



**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

---

## **Προγραμματισμός Διαδικτύου**

### **Ασκήσεις Εργαστηρίου**

**Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 06**

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto:mdasyg@ieee.org)

**Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών**

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

[http:// arch.ict.e.uowm.gr/mdasyg](http://arch.ict.e.uowm.gr/mdasyg)

---

## Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



**Έκδοση Σεπτέμβριος 2011**

## Περιεχόμενα

1. Σκοπός της άσκησης..... 3
2. Ερωτήσεις/Ασκήσεις..... 3

## 1. Σκοπός της άσκησης

Αποσφαλμάτωση στοιχείων υποβολής φόρμας. Επεξεργασία των στοιχείων φόρμας που υποβάλλεται. Πολλαπλές ταυτόχρονες μεταμορφώσεις. Δημιουργία μικρογραφιών εικόνων. Στοιχεία ασφάλειας για μεταμόρφωση αρχείου.

## 2. Ερωτήσεις/Ασκήσεις

Δημιουργήστε ένα φάκελο με το όνομα lab06 μέσα στο htdocs του xamp και τοποθετήστε μέσα σε αυτόν τα αρχεία που είχαν δημιουργηθεί στο προηγούμενο εργαστήριο.

Επισκεφτείτε τη σελίδα <http://localhost/lab06> και επιβεβαιώστε ότι εμφανίζεται σωστά ο ιστοχώρος.

### Κατασκευή της σελίδας χειρισμού της φόρμας προσθήκης αξιολόγησης add\_process.php

- Αντιγράψτε το αρχείο `template.php` στο αρχείο `add_process.php` μέσα στον κατάλογο lab06

Μέσα στο τμήμα `maincontent` θα τοποθετήσουμε τον κώδικα χειρισμού της φόρμας.

- Πηγαίνετε στο Προσθήκη αξιολόγησης, συμπληρώστε ενδεικτικά τη φόρμα, επιλέξτε ένα αρχείο για αποστολή και πατήστε 'ΑΠΟΣΤΟΛΗ'. Θα μας εμφανιστεί η σελίδα `add_process.php` η οποία ασφαλώς δεν έχει καθόλου περιεχόμενο. Μην κλείσετε το παράθυρο. Θα κάνουμε αλλαγές στον κώδικα `add_process.php` και θα πατάτε ανανέωση για να επαναλαμβάνεται η αποστολή της φόρμας.
- Αρχικά θέλουμε να δούμε κατά την αποστολή τι στοιχεία μεταδίδονται με τη φόρμα. Για να το πετύχουμε αυτό μέσα στο `maincontent` θα χρησιμοποιήσουμε τη συνάρτηση `var_dump` με όρισμα τον πίνακα `$_POST`, και στη συνέχεια τον πίνακα `$_FILES` ο οποίος έχει το αρχείο που ανεβάζουμε ως εξής:

```
<?php  
var_dump($_POST);  
var_dump($_FILES);  
?>
```

- Αποθηκεύστε το αρχείο `add_process.php` και πατήστε στο παράθυρο του φυλλομετρητή **ανανέωση** για να σταλούν πάλι τα δεδομένα της φόρμας στο αρχείο `add_process.php`. Παρατηρήστε τα ονόματα των μεταβλητών που εμφανίζονται και τα περιεχόμενα.
- Εναλλακτικά, αντί για `var_dump()` χρησιμοποιήστε τη συνάρτηση `print_r()` και δείτε τα αποτελέσματα με ανανέωση στο φυλλομετρητή. Όποια συνάρτηση από τις 2 σας είναι πιο εύκολη να τη χρησιμοποιείτε κατά την ανάπτυξη σελίδων php.
- Αφού σημειώσετε τα ονόματα των μεταβλητών που μεταδίδονται μέσω `$_POST` και `$_FILES` θα τα περάσετε σε τοπικές μεταβλητές. Γράψτε τις παρακάτω γραμμές και **συμπληρώστε** τα κενά:

```

$category=$_POST['category'];
$perfecture=
$elementname=
$rating=
$review=
$filename[0]=$_FILES['image_file']['name'];
$tmpfilename[0]=
$size[0]=

echo "<p>Category=$category, perfecture=$perfecture, rating=$rating,
elementname=$elementname,
Review=$review, filename=$filename[0], tmpfilename=$tmpfilename[0], size=$size[0]";

