

Semestrul II

Laborator săptămâna 10-11

02.05 - 06.05.2022

09.05 – 13.05.2022

OBIECTIVE: - Realizarea unor aplicații simple în care să se utilizeze PHP și MySQL.

1. Realizați o aplicație care are crează un formular ce preia informații despre mai multe persoane: numele, data nașterii (zi, luna, an), profesia și locul de muncă. Scrieți un script care să salveze aceste datele introduse într-un fișier. La încheierea introducerii datelor se va afișa conținutul fișierului în ordinea alfabetică a numelui. În plus pentru fiecare persoană se va afișa și vârsta.
2. Realizați o aplicație care are următoarele cerințe: **crearea unui formular de prelucrare a datelor, introducerea informațiilor într-o bază de date și vizualizarea acestora.**
 - formularul va avea următoarele câmpuri: nume (tipul text), prenume (tipul text), oraș (tipul text) și județ (tipul text); toate câmpurile sunt obligatorii.
 - datele introduse vor fi salvate într-o tabelă.
 - se vor afișa utilizatorii în ordine alfabetică după nume
 - se vor afișa în browser toți utilizatorii care sunt din Timișoara
 - se vor afișa în browser toți utilizatorii care sunt dintr-un județ care începe cu litera T
3. Să se creeze o tabelă numită Student care are următoarele câmpuri: **număr matricol, nume, denumire disciplină opțională, nota test, nota proiect, media, număr prezențe.** Să se realizeze o aplicație care permite:
 - Preluarea informațiilor pentru un student prin intermediul unui formular. Câmpurile se vor valida. (nota în intervalul 1-10, nr prezențe 0-14). Câmpul “media” se va calcula după formula $40\% \text{ nota test} + 60\% \text{ nota proiect}$. Denumirea cursurilor se preia dintr-o listă.
 - Salvarea datelor într-o tabelă MySQL
 - Afișarea studenților grupați pe disciplină în ordinea mediilor.
 - Afișarea mediei studenților pentru o disciplină dată.
 - Afișarea studenților care au mai mult de 10 prezențe.
 - Căutarea unui student după număr matricol sau nume.
 - Ștergerea unui student pe baza numărului matricol.

Exemple de scripturi PHP cu functii de prelucrare a fisierelor

```
<?php
    $filename = "fisier.txt";
    $handle = fopen ($filename, "r") or
    die("Eroare la deschidere");
    $contents = fread ($handle, 1024);
    fclose ($handle);
?>
```

```
<?php
    //scrie o linie noua in fisierul ex2.txt
    $filename = 'ex2.txt';
    $text = "O linie singura linie\n";

    $handle = fopen($filename, "w") or
    die("Eroare la deschidere");
    fwrite($handle, $text); fclose($handle);
    print "Success, am scris $text in fisierul $filename";
?>
```

```
<?
//afiseaza prima linie dintr-un fisier
$nume_fisier="exemplu.txt";
$handle=fopen($nume_fisier,"r") or
die("Eroare la deschidere");
$linie=fgets($handle,4096);
fclose($handle);
echo $linie;
?>
```

```
<?php
//scrie o linie noua la sfarsitul fisierului ex2.txt
$filename = 'ex2.txt';
$text = "O linie noua \n";
$handle = fopen($filename, "a") or
die("Eroare la deschidere");
fputs($handle, $text);
fclose($handle);
print "Success, am scris $text in fisierul $filename";
?>
```

```
<?php
    $nume_fisier="fisier.txt";
    $handle=fopen($nume_fisier,"r") or
    die("Eroare la deschidere");
    $n=0;
    while(!feof($handle))
    {
        $c=fgetc($handle);
        echo $c;
    }
    fclose($handle);
?>
```

int fprintf (resource handle string format [, mixed args])

Scrie un string formatat conform string-ului de formatare in fisierul referit de handle. Stringul de formatare este compus din una sau mai multe directive: caractere normale (fara de %) care sunt copiate direct la rezultat si specificatii de conversie, care descriu modul de conversie a celorlalti parametri.

Un specificator de conversie poate contine in principal:

% - un caracter procent.

b - argumentul este tratat ca un integer și prezentat ca un numar binar.

c - argumentul este tratat ca un integer și prezentat ca un caracter cu acel cod ASCII.

d - argumentul este tratat ca un integer și prezentat ca un numar zecimal cu semn.

u - argumentul este tratat ca un integer și prezentat ca un numar zecimal fara semn.

f - argumentul este tratat ca un numar real și prezentat ca un numar real.

o - argumentul este tratat ca un integer și prezentat ca un numar in baza 8.

s - argumentul este tratat ca un sir de caractere și prezentat ca un sir de caractere.

x - argumentul este tratat ca un integer și prezentat ca un numar in baza 16 (cu litere mici).

X - argumentul este tratat ca un integer și prezentat ca un numar in baza 16 (cu litere mari).

