

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή	3
1.1 Γενικά.....	3
1.2 Σκοπός της πτυχιακής	4
1.3 Υλοποίηση	6
1.4 Διάρθρωση της εργασίας	7
Κεφάλαιο 2 Η γλώσσα προγραμματισμού PHP	8
2.1 Γενικά.....	8
2.2 Ιστορικά στοιχεία.....	8
2.3 Τι μπορεί να κάνει;	11
2.4 Λόγοι χρησιμοποίησης της PHP στην εργασία μας.....	12
2.4.1 Απλότητα και αξιοπιστία	13
2.4.2 Συνεχής ενημέρωση και επεκτασιμότητα	13
2.4.3 Ελευθερία επιλογής λειτουργικού συστήματος και web server	13
2.4.4 Υψηλή δημοτικότητα.....	14
2.4.5 Συνδεσιμότητα με πολλά συστήματα βάσεων δεδομένων	15
2.5 PHP και MySQL.....	16
2.5.1 MySQL	16
2.5.2 Λόγος επιλογής της MySQL.....	16
2.5.3 Αλληλεπίδραση PHP και MySQL	18
2.5.4 Παράδειγμα χρήσης MySQL μέσω της PHP	19
Κεφάλαιο 3 Διαχείριση βιβλιογραφίας και δημοσιεύσεων	21
3.1 Γενική περιγραφή.....	21
3.2 Εργαλεία ανάπτυξης του συστήματος	22
3.3 Η βάση δεδομένων του συστήματος.....	23
3.4 Η ασφάλεια του συστήματος	25
3.4.1 Πρόσβαση στην ‘Ιδιωτική Περιοχή’ του συστήματος.....	25
3.4.2 Πρόσβαση στη βάση δεδομένων του συστήματος	27
3.5 Διαθέσιμες υπηρεσίες	28
3.5.1 Λειτουργίες διαχειριστή.....	28
3.5.1.1 Εισαγωγή νέας εργασίας.....	28
3.5.1.2 Τροποποίηση / Διαγραφή των εργασιών	30
3.5.2 Η λίστα των εργασιών	32
3.5.2.1 Αναζήτηση ανάμεσα στις εργασίες	32
3.5.2.2 Αλλαγή του τρόπου εμφάνισης των εργασιών	33
3.6 Φορητότητα συστήματος	34
Κεφάλαιο 4 Περιγραφή του File Manager	35
4.1 Σκοπός δημιουργίας	35
4.2 Η ασφάλεια του συστήματος	35
4.3 Παρεχόμενες υπηρεσίες.....	36
4.3.1 Σύντομη περιγραφή.....	36
4.3.2 Λειτουργίες πάνω στα αρχεία και τους καταλόγους.....	37
4.3.3 Επιλογές για την εμφάνιση των αρχείων	47
4.3.4 Επιπλέον δυνατότητες που παρέχει το σύστημα	50
4.4 Ιδιαιτερότητες υλοποίησης	52
4.5 Διάγραμμα λειτουργίας.....	56

4.6	Σχετικά υπάρχοντα συστήματα διαχείρισης αρχείων	59
4.6.1	Αναλυτική σύγκριση.....	59
4.6.2	Συγκριτικός πίνακας	68
4.6.3	Συνοψίζοντας.....	70
Κεφάλαιο 5 Σύνοψη και μελλοντικές επεκτάσεις		72
5.1	Σύνοψη.....	72
5.2	Μελλοντικές επεκτάσεις.....	73
Βιβλιογραφία		75
Παράρτημα Α Οδηγίες εγκατάστασης του συστήματος.....		77
A.1	Σύστημα διαχείρισης βιβλιογραφίας.....	77
A.2	Σύστημα διαχείρισης αρχείων.....	78
Παράρτημα Β Πηγαίος κώδικας		80
B.1	Τα αρχεία του συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων	80
B.1.1	showpub.inc.php.....	80
B.1.2	searchpubs.php	85
B.1.3	dbinsert.php	87
B.1.4	mod_del_list.php	91
B.1.5	modify.php	92
B.1.6	dbmodify.php	93
B.1.7	delete.php	98
B.2	Ο κώδικας του διαχειριστή αρχείων	100