```

Σημειώστε ότι το `$_POST['category']` είναι το ίδιο με το `$_POST[category]` (δηλ. χωρίς εισαγωγικά). Αν όμως το όνομα της μεταβλητής έχει κενά, τότε θα δημιουργηθεί πρόβλημα. Είναι λοιπόν μια καλή πρακτική να βάζουμε το index **πάντα σε εισαγωγικά**.

- Μόλις επιβεβαιώσετε ότι εμφανίζονται από το echo τα σωστά στοιχεία που συμπληρώσατε στη φόρμα τότε βάλτε σε σχόλια τις γραμμές echo,var\_dump.
- Αρχικά θα πρέπει να ελέγξουμε ότι οι 4 πρώτες μεταβλητές είναι συμπληρωμένες με μια τιμή. Για να το κάνουμε αυτό θα χρησιμοποιήσουμε τη συνάρτηση empty().
  - Γράψτε μια δομή if η οποία θα ελέγχει τις 4 πρώτες μεταβλητές και αν δεν έχουν συμπληρωθεί τότε θα εμφανίζει το κατάλληλο μήνυμα. ( `if(empty($category) || ..... )` )

```

{ die("Πρέπει να συμπληρώσετε όλα τα πεδία της φόρμας. Πηγαίνετε στην προηγούμενη σελίδα για να το κάνετε.");}

```
- Πηγαίνετε στο φυλλομετρητή και αφήστε κενά κάποια πεδία, και επιβεβαιώστε ότι λειτουργεί σωστά η δομή ελέγχου.
- Επειδή θέλουμε να επιτρέπουμε το χρήστη να μπορεί να ανεβάσει έως 3 αρχεία εικόνων, θα πρέπει να τροποποιήσουμε το αρχείο add.php και να **τοποθετήσουμε ακόμη δύο στοιχεία** στη φόρμα τύπου αρχείου (type="file"). Όμως το όνομα της μεταβλητής δε θα είναι το ίδιο, γιατί θα έχει ως συνέπεια στο τέλος να μεταφέρεται μόνο το τελευταίο αρχείο. Θα πρέπει να αλλάξουμε το όνομα της μεταβλητής σε πίνακα. Δηλαδή να προσθέσουμε στο όνομα κάθε input τύπου αρχείου τις αγκύλες [ ]
  - Απομακρύνετε τα σχόλια από το var\_dump ή το print\_r(\$\_FILES) για να δείτε τώρα τι στέλνει η φόρμα στο add\_process.php , τροποποιήστε τις μεταβλητές \$tmpfilename[], \$filename[], \$size[] κατάλληλα και προσθέστε τις μεταβλητές \$tmpfilename[1], \$tmpfilename[2], \$filename[1],\$filename[2], \$size[1], \$size[2] οι οποίες θα συνδέονται με τα νέα αυτά πεδία.
  - Απομακρύνετε τα σχόλια από το echo για να επιβεβαιώσετε ότι ορίσατε σωστά τις μεταβλητές. Θα πρέπει από τη φόρμα εισαγωγής αξιολόγησης να τοποθετήσετε και στα 3 κουτιά μεταφόρτωσης αρχείου κάτι και να πατήσετε αποστολή.
  - **Αφού επιβεβαιώσετε** ότι στο echo εμφανίζονται όλες οι μεταβλητές με τιμές, τοποθετήστε σχόλια στις εντολές var\_dump/print\_r, echo.

==> Η μεταφόρτωση αρχείων μέσω φόρμας με το πρωτόκολλο HTTP POST, δημιουργεί ένα αρχείο προσωρινά στο διακομιστή στη θέση tmpname όπως φαίνεται στο `$_FILES` . Στη συνέχεια ο χρήστης θα πρέπει να αντιγράψει το αρχείο αυτό στην τελική τοποθεσία, γιατί το προσωρινό αυτό αρχείο θα απομακρυνθεί στο επόμενο διάστημα.

