Web2, curs07, Aplicatii

Nota: Notiunile teoretice se regasesc in capitolul 9 de pe www.invata-online.ro

1. Output Buffering. Functia *header()*

Notiuni teoretice

a) Output buffering

- In mod implicit, tot ceea ce un script PHP afiseaza si tot ceea ce inseamna cod HTML in afara scriptului, se transmite imediat catre browser.
- *Output buffering* este un instrument PHP care poate schimba acest comportament: ceea ce se transmite catre browser este va fi dus intr-o memorie temporara numita *buffer;* apoi, cand buffer-ul este eliberat (*flushed)*, continutul este trimis catre browser.
- Avantaje:
  - uneori se poate creste performanta folosind output buffering
  - elimina eroarea de tipul "*headers already sent*" care poate aparea la apelul unor functii precum *header( ), setcookie( ), session_start( )* (aceste functii pot fi apelate numai daca nu s-a transmis nimic catre browser)
- Functii:
  - *ob_start()* - activeaza buffer
  - *ob_get_contents()* - returneaza continutul memorat in buffer
  - *ob_flush( )* - trimite catre browser continutul bufferului si il elibereaza
  - *ob_clean()* - elibereaza buffer
  - *ob_end_clean()* - elibereaza buffer si il seteaza pe Off
  - *ob_end_flush( )* - trimite catre browser continutul bufferului, il elibereaza si il seteaza pe Off

b) Functia *header('string_header')*

- Ex de utilizare:
  - redirectare: *header('Location: cale/script.php')*;
  - trimitere de fisiere:
    - *header ("Content-Type: image/png\n");*
    - *header ("Content-Disposition: inline; filename=contact.png\n");*
    - *header ("Content-Length: 2456\n");*
    - *readfile("cale/nume_fisier");*
  - creare cooki-urilor
  - etc.
- Daca un script foloseste mai multe apeluri ale functiei header(), fiecare argument trebuie sa se termine cu newline (\n): *header("Content-Type:application/pdf\n"); header ("Content-Length: 4096\n");*
- Daca **nu** se foloseste *output buffering,* functia *header()* trebuie apelata inainte de a transmite ceva catre browser. Daca inainte de *header()* scriptul contine instructiuni echo sau print, are linii goale inafara etichetelor PHP sau include fisiere cu astfel de elemente, se va semnala un mesaj de eroare.

Aplicatie 1. Output buffering

a) rulati scriptul PHPinfo.php (contine apelul functiei phpinfo()) si verificati valoarea parametrului *output_buffering.* Daca are setata o valoare inseamna ca bufferul este activ in mod implicit. Cautati parametrul in *php.ini* si stabiliti-i valoarea *Off.*

b) Accesati scriptul *admin.php* si incercati o operatie de sterger. Scriptul *deleteuser.php* incearca dupa stergere o redirectare catre *admin.php.* Insa, transmitandu-se date catre browser (de, ex. *header.html*) inainte de utilizarea functiei *header(),* se semnaleaza un mesaj de eroare. Remediati utilizand Output Buffering.

c) Interveniti si in scriptul *edit_user.php* si faceti redirectare catre *admin.php* in caz de update cu succes.

Indicatie:

- Se va folosi functia *Header('Location: cale/script.php')*.
- Se va activa buffer-ul in *header.php - <?php ob_start(); ?>*
- Se va elibera buffer-ul in *footer.php- <?php ob_end_flush( ); ?>*

Aplicatie 2. Trimiterea unui fisier prin *header()*

a) Creati scriptul *list_images.php (Fisiere → Lista imagini)* care afiseaza sub forma de lista linkuri cu denumirile imaginilor din folderul ../uploads (folderul se afla inafara radacinii). Ca si informatie, se va afisa numele fisierului, dimensiunea acestuia in Kb si data la care a fost incarcat. La click, se va afisa imaginea asociata. Testati pe server (nu se pot accesa fisiere care nu sunt in radacina).

b) Testati scriptul *view_image.php (Fisiere → Script proxy)* adaugand querystring *image=fisier_imagine,* unde *fisier_imagine* este denumirea unui fisier imagine din ../uploads

c) Modificati scriptul *list_images.php* astfel incat vizualizarea imaginii sa se faca prin intermediul scriptului *view_image.php* care trimite catre browser fisierul prin headere. Testati pe server (asa ar trebui sa fie ok)

