovan
 - funkcija uz pomoć switch uslova proverava da li je korisnik prijavljen kao “administrator”, ili kao običan “korisnik”
 - ukoliko je korisnik prijavljen kao administrator u neodređenoj listi se kao prvi element prikazuje username i uloga korisnika, zatim link ka početnoj strani, zatim link ka admin panelu, zatim link za odjavu
 - ukoliko je korisnik prijavljen kao običan korisnik u neodređenoj listi se kao prvi element prikazuje username i uloga korisnika, zatim link ka početnoj strani i link za odjavu (običan korisnik nema pristup admin panelu)
 - ukoliko korisnik nije ulogovan
 - u neodređenoj listi se kao prvi element prikazuje link ka početnoj strani zatim link ka stranici za prijavu
 - funkcija kao rezultat vraća html (html je zapravo “navigacija”)
- na stranici index.php se ispod navigacije prikazuje lista svih korisnika, napraviti funkciju **show_users()** koja kao ulazni parametar ne prima ništa, a funkcija vraća html kod (koji je zapravo prikaz svih korisnika za početnu stranu).

login.php

- login.php stranica sadrži jednu formu za prijavljivanje korisnika
- forma ima sledeća polja: **username**, **password**
- Kreirati funkciju u okviru functions.php fajla **login()**
 - funkcija kao ulazni parametar prima username i password iz forme
 - ukoliko je neko od dva polja prazno funkcija vraća odgovarajuću poruku
 - funkcija proverava da li korisnik postoji
 - ukoliko korisnik postoji, korisnik je uspešno prijavljen i vrši se redirekcija na index.php
 - ukoliko korisnik ne postoji funkcija vraća odgovarajuću poruku

admin.php

- ova stranica treba da sadrži formu za dodavanje korisnika (izbor uloge korisnika treba biti prikazan u okviru <select> taga. Prilikom unosa ukoliko neko polje nije popunjeno, ispisati odgovarajuću poruku.
- ova stranica sadrži listu svih korisnika, korisnici se prikazuju u tabeli
 - kod svakog korisnika postoji dugme za izmenu i brisanje korisnika

- kod za brisanje korisnika implementirati na stranici delete.php
 - kada se klikne na dugme za brisanje korisnika, id korisnika koji se briše se šalje kao GET parametar na stranicu delete.php. Kada se korisnik obriše vrši se redirekcija na admin.php
- dugme za izmenu korisnika
 - kada se klikne na dugme za izmenu korisnika, id korisnika koji se menja se šalje kao GET parametar na stranicu change.php

delete.php

- kod za brisanje korisnika

change.php

- ova stranica sadrži formu gde se učitavaju svi podaci korisnika koji se menja
- klikom na submit dugme, korisnikovi podaci treba da su izmenjeni i vrši se redirekcija na admin.php

Napraviti funkciju koja će služiti za **konekciju** sa bazom podataka. Funkcija ne prima ni jedan parametar, a vraća konekciju sa bazom podataka.

Link za odjavu služi za odjavljivanje korisnika sa sistema i vrši se redirekcija na index.php. Implementirati kod u funkciju **logout()** koja nema ulazne parametre i koja ne vraća ništa.

“Navigacija” se treba uključiti na sledeće stranice: index.php, admin.php, change.php.

Napomena:

- korisnik kada nije prijavljen, ne može da pristupi admin panelu (kroz URL)
 - redirekcija na početnu stranu
- korisnik kada je prijavljen, a ima ulogu pretplatnika takođe ne može da pristupi admin panelu (kroz URL)
 - redirekcija na početnu stranu
- ukoliko je bilo koji korisnik prijavljen, korisnik ne može da pristupi stranici za prijavljivanje, jer je već prijavljen (kroz URL)
 - redirekcija na početnu stranu

34. Napisati PHP kod koji ispisuje trenutni mesec i tri meseca unazad. Mesec se prikazuje slovima.

35. Napisati PHP kod koji ispisuje broj dana između dva datuma.

Primer:

- ulaz: 05.09.2021. i 10.09.2021.
- izlaz: 4

36. Napraviti klasu **Voće**, klasa treba da ima sledeća svojstva:

- **ime**
- **boja**

Treba da ima svoje metode **set_name** , **get_name**. Set_name metoda služi da podesimo ime za **Voće**.

Napraviti objekat **Jabuka**, pomoću set_name metode podesiti ime da bude "**Jabuka**".

Ispisati ime na ekran pomoću get_name metode.

37. Napraviti klasu **Voće**, **Jagoda**, **Jabuka**. Voće ima sledeća svojstva:

- **ime**
- **boja**

Ima **konstruktor**, s kojom možemo podesiti ime i boju Voća.

Ima metodu **getName**, koja ispisuje ime Voća.

Jagoda, i Jabuka nasledjuju sva svojstva od klase Voće.

Imaju **konstruktor**, s kojom možemo podesiti ime i boju...

Imaju metodu **getName** , koja ispisuje ime Jagode na ekran.

Napraviti dva objekta (Jagoda, Jabuka) podesiti ime Jabuke, i Jagode, ispisati ime na ekran.

38. Napraviti **abstraktnu** klasu **Auto**, klasa ima sledeća svojstva:

- **ime**
- **boja**

Ima **konstruktor** za podešavanje imena i godine Auta.

Definisati **abstrakt** metodu **intro** tako da može samo string da vraća.

Napraviti klasu **Audi**, **BMW**, oni nasledjuju sva svojstva od Auto klase.

Imaju metodu **intro**. Metoda za Audi ispisuje na ekran "Im Audi", za BMW "Im BMW".

Napraviti dva objekta Audi, BMW. Pozvati **intro** metodu na oba objekta.

39. Napraviti **interface** **Zivotinja**. Definisati metodu **makeSound**.

Napraviti **zivotinju** **Macka** i ispisati na ekran "Meow".

40. Cilj zadatka je pre svega da se pokaže dobro poznavanje **OOP** principa. Nije esencijalno da program radi. Zadatak treba realizovati bez dodatnih pitanja.

Napomena:

- Ne treba pisati bazu podataka
- Ne treba pisati HTML/CSS kod
- Akcenat je na OOP programiranju
- Ukoliko je moguće, potrebno je na odgovarajućim mestima implementirati:
 - Nasleđivanje
 - Interfejse
 - Polimorfizam
 - Exception handling

Klasom Voće se opisuje voćka, koja je određena svojom bojom i Zapreminom.

Jabuka je voće koje može biti crvljivo.

Sokovnik se sastoji iz dva dela: posude za voće i cediljke.

Posuda za voće ima svoj kapacitet i može da sadrži voće.

U posudu za voće mogu da dodam voćku, mogu da vidim koliko voćki ima unutra i koliko je prostora ostalo.

Cediljka služi za ceđenje voća. Pri svakom ceđenju mogu da vidim koliko soka je dobijeno.

Ceđenjem jedne voćke se dobija količina soka jednaka 50% zapremine Voćke.

Napisati program koji simulira rad sokovnika čija zapremina je 20 litara:

- a) Nad sokovnikom se vrše akcije, pri čemu se svaka akcija loguje u vidu ispisa na ekran.
- b) Sokovnik je programiran tako da uzastopno izvrši 100 akcija.
- c) Na svakih 9 ceđenja se doda još jedna jabuka.
- d) Jabuka ima zapreminu u intervalu od 1 do 5.
- e) Jabuka ima 20% verovatnoće da bude crvljiva.