- Ο πρώτος έλεγχος ασφαλείας που θα πρέπει να κάνουμε είναι να επιβεβαιώσουμε ότι τα `tmpfilename[0]`, `tmpfilename[1]`, `tmpfilename[2]` είναι αρχεία που ήρθαν μέσω HTTP POST και δεν προέρχονται από παραπλάνηση της φόρμας από κάποιον hacker. Θα χρησιμοποιήσουμε τη συνάρτηση `is_uploaded_file()`; η οποία δέχεται ένα όρισμα κάθε φορά (το προσωρινό αρχείο) και επιστρέφει TRUE αν πράγματι το προσωρινό αυτό αρχείο ήρθε μέσω HTTP POST. (Ένας hacker θα μπορούσε να ξεγελάσει το `tmp_name` να δείχνει σε σημαντικά αρχεία ρυθμίσεων του συστήματος όπως το αρχείο των κωδικών `/etc/passwd` στο UNIX).

==> Δημιουργήστε μια δομή if ελέγχου η οποία θα ελέγχει ταυτόχρονα (AND) αν η μεταβλητή `$filename[]` έχει τιμή (με τη συνάρτηση `isset()`;) και αν το `$tmpfilename[]` έχει προέλθει από HTTP POST (με τη συνάρτηση `is_uploaded_file`). Αν κάποια από τις παραπάνω συναρτήσεις επιστρέψει FAIL τότε θα εκτελείται μια εντολή `die('Παρουσιάστηκε ένα πρόβλημα κατά τη μεταφόρτωση του αρχείου');`

==> Τοποθετήστε άλλες δυο δομές ελέγχου, οι οποίες παρομοίως θα ελέγχουν τα 2 επόμενα αρχεία.

==> Επιβεβαιώστε με το φυλλομετρητή ότι τα 3 αρχεία που μεταφορτώνονται δεν έχουν πρόβλημα με τις δομές if

==> Μπορείτε να καταλάβετε γιατί χρησιμοποιούμε 3 διαφορετικές δομές if για να ελέγχουμε τα 3 αρχεία; Μπορούμε να χρησιμοποιήσουμε 1 δομή if για όλους αυτούς τους ελέγχους; ΝΑΙ...Κάντε το. (θα ενώσετε κάθε συνθήκη με το Η' (OR - ||))

==> Επιβεβαιώστε με το φυλλομετρητή ότι τα 3 αρχεία που μεταφορτώνονται δεν έχουν πρόβλημα με τη μια δομή if

- Ο δεύτερος έλεγχος ασφαλείας είναι ότι θέλουμε ο χρήστης να στέλνει αρχεία JPG ή BMP (εικόνες) μόνο. Για να το επιτύχουμε αυτό θα πρέπει να ελέγχουμε το `filename` κάθε φορά ότι είναι όνομα που έχει τη συγκεκριμένη κατάληξη. Θα χρησιμοποιήσουμε μια κανονική έκφραση για αυτό και τη συνάρτηση `preg_match`. Η κανονική έκφραση που χρειαζόμαστε είναι η `'\.(jpg|bmp)$/i'`

Δηλαδή, να υπάρχει τελεία (τοποθετήσαμε το διαγώνιο για να απομακρύνουμε την ειδική σημασία της τελείας ως `\.` γιατί δε θέλουμε η τελεία να έχει την ειδική σημασία του ΕΝΑΣ ΟΠΟΙΟΣΔΗΠΟΤΕ ΧΑΡΑΚΤΗΡΑΣ, στη συνέχεια να υπάρχει είτε `jpg` είτε το `bmp` και να τελειώνει το όνομα (προσοχή στη χρήση του `$`). Επίσης χρησιμοποιείται η παράμετρος `/i` για να δηλώσουμε ότι δε μας ενδιαφέρει το case, δηλαδή η διαφοροποίηση σε μικρά ή κεφαλαία. Έτσι, η παραπάνω έκφραση θα ταιριάζει είτε με ένα αρχείο `.JPG` είτε με ένα αρχείο `.jpg` είτε ακόμη και με `.JpG`.