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά

Τα τελευταία χρόνια, έχει παρατηρηθεί μια εντυπωσιακή αύξηση της χρήσης του διαδικτύου. Ολοένα και περισσότεροι άνθρωποι επισκέπτονται τις υπάρχουσες ιστοσελίδες, έχοντας ως βασικό τους σκοπό την ενημέρωση, την ψυχαγωγία, την επικοινωνία με τους συνανθρώπους τους, την συλλογή και την ανταλλαγή πληροφοριών. Κάποιοι άλλοι επιθυμούν τη δημιουργία των δικών τους ιστοσελίδων προκειμένου να εκπληρώσουν τους σκοπούς τους όπως το να προσφέρουν στους επισκέπτες της ιστοσελίδας τους τις πληροφορίες που επιθυμούν και να δεχθούν τα δικά τους δεδομένα. Για να φανεί αυτή η εντυπωσιακή στροφή του κόσμου προς την χρήση του διαδικτύου, ας σημειωθεί το αποτέλεσμα της έρευνας Netcraft[9] που δείχνει την αύξηση των IP διευθύνσεων στο Internet, από 21.600.000 τον Σεπτέμβριο του 2000 σε 71.723.098 το Σεπτέμβριο του 2005. Μάλιστα, από τον Αύγουστο μέχρι τον Σεπτέμβριο του 2005 είχαμε μια σημαντική αύξηση των 1.330.000 διευθύνσεων.

Υπάρχουν διάφοροι τρόποι πρόσβασης στο διαδίκτυο, όπως για παράδειγμα μέσω των ftp εφαρμογών. Τα τελευταία χρόνια όμως, ο βασικός τρόπος πρόσβασης στο διαδίκτυο είναι ο Παγκόσμιος Ιστός (World Wide Web-WWW). Πρόκειται για μια συλλογή από αρχεία κειμένου και πολυμέσων αλλά και διαφόρων υπηρεσιών δικτύου, που συνδέονται μεταξύ τους με τέτοιο τρόπο ώστε να δημιουργούν μια απέραντη ηλεκτρονική βιβλιοθήκη από την οποία μπορούμε εύκολα να ανακτήσουμε σημαντικές πληροφορίες.

Συμπεραίνουμε λοιπόν πως η πρόσβαση στο διαδίκτυο μέσω του Παγκόσμιου Ιστού είναι αρκετά σημαντική σε πολλούς ερευνητικούς τομείς, όπως για παράδειγμα σε κάποιο ακαδημαϊκό χώρο. Σε έναν τέτοιο χώρο είναι πιθανή η ύπαρξη ενός εργαστηρίου ή μιας ερευνητικής ομάδας που θα παράγει μια σειρά εργασιών, όπως συλλογές από βιβλία, άρθρα, τεχνικές αναφορές, δημοσιεύσεις σε συνέδρια, πτυχιακές εργασίες και άλλα. Είναι λοιπόν απαραίτητη η ύπαρξη ενός συστήματος που θα φροντίζει για τη διαχείριση των εργασιών της ομάδας.

Η ίδια ανάγκη υπήρχε και για την Ομάδα Παράλληλης Επεξεργασίας (PPG - Parallel Processing Group) του Πανεπιστημίου των Ιωαννίνων. Έτσι, οδηγηθήκαμε στη δημιουργία ενός συστήματος διαχείρισης βιβλιογραφικών δεδομένων και

δημοσιεύσεων. Το σύστημα που δημιουργήθηκε είναι γενικό. Μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση όπου απαιτείται η ύπαρξη ενός συστήματος για τη διαχείριση μιας ομάδας εργασιών. Η υλοποίηση του συστήματος μπορεί να έγινε για το PPG, αλλά αυτό δεν απαγορεύει την επέκτασή του για την επεξεργασία οποιασδήποτε άλλης ομάδας εργασιών. Ελάχιστη τροποποίηση κάποιων παραμέτρων του συστήματος αρκεί για να εφαρμοστεί αυτόσιο σε οποιοδήποτε άλλο περιβάλλον.

1.2 Σκοπός της πτυχιακής

Ο σκοπός της πτυχιακής εργασίας είναι η υλοποίηση ενός συστήματος για τη διαχείριση των βιβλιογραφικών δεδομένων μιας ερευνητικής ομάδας αλλά και των αρχείων που υπάρχουν στον κοινόχρηστο χώρο της ομάδας, μέσω του Παγκόσμιου Ιστού. Ως πρακτική εφαρμογή, το σύστημα που υλοποιήθηκε ενσωματώθηκε στις ιστοσελίδες του PPG.