Indicatie: *echo "
$image $file_size ($image_date)
\n";*

Nota: Notiunile teoretice se regasesc in capitolul 9 de pe www.invata-online.ro

2. Lucrul cu fisiere in PHP. Upload. Mutarea fisierelor intr-un folder

I. Aplicatie.

Folosind functia *phpinfo()*, verificati valorile urmatoarelor variabile din php.ini care pot influenta uploadul de fisiere prin metoda POST

- "*file_uploads*" – variabila booleana care indica daca este permis sau nu uploadul de fisiere; trebuie sa fie *On*
- *"max_execution_time"* - determina numarul maxim de secunde in care un script poate rula inainte de a fi intrerupt automat. Se va creste valoarea acestei variabile daca se anticipeaza upload de fisiere mari sau viteza mica la transfer.
- *"max_input_time"* - controleaza timpul maxim permis pentru preluarea datelor de intrare, inclusiv uploadul de fisiere prin POST. Se va creste valoarea acestei variabile daca se anticipeaza upload de fisiere mari sau viteza mica la transfer. Valoare implicita 60s
- *"upload_max_filesize"* - determina dimensiunea maxima acceptata pentru un fisier la upload; are prioritate fata de campul ascuns MAX_FILE_SIZE; valoare implicita 2M
- *"post_max_size"* - determina dimensiunea maxima a datelor care pot fi transmise printr-o cerere POST (inclusiv upload de fisiere); trebuie sa fie cel putin cat *upload_max_filesize*; valoare implicita 8M
- *"upload_tmp_dir"* - determina directorul temporar folosit pentru upload-ul de fisiere

Va trebui intervenit in php.ini daca:

- *file_uploads* este Off
- PHP nu are un director temporar (numai daca se lucreaza in Windows)
- se doreste upload de fisiere mari (mai mari de 2MB, valoarea predefinita)

II. **Creat**i scriptul **upload.php** in care:

a) Veti avea un formular care permite incarcarea unui fisier. Pentru a vedea informatiile referitoare la fisierele incarcate, daca a fost transmis formularul, afisati cu *print_r()* continutul variabilei superglobale $_FILES.

Indicatii:

- Fisierele pot fi uploadate numai prin intermediul unui formular trimis cu metoda POST.
- In interiorul tagului form trebuie sa existe atributul ***enctype*** cu valoarea "***multipart/form-data***" pentru a avea siguranta ca fisierul este uploadat corect (caracterele din fisier nu sunt codificate)
- Formularul va contine si un camp de tip ***hidden*** cu ***name* = MAX_FILE_SIZE** si value dimensiunea maxima permisa la upload (in bytes); fisierele mai mari decat MAX_FILE_SIZE nu vor fi incarcate. Nu va avea efect daca va avea o valoare mai mare decat *upload_max_filesize* din php.ini

b) Daca a fost transmis fisierul, (este setata variabila *$_FILES['nume_fisier']*), se vor afisa cu print_r datele acestuia acestuia.

c) **(tema de lucru)** Daca am eroare la upload (codul erorii > 0), memorez mesajul de eroare in array-ul $errors

d) Daca nu am erori (array-ul $errors este gol), se va muta fisierul in folderul *uploads,* altfel voi afisa erorile.

- **Securitate:** se recomanda ca folderul *uploads* sa fie in afara directorului radacina (va avea drepturi de scriere si va putea fi accesat de un alt utilizator care are un site pe acelasi server; s-ar putea copia acolo un script php care sa creeze probleme; daca nu este in radacina, scriptul php nu va rula; o protectie se poate asigura si printr-un fisier .httaccess)
- Pentru a muta fisierul, se va folosi functia ***move_uploaded_file($tmp_name,$destination)*** unde ***$tmp_name*** este numele temporar al fisierului iar ***$destination*** este de forma ***$upload_path/nume_nou_fisier*** (***$upload_path*** este calea relativa catre folderul uploads iar ***nume_nou_fisier*** este numele pe care il dau fisierului (fara extensie) dupa ce il mut). Exemplu:

$upload_path = "../uploads";

$file_name = md5($tmp_name);$destination = $upload_path."/".$file_name;

- functia ***move_uploaded_file()*** returneaza TRUE daca fisierul a fost mutat cu succes; returneaza FALSE daca $tmp_name nu refera un fisier uploadat, valid; returneaza tot FALSE in cazul in care $tmp_name refera un fisier valid, dar nu s-a putut fac mutarea - de ex, folderul destinatie un are drepturi de scriere;
- daca fisierul nu se poate muta (fie functia nu a rulat, fie a rulat si a returnat false), se va memora mesajul de eroare in ***$errors***

e) Impuneti ca fisierele acceptate la upload sa fie numai de tip imagine .jpeg, .gif, .png,