- Αυτή τη φορά θα κάνουμε τον έλεγχο μέσα σε μια δομή `for`. Θα χρησιμοποιήσουμε ένα μετρητή `$number_of_files` ο οποίος θα έχει το συνολικό αριθμό αρχείων που έχουμε ανεβάσει. Θα χρησιμοποιήσουμε τη συνάρτηση `count()` η οποία δέχεται ένα όρισμα που αντιστοιχεί σε ένα πίνακα. Μετράει τα στοιχεία του πίνακα και επιστρέφει μια ακέραια τιμή. Εμείς θα μετρήσουμε τα στοιχεία του πίνακα:

```
$tmpfilename[]
```

Όσα προσωρινά αρχεία έχουν δημιουργηθεί τόσα αρχεία θα έχουν ανέβει. Για να επιβεβαιώσουμε ότι συμπληρώσαμε σωστά τη μεταβλητή θα τοποθετήσουμε μια εντολή `echo`. Συμπληρώστε λοιπόν τη γραμμή, και επιβεβαιώστε στο φυλλομετρητή ότι αν ανεβάσετε 3 αρχεία η μεταβλητή `number_of_files` θα έχει την τιμή 3:

```
$number_of_files=count(.....);
```

```
echo $number_of_files;
```

- Δημιουργήστε μια δομή if με τιμές από 0 έως το `$number_of_files` όπου θα ελέγχει με το `empty()` αν δεν έχει κενή τιμή το `$filename[$i]` και στη συνέχεια (αν υπάρχει αρχείο) να ελέγχει την κατάληξη με την κανονική έκφραση ως (προσθέστε ένα if που να ελέγχει με το `empty` αν υπάρχει το `$filename[$i]`):

```
if (!preg_match('\.(jpg|bmp)$/i', $filename[$i])) { die("Παρακαλώ μεταφορτώστε μόνο εικόνες jpg ή bmp");}
```

- Εκτός από την κατάληξη του αρχείου θέλετε να ελέγξετε ότι το μέγεθος του κάθε αρχείου είναι μικρότερο από ένα προκαθορισμένο μέγεθος (π.χ. 10MB). Θα πρέπει

λοιπόν να δημιουργήσουμε ένα for loop το οποίο θα ελέγχει το μέγεθος κάθε αρχείου που έχει μεταφορτωθεί (πίνακας size[]).

- Δημιουργήστε μια δομή if με τιμές από 0 έως το \$number\_of\_files όπου θα ελέγχει με το empty() αν δεν έχει κενή τιμή το \$filename[\$i] και στη συνέχεια αν το μέγεθος κάποιου αρχείου είναι μεγαλύτερο από 10MB θα το σβήνουμε και θα ενημερώνουμε κατάλληλα το χρήστη.

==>Το μέγεθος θα το ελέγχουμε από τον πίνακα \$size[\$i]

==>Η διαγραφή του αρχείου αν είναι μεγαλύτερο από 10MB θα γίνεται μέσω της συνάρτησης unlink() ως εξής (να μπει μέσα στον κατάλληλο βρόγχο if):

**echo "Το αρχείο \$filename[\$i] είναι μεγαλύτερο από το μέγιστο επιτρεπόμενο όριο (10MB). Το αρχείο αυτό δε γίνεται αποδεκτό" ;**

**unlink(\$tmpfilename[\$i]);**

- Σε περίπτωση που θέλουμε να δούμε το μέγιστο μέγεθος αρχείου που επιτρέπεται από τις ρυθμίσεις του διακομιστή μπορούμε να χρησιμοποιήσουμε την παρακάτω εντολή:

**echo "<br>".ini\_get('post\_max\_size')."<br>";**

- Δοκιμάστε την παραπάνω γραμμή και δείτε ποιο είναι το μέγιστο μέγεθος του διακομιστή που χρησιμοποιείται. Αφού το δείτε, τοποθετήστε το σχόλιο από μπροστά για να μην εκτελείται.
- Αφού ολοκληρωθούν οι παραπάνω έλεγχοι, θα πρέπει να μετακινήσουμε το προσωρινό αρχείο στην τελική του διεύθυνση μέσα στον κατάλογο του ιστοχώρου μας. Πρέπει να το μετακινήσουμε στον κατάλογο για 2 λόγους:
  - το προσωρινό αρχείο θα διαγραφεί αυτόματα μετά από κάποιο διάστημα,
  - το προσωρινό αρχείο δεν είναι διαθέσιμο για προβολή στην (προσωρινή) τοποθεσία που βρίσκεται.

==>Δημιουργήστε έναν κατάλογο userimages μέσα στον κατάλογο του εργαστηρίου.

==>Δημιουργήστε επίσης, έναν κατάλογο thumbnails μέσα στον κατάλογο userimages που δημιουργήσατε στο προηγούμενο βήμα. Ο κατάλογος αυτός θα φέρει μια σμίκρυνση της φωτογραφίας που θα έχει ανεβάσει ο χρήστης. Θα μπορούσαμε κάθε φορά να κατασκευάζουμε δυναμικά τις μικρογραφίες των εικόνων αλλά θα επιλέξουμε την καλύτερη μέθοδο (γιατί είναι καλύτερη;) να τις δημιουργούμε μια φορά μόνο.

- Πριν μετακινήσουμε μια εικόνα θα πρέπει να ελέγξουμε αν το αρχείο προορισμός υπάρχει. Αν υπάρχει τότε η μετακίνηση του αρχείου θα έχει ως συνέπεια να διαγραφεί το υπάρχον αρχείο. Ο έλεγχος θα γίνει με τη συνάρτηση file\_exists(); η οποία δέχεται ένα όνομα αρχείου ως όρισμα και επιστρέφει TRUE αν υπάρχει αυτό το αρχείο, διαφορετικά FALSE.
- Δημιουργήστε τις παρακάτω δύο μεταβλητές οι οποίες θα φέρουν την ακριβή τελική τοποθεσία του καταλόγου προορισμού. Παρατηρήστε ότι χρησιμοποιούμε το DOCUMENT\_ROOT για να προσδιορίσουμε την τοποθεσία htdocs του xamp:

**\$userimages=\$\_SERVER{'DOCUMENT\_ROOT'}."/lab06/userimages/";**

**\$thumbnails=\$\_SERVER{'DOCUMENT\_ROOT'}."/lab06/userimages/thumbnails/";**

- Δημιουργήστε μια δομή if με τιμές από 0 έως το \$number\_of\_files όπου θα ελέγχει με το empty() αν δεν έχει κενή τιμή το \$filename[\$i] και στη συνέχεια θα ελέγχει με τη συνάρτηση file\_exists(); κατά πόσο υπάρχει το αρχείο \$images/\$filename[\$i]
  - Αν υπάρχει θα εμφανίζει ένα μήνυμα ότι το αρχείο \$images/\$filename[\$i] υπάρχει.
  - Αν δεν υπάρχει τότε θα χρησιμοποιείται η συνάρτηση move\_uploaded\_file(); η οποία θα μετακινεί το προσωρινό αρχείο στην τελική της τοποθεσία ως εξής:

```
move_uploaded_file( "$tmpfilename[$i]", "$images/$filename[$i]" );
```

Επειδή, πρέπει να ενημερώνουμε το χρήστη ότι το αρχείο μεταφορτώθηκε (δεν είναι σωστό να αφήνουμε το χρήστη να μαντεύει αν κάτι λειτούργησε σωστά ή όχι) τοποθετήστε και το μήνυμα:

```
echo "Το αρχείο $filename[$i] μεταφορτώθηκε με επιτυχία";
```

- Δοκιμάστε στο φυλλομετρητή να ανεβάσετε μια εικόνα μερικές φορές και επιβεβαιώστε ότι λειτουργεί σωστά η ανωτέρω δομή.
- Στην προηγούμενη περίπτωση, αν το αρχείο υπάρχει δεν τοποθετείται στην τελική του τοποθεσία. Επειδή, είναι καλή πρακτική ο προγραμματιστής να προλαμβάνει και να διορθώνει τέτοια θέματα, θα δημιουργήσουμε μια δομή while η οποία θα ψάξει να βρει ένα όνομα αρχείου που δεν υπάρχει για να τοποθετήσει το αρχείο, ώστε να μη χαθεί το upload. Θα προσθέσουμε λοιπόν τον παρακάτω κώδικα στην ανωτέρω δομή στο σημείο που αναγράφεται ότι το αρχείο υπάρχει.
  - Δημιουργήστε μια μεταβλητή \$name\_prepend η οποία αρχικά θα έχει τιμή 0.
  - Ορίστε τη μεταβλητή \$destination\_file\_name η οποία θα φέρει ένα όνομα στο οποίο το πρώτο κομμάτι είναι ένας αριθμός και το υπόλοιπο είναι το όνομα που έχει δώσει ο χρήστης, ως:

```
$destination_file_name=$name_prepend." ".$filename[$i];
```

- Στη συνέχεια μέσα σε μια δομή while

```
while (file_exists("$userimages/$destination_file_name")) {...
```

θα αυξάνεται η τιμή του \$name\_prepend και θα επανακαθορίζεται το \$destination\_file\_name και αυτό θα γίνεται έως ότου δεν υπάρχει το \$userimages/\$destination\_file\_name

- Όταν ολοκληρωθεί το ανωτέρω loop τότε θα έχουμε βρει ένα όνομα αρχείου το οποίο δεν υπάρχει και μπορούμε να το χρησιμοποιήσουμε ως εξής:

```
move_uploaded_file( "$tmpfilename[$i]", "$userimages/$destination_file_name" );
```

```
$filename[$i]=$destination_file_name;
```

```
echo "Το αρχείο $filename[$i] μεταφορτώθηκε με επιτυχία";
```

- Το τελευταίο κομμάτι είναι να κατασκευάσουμε τη μικρογραφία της εικόνας που έχει ανεβάσει ο χρήστης και να την τοποθετήσουμε στον κατάλογο \$thumbnails
- Η PHP διαθέτει αρκετές συναρτήσεις για το χειρισμό εικόνων. Όμως θα χρειαστούμε τη συνάρτηση ImageCreateFromBmp() η οποία δεν υπάρχει στην τυπική εγκατάσταση. Επειδή θα χρειαστούμε αυτή τη συνάρτηση πηγαίνετε στην παρακάτω σελίδα και αντιγράψτε τον κώδικα της συνάρτησης ImageCreateFromBmp() και τοποθετήστε την στο αρχείο site\_functions.php <http://www.php.net/manual/en/function.imagecreate.php#53879>
  - Δημιουργήστε μια δομή if με τιμές από 0 έως το \$number\_of\_files όπου θα ελέγχει με το empty() αν δεν έχει κενή τιμή το \$filename[\$i] και στη συνέχεια θα εκτελούνται μια σειρά από εντολές οι οποίες θα κλιμακώνουν τις εικόνες για να δημιουργήσουν τις μικρογραφίες σε 80x60.
  - Αρχικά θα τοποθετήσουμε τις μέγιστες διαστάσεις που θέλουμε να έχουν οι μικρογραφίες μας:

```
if (!$max_width)
```

```
$max_width=80;
```

```
if (!$max_height)
```

```
$max_height=60;
```



- Στη συνέχεια καλούμε τη συνάρτηση GetImageSize() η οποία δέχεται ένα όρισμα (αρχείο εικόνας) και επιστρέφει έναν πίνακα με τιμές (διαστάσεις, μέγεθος, χρώματα κτλ. Η τιμή με index=0 έχει το πλάτος ενώ η τιμή με index=1 έχει το ύψος). :

```
$size=GetImageSize("$userimages/$filename[$i]");
```

```
$width=$size[0];
```

```
$height=$size[1];
```

- Το κλειδί για την αλλαγή μεγέθους είναι ο υπολογισμός των νέων παραμέτρων πλάτους και ύψους. Βρίσκετε την αναλογία μεταξύ των πραγματικών και των μέγιστων διαστάσεων:

```
$x_ratio=$max_width / $width;
```

```
$y_ratio = $max_height / $height;
```

- Αν η εικόνα είναι ήδη μικρότερη από τις μέγιστες διαστάσεις, το πλάτος και το ύψος της μένουν όπως είναι:

```
if (($width <=$max_width) && ($height <=$max_height)) {
```

```
$tn_width=$width;
```

```
$tn_height=$height;
```

```
}
```

- Διαφορετικά, χρησιμοποιείται η αναλογία X ή Y για να κλιμακωθούν και οι δύο διαστάσεις εξίσου, έτσι ώστε η εικόνα με το μειωμένο μέγεθος να μη δείχνει παραμορφωμένη ή συρρικνωμένη, με τον εξής τρόπο:

```
else if (($x_ratio * $height ) < $max_height) {
```

```
$tn_height = ceil($x_ratio * $height );
```

```
$tn_width = $max_width;
```

```
}
```

```
else {
```

```
$tn_width = ceil($y_ratio * $width);
```

```
$tn_height = $max_height;
```

```
}
```

- Αφού υπολογιστεί το επιθυμητό μέγεθος, η εικόνα αλλάζει μέγεθος ως εξής: Η αρχική εικόνα φορτώνεται στη μεταβλητή src. Για να φορτωθεί, χρησιμοποιείται είτε η συνάρτηση ImageCreatefromJpg() είτε η συνάρτηση ImageCreateFromBmp() αναλόγως τι επέκταση έχει το αρχείο μας:

```
if (preg_match('/\.(jpg)$/i', $filename[$i])) {
```

```
$src=ImageCreateFromJpeg("$userimages/$filename[$i]");
```

```
}
```

```
elseif (preg_match('/\.(bmp)$/i', $filename[$i])) {
```

```
$src=ImageCreateFromBmp("$userimages/$filename[$i]");
```

```
}
```

```
else
```

```
{ echo "Not a valid Image "; continue;}
```



- Στη συνέχεια δημιουργούμε μια νέα εικόνα με τη συνάρτηση ImageCreate()

**`$dst=ImageCreate($tn_width,$tn_height);`**

- Αντιγράφουμε στο \$dst την εικόνα \$src αφού την κλιμακώσουμε ως εξής:

**`ImageCopyResized($dst,$src,0,0,0,0,$tn_width,$tn_height,$width,$height);`**

- Αποθηκεύουμε την εικόνα με κατάληξη gif ως μικρογραφία στο thumbnails:

**`imagegif($dst, "$thumbnails/$filename[$i].gif");`**

- Για να μην έχουμε απώλεια μνήμης (memory leak) στο διακομιστή, πρέπει να καταστρέφουμε τις μεταβλητές των εικόνων ως εξής:

**`imagedestroy($src);`**

**`imagedestroy($dst);`**

- Επιβεβαιώστε την ορθή λειτουργία της φόρμας από το φυλλομετρητή σας. Ανεβάστε κάποιες φορές μια εικόνα και διαπιστώστε αν λειτουργεί σωστά το script σας.

==> Χρησιμοποιώντας κανονικές εκφράσεις και ελέγξτε ότι τα πεδία που υπέβαλλε ο χρήστης αποτελούνται μόνο από λατινικούς ή ελληνικούς χαρακτήρες, αριθμούς, στοιχεία στίξης.

Σε αυτό το σημείο έχουμε ολοκληρώσει τη φόρμα επεξεργασίας. Σε επόμενο εργαστήριο θα τοποθετήσουμε τα στοιχεία σε βάση δεδομένων.