Δημιουργήθηκε έτσι μία βιβλιογραφική βάση δεδομένων όπου θα αποθηκεύονται οι εργασίες της ομάδας, όπως για παράδειγμα οι εργασίες των προπτυχιακών και των μεταπτυχιακών φοιτητών ή των καθηγητών της ομάδας. Κάθε χρήστης του συστήματος θα μπορεί, μέσω της ιστοσελίδας του PPG, να βλέπει τις υπάρχουσες εργασίες και να αποθηκεύει στον υπολογιστή του αυτές που επιθυμεί. Επίσης, παρέχεται και η δυνατότητα προσθήκης νέων εργασιών ή τροποποίησης των ήδη υπάρχοντων. Έτσι εξασφαλίζεται η ανταλλαγή πληροφοριών μεταξύ των διαφόρων μελών της ομάδας. Κατά την προσθήκη μιας νέας εργασίας, είναι δυνατή και η φόρτωση (μέσω της διαδικασίας upload) της ηλεκτρονικής μορφής της. Για παράδειγμα, μπορεί να προσφέρεται η εργασία σε μορφή pdf ή ps. Έτσι, εκτός από τη θέαση των στοιχείων της εργασίας (όπως τίτλος, συγγραφέας κ.τ.λ), θα παρέχεται στο χρήστη και η δυνατότητα θέασης του περιεχομένου της και αποθήκευσης του στον υπολογιστή του. Τέλος, θεωρήθηκε απαραίτητη και η δυνατότητα απομακρυσμένης πρόσβασης στις εργασίες του τμήματος και η εκτέλεση διαφόρων λειτουργιών πάνω σε αυτές. Για το λόγο αυτό δημιουργήθηκε ένας διαχειριστής αρχείων που ενσωματώθηκε στο υπόλοιπο σύστημα. Έτσι, η εργασία μας διαιρέθηκε σε δύο τμήματα:

1) Διαχείριση βιβλιογραφίας και δημοσιεύσεων

Το σύστημα που δημιουργήθηκε παρέχει τις εξής λειτουργίες:

- a. Προβολή της λίστας των εργασιών (βιβλίων, άρθρων, πτυχιακών εργασιών, τεχνικών αναφορών, δημοσιεύσεων) που παράγει η ερευνητική ομάδα σε φθίνουσα χρονολογική σειρά.
- b. Αναζήτηση ανάμεσα στις εργασίες με βάση διάφορα κριτήρια, όπως τον τίτλο, τον συγγραφέα, το έτος δημοσίευσης, τον τύπο της εργασίας και άλλα.
- c. Εισαγωγή στην Ιδιωτική Περιοχή του συστήματος, με χρήση του κατάλληλου μυστικού κωδικού, και παροχή της δυνατότητας τροποποίησης, διαγραφής και προσθήκης νέων εργασιών.

Οι εργασίες της ομάδας αποθηκεύονται στη βάση δεδομένων του συστήματος. Για την επεξεργασία των δεδομένων της βάσης και την εκτέλεση των παραπάνω λειτουργιών χρησιμοποιήθηκε το Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL.

2) Διαχείριση αρχείων

Ο διαχειριστής αρχείων (file manager) που δημιουργήθηκε έχει ως σκοπό την παροχή απομακρυσμένης πρόσβασης στα ηλεκτρονικά αρχεία της ομάδας του PPG. Υπάρχουν διάφοροι τύποι αρχείων που μπορούν να αποτελέσουν την ομάδα των ηλεκτρονικών αρχείων του PPG, όπως για παράδειγμα pdf, ps, txt, zip, rhp, html. Βέβαια, θα πρέπει να σημειωθεί πως δεν υπάρχει κάποιος περιορισμός στον τύπο των αρχείων της ομάδας καθώς θα μπορούν να ανήκουν σε αυτή αρχεία σε οποιαδήποτε ηλεκτρονική μορφή, ακόμα και αρχεία ήχου ή εικόνας. Ο file manager τοποθετήθηκε στον κατάλληλο αρχικό (root) κατάλογο, επιτρέποντας με αυτό τον τρόπο την επεξεργασία των περιεχομένων αυτού του καταλόγου και όλων των εσωτερικών υποκαταλόγων του, μέσω του web. Οι δυνατότητες που παρέχει ο διαχειριστής στον χρήστη είναι οι εξής:

- a. Δημιουργία νέου αρχείου και καταλόγου
- b. Προσθήκη ενός αρχείου οποιουδήποτε τύπου μέσω της διαδικασίας upload
- c. Μετονομασία αρχείου και ταυτόχρονη αλλαγή του τύπου του
- d. Μετονομασία καταλόγου
- e. Τροποποίηση αρχείου
- f. Ατομική/Ομαδική διαγραφή αρχείων και καταλόγων
- g. Ατομική/Ομαδική αντιγραφή αρχείων και καταλόγων
- h. Ατομική/Ομαδική μετακίνηση αρχείων και καταλόγων
- i. Ταξινόμηση καταλόγων και αρχείων
- j. Εμφάνιση/απόκρυψη των λεπτομερειών των αρχείων

- k. Εμφάνιση των περιεχομένων του καταλόγου σε στήλες που επιλέγονται από τον χρήστη

Ας σημειωθεί ότι ο διαχειριστής αρχείων που δημιουργήθηκε αποτελεί ένα γενικό σύστημα διαχείρισης αρχείων που μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση όπου απαιτείται πρόσβαση στα περιεχόμενα ενός καταλόγου. Μπορεί να βρίσκει εφαρμογή στη διαχείριση των αρχείων του PPG, αυτό όμως δεν τον αποκλείει από την εφαρμογή του σε οποιοδήποτε άλλη ομάδα αρχείων, αλλάζοντας ελάχιστες ρυθμίσεις του κώδικα που τον υλοποιεί.