- daca nu am eroare la upload, se va verifica daca “mime_type-ul” fisierului este in mutimea {'image/jpeg','image/gif','image/png'}
- in caz contrar, se memoreaza mesajul de eroare in $error
- mai multe despre mime type - http://www.freeformatter.com/mime-types-list.html

f) In caz de eroare la upload, semnalati eroarea standard generata;
- pentru a vedea erorile standard la upload, accesati ***http://www.php.net/manual/en/features.file-upload.errors.php***
- in cod puteti folosi urmatorul array:

```
//erorile standard

$standard_errors[1] = 'The uploaded file exceeds the upload_max_filesize directive in php.ini. ';

$standard_errors[2] = 'The uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the HTML form. ';

$standard_errors[3] = 'The uploaded file was only partially uploaded.';

$standard_errors[4] = 'No file was uploaded. ';

$standard_errors[5] = 'Missing a temporary folder';

$standard_errors[6] = 'Failed to write file to disk.';

$standard_errors[7] = 'File upload stopped by extension.';
```

III. Modificati scriptul **upload_1.php** astfel incat sa permita uploadul a cel putin doua fisiere.

IV. Modificati miniaplicatia *useri* astfel incat fiecare user sa aibe si o poza.
- Fisierul imagine se va incarca la inregistrare; in baza de date se va memora calea relativa catre fisier
- La afisarea utilizatorilor se va afisa si imaginea asociata

Nota: Notiunile teoretice se regasesc in capitolul 9 de pe www.invata-online.ro

3. Trimiterea unui email. Formular de contact

Notiuni teoretic
Functia *mail()* nu trimite emailul ci spune serverului de mail de pe calculatorul cu PHP sa faca asta. Deci, pentru a testa pe localhost, este necesar un server de mail. Daca site-ul este gazduit pe un server, ar trebui sa existe implicit un server de mail.
Sintaxa:
mail (to, subject, body, [headers]);

- *to – o adresa de email valida sau o lista de adrese separate prin virgula; o adresa de email valida este de forma **email@exemplu.com** sau **Nume <email@exemplu.com>***
- *subject – subiectul emailului*
- *body – continutul emailului*
- *exemplu:*

```
$to = 'email@exemplu.com';

$subject = 'Acesta este subiectul';

$body = 'Acesta este continutul.

Poate fi scris pe mai multe randuri.

';

mail ($to, $subject, $body);
```

- *bo*dy pe mai multe randuri se mai poate scrie folosind ghilimele duble astfel:
  *$bod*y = “Acesta este continutul. \r\nPoate fi scris pe mai multe randuri.”;
- o *linie* din body nu trebuie sa contina mai mult de 70 caractere; pentru a asigura aceasta conditie, se poate folosi functia wordwrap(): `$body = wordwrap($body, 70, "\r\n");`
- *headers* – este optional; aici se pot specifica datele pentru From, Reply-To, Cc, Bcc
  - exemplu: mail ($to, $subject, $body, 'From: popescu@exemplu.com');
- daca *se* folosesc mai multe headere, se vor separa prin \r\n
  - exemplu:
    $headers = "From: popescu@exemplu.com\r\n";
    $headers .= "Cc: ana@exemplu.com, mihai@exemplu.com\r\n";
    mail ($to, $subject, $body, $headers);

Aplicatie

- formular de contact
- preluarea si validarea datelor, pastrarea datelor in formular
- daca datele sunt valide, afisati in browser datele preluate din formular
- Cateva elemente de securitate:
  - in campul mesaj scrieti cod HTML; observati efectul
  - in campul mesaj scrieti urmatorul cod JS: <script type="text/javascript">alert("Bauuuuuuu!")</script>; observati efectul
  - aplicati functia *htmlentities()* sau *htmlspecialchars()* valorii preluate din campul mesaj; reluati testele anterioare
- trimiteti prin email catre propria adresa mesajul preluat din formular; testati pe server