Exemple:

```
$a=65;
```

```
$b="un text";
```

```
$y="f";
```

```
$z=12.33;
```

```
fprintf($handle,"%d",$a); //scrie in fisierul referit de $handle numarul intreg 65;
```

```
fprintf($handle,"%b",$a); //scrie in fisierul referit de $handle numarul 65 in baza 2 adica
```

```
1000001
```

```
fprintf($handle,"%x",$a); // scrie in fisierul referit de $handle numarul 65 in baza 16 adica 41
```

```
fprintf($handle,"%c",$a); // scrie in fisierul referit de $handle caracterul care are codul ascii 65  
adica A
```

```
fprintf($handle,"%o",$a); // scrie in fisierul referit de $handle numarul 65 in baza 8 adica 101
```

```
fprintf($handle,"%f",$z); // scrie in fisierul referit de $handle numarul real 12.330000
```

```
fprintf($handle,"%0.2f",$z); // scrie in fisierul referit de $handle numarul real 12.33 doar cu doua  
zecimale dupa virgula.
```

```
fprintf($handle,"%s",$z); //scrie in fisierul referit de $handle textul 12.33.
```

```
fprintf($handle,"%s",$b); // scrie in fisierul referit de $handle textul un text.
```

```
fprintf($handle,"variabila a are valoarea %d, variabila b are valoarea %s",$a,$b);
```

```
//scrie in fisierul referit de $handle: variabila a are valoarea 65, variabila b are valoarea un text
```

fscanf (resource handle string format [string var1])

Funcția `fscanf()` este similară cu `sscanf()`, doar că citește dintr-un fișier referit de handle și interpretează stringul citit conform stringului de formatare specificat. Dacă se trimit doar doi parametri funcției aceasta va returna rezultatul sub forma unui vector.

Exemple:

```
<?php
$handle = fopen ("fisier.txt","r") or
die("Eroare la deschidere");
while ($info = fscanf ($handle, "%s\t%s\n"))
{
echo $info[0] . "<br>";
}
fclose($handle);
?>
```

pentru `fisier.txt`:

Ion Popescu
George Octavian
Vasile Ionescu

va afișa:

Ion
George
Vasile

int **fseek** (resource handle, int offset [, int whence])

Modifică poziționarea în fișierul referit de handle. Noua poziție măsurată în bytes de la începutul fișierului este obținută adăugând un număr de bytes la poziția specificată de *whence*, ale cărei valori sunt specificate mai jos:

SEEK_SET - Setează poziția egală cu numărul de bytes specificați de offset.

SEEK_CUR - Setează poziția egală cu poziția curentă plus offset.

SEEK_END - Setează poziția egală cu poziția sfârșitului de fișier plus offset (Pentru a ne muta la o poziție înainte de sfârșitul de fișier trebuie ca offset să aibă o valoare negativă.)

Dacă *whence* nu este specificat se presupune că este **SEEK_SET**. Dacă mutarea s-a realizat cu succes funcția returnează 0; altfel, returnează - 1.

Exemple:

```
<?php
    $handle = fopen ("exemplu.txt", "r");
    fseek($handle,2);
    fscanf($handle,"%c",$char); echo $char;
?>
```

Daca fisierul exemplu.txt contine:

abcdefsds
sdadad

exemplul va afisa:

c

Alte exemple:

Ex1:

```
<?php
//suma numerelor intregi dintr-un fisier, numerele fiind scrise cate unul pe o
linie
$nume_fisier="ex1.txt";
$handle = fopen ($nume_fisier, "r") or
    die("Eroare la deschidere");
$s=0;
while (!feof($handle))
{
    $buffer = fgets($handle, 4096);
    $s+=(int)$buffer;
}
echo $s;
fclose ($handle);
?>
```

Ex2:

```
<?
//afiseaza numarul de aparitii ale unui caracter intr-un fisier
$nume_fisier="ex2.txt";
$l='a';
$handle=fopen($nume_fisier,"r") or
    die("Eroare la deschidere");
$n=0;
while(!feof($handle))
{
```

```

    $c=fgetc($handle);
    if($c==$l) $n ++;
}
fclose($handle);
echo "Litera ($l) apare de $n ori in fisierul ($nume_fisier)"
?>

```

Ex3:

```

<?
//afiseaza numarul de aparitii al unui cuvant intr-un fisier
$nume_fisier="ex3.txt";
$cuvant='cuvant';
$handle=fopen($nume_fisier,"r") or
    die("Eroare la deschidere");
$n=0;
while(!feof($handle))
{
    $linie=fgets($handle,4096);
    $n += substr_count($linie, $cuvant); ;
}
fclose($handle);
echo "Cuvantul $cuvant apare de $n ori in fisierul $nume_fisier";
?>

```

Ex4:

```

<?
//Copiază conținutul primului fisier în cel de-al doilea.
$fișier1 = "origine.txt";
$fișier2 = "destinație.txt";
$handle1 = fopen($fișier1, "r") or
    die("Eroare la deschidere");
$handle2 = fopen($fișier2, "w") or
    die("Eroare la deschidere");
while (!feof($handle1))
{
    $buffer=fread($handle1,1024);
    fwrite($handle2,$buffer,1024);
}
fclose($handle1);
fclose($handle2);
echo "Am scris conținutul fișierului $fișier1 în fișierul $fișier2";
?>

```