1.3 Υλοποίηση

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε και στα δύο τμήματα της εργασίας είναι η PHP[1,6]. Ο Web Server που χρησιμοποιήθηκε προκειμένου να εκτελεί τις λειτουργίες του συστήματός μας μέσω των PHP scripts είναι ο Apache[7] Web Server και το σύστημα διαχείρισης βάσης δεδομένων είναι η MySQL[2,3,8].

Ο χρήστης επισκεπτόμενος την ιστοσελίδα της ομάδας, επικοινωνεί με το σύστημα μέσω του browser του και έρχεται σε επαφή με τον Apache server του συστήματος. Έτσι, του δίνεται η δυνατότητα επιλογής της λειτουργίας που θέλει να εκτελέσει πάνω στις υπάρχουσες εργασίες, οπότε και συνδεόμαστε με τη βάση δεδομένων του συστήματος προκειμένου να εκτελεστεί το ερώτημα του χρήστη.

Αν ο χρήστης θελήσει να εισέλθει στον κοινόχρηστο χώρο, θα έρθει σε επαφή με τον file manager του συστήματος. Ο κοινόχρηστος χώρος είναι ένας κατάλογος που μπορεί να χρησιμοποιηθεί για να τοποθετεί εκεί η ερευνητική ομάδα κοινόχρηστα αρχεία όπως ερευνητικά άρθρα, βιβλία, τεχνικές αναφορές, δημοσιεύσεις σε συνέδρια και πολλούς άλλους τύπους αρχείων, ακόμα και τραγούδια ή αρχεία εικόνας. Στόχος είναι η διαχείριση των αρχείων του αρχικού (root) καταλόγου και όλων των υποκαταλόγων του. Απαραίτητη είναι η χρήση ενός μυστικού κωδικού ώστε να μην επιτρέπεται η είσοδος στα μη εξουσιοδοτημένα άτομα. Οποιοσδήποτε επισκέπτης της ιστοσελίδας του συστήματος θα μπορεί, αφού έχει γίνει δεκτός ο κωδικός του, να εισέλθει στον κοινόχρηστο αυτό χώρο, να δει τα περιεχόμενα του αρχικού καταλόγου, να εκτελέσει σε αυτά διάφορες λειτουργίες και να μεταφορτώσει (upload) τα δικά του αρχεία αν το επιθυμεί. Βέβαια, ο συγκεκριμένος web file manager μπορεί να γενικευτεί και να τοποθετηθεί σε οποιοδήποτε κατάλογο ενός συστήματος, τα περιεχόμενα του οποίου θέλουμε να διαχειριστούμε.

Για την υλοποίηση του συστήματος δημιουργήθηκαν γύρω στα 15 php αρχεία. Σε πολλά από αυτά, δεν χρησιμοποιήθηκαν μόνο συναρτήσεις της PHP αλλά και κλήσεις της MySQL, προκειμένου να επιτευχθεί η σύνδεση με τον server και τη βάση του συστήματος και η εκτέλεση ερωτημάτων σε αυτή.

Ο διαχειριστής αρχείων υλοποιήθηκε σε ένα μόνο αρχείο με 1800 γραμμές κώδικα. Όλες οι λειτουργίες που παρέχει στους χρήστες του υλοποιούνται μέσα σε αυτό το αρχείο. Κάθε φορά που επιλέγεται προς εκτέλεση κάποια λειτουργία, τρέχει το ίδιο αρχείο και καλούνται οι κατάλληλες συναρτήσεις, με τη βοήθεια των οποίων οδηγούμαστε στην εκτέλεση της επιλεγμένης λειτουργίας.

Το σύνολο των αρχείων που υλοποιήθηκε για την ανάπτυξη του συστήματος διαχείρισης βιβλιογραφίας και αρχείων, αποτελεί ένα πακέτο από php αρχεία που ξεπερνούν συνολικά τις 2800 γραμμές κώδικα.

1.4 Διάρθρωση της εργασίας

Η δομή της εργασίας έχει ως εξής: Αρχικά, στο κεφάλαιο 2