Nota: Notiunile teoretice se regasesc in capitolul 9 de pe www.invata-online.ro

4. **Accesarea fisierelor. Deschidere, citire, scriere, inchidere;**

Aplicatii. Lucru independent

1. Creati fisierul *date.txt* care va contine cateva linii de text (puteti lua text de pe www.lipsum.com)
2. creati un script php care va citi fisierul si va afisa in browser, sub forma de lista numerotata, liniile de text. Pentru citirea din fisier folositi functia ***fgets()*** (vezi php.net/fgets). Putei aplica urmatorul algoritm:
   - se verifica existenta fisierului: ***file_exists($filename)***
   - daca fisierul exista,
     - se deschide fisierul pentru citire: ***$fp=fopen($filename,'r');***
     - daca am date in fisier: ***filesize($filename)***>0
       - cat timp am o linie de citit: *while($line=**fgets($fp)**)*
         - afisez linia citita
3. Modificati scriptul anterior astfel incat sa folositi functia ***file($filename)*** care returneaza continutul fisierului sub forma de array. Elementele array-ului vor fi liniile fisierului
4. Creati un script php care va citi tot continutul fisierului folosind functia ***fread($fp,$length)*** care citeste $length bytes din fisier. Cu functia ***filesize($filename)*** se va determina dimensiunea in bytes a fisierului.
5. Creati un script php care va citi tot continutul fisierului folosind functia ***file_get_contents($filename)*** care returneaza intr-un string continutul fisierului.
6. Creati un script php care va citi tot continutul fisierului date.txt si-l va transmite catre browser folosind functia ***readfile($filename).***
7. Creati prin intermediul unui script php un fisier CSV (Comma Separated Value) emails.csv pe baza arrayului *$emails = array(array('Ion Popa', 'ipopa@gmail.com'),array('Ana Stefanescu', 'astefanescu@gmail.com'), array('Mihai Cucu', 'mcucu@gmail.com'),array('Mihaela Ion', 'mion@gmail.com'));*
   Pentru scrierea in fisier se poate folosi functia ***fputcsv (resource $handle , array $fields [, string $delimiter = ',' [, string $enclosure = '"']])*** care scrie intr-un fisier, pe aceeasi linie, elementele unui array separat prin $dekiniter.
8. Creati un script php care va citi fisierul *emails.csv* folosind functia ***array fgetcsv (resource $handle [, int $length = 0 [, string $delimiter = ',' [, string $enclosure = '"' [, string $escape = '\\']]]])*** care preia intr-un array elementele de pe o linie a fisierului. Se vor transmite catre browser numele si emailul fiecarei persoane.
9. (optional) Creati fisierul emails.csv folosind functiile ***fwrite (resource $handle , string $string [, int $length])*** si respectiv int ***file_put_contents (string $filename , mixed $data [, int $flags = 0 [, resource $context]])***
10. Testati pe server scriptul trimite_email.php care trimite un email la o lista de persoane.
11. Creati un script php care trimite un email la toate persoanele din fisierul emails.csv.

Nota: Notiunile teoretice se regasesc in capitolul 9 de pe www.invata-online.ro

5. **Lucrul cu fisiere si directoare in PHP. Deschiderea, inchiderea, scrierea si cititrea fisierelor. Citirea unui director. Tema de lucru**

1. Creati scriptul **export.php** in care:
   a) daca nu exista, se va crea folderul specificat de variabila $path
      - se va folosi functia ***bool file_exists (string $filename)*** care verifica existenta unui fisier sau folder
      - se va folosi functia ***bool mkdir (string $pathname [, int $mode= 0777 [, bool $recursive= false]])*** pentru crearea unui director
   b) in folderul specificat de variabila $path, se va crea fisierul **utilizatori.csv** care va contine datele din tabelul utilizatori din baza de date myapp, cate o inregistrare pe fiecare linie, campurile unei inregistrari fiind separate prin virgula.

2. Creati scriptul **readFile.php** care va citi datele din fisier si le va afisa sub forma de tabel.
3. In folderul curent creati folderul img in care sa aveti si 6 fisiere jpg.
   a) Creati scriptul **dir1.php** care va deschide folderul img si va afisa continutul acestuia
      - $d=***opendir($dir)***; - deschide directorul $dir
      - ***readdir($d)***; - citeste o intrare - un fisier, un director, sau fisierele speciale '.' si '..'; returneaza numele intrarii sau FALSE daca s-a ajuns la final
   b) Creati scriptul **dir2.php** care va deschide folderul img si va afisa, pentru fiecare intrare, valoarea returnata de functia ***pathinfo()***
      - ***mixed pathinfo (string $path)*** - functia primeste o cale catre un fisier, si returneaza intr-un array componentele ei.
   c) Creati scriptul **dir3.php** care va deschide folderul img si va afisa in browser imaginile la dimensiune 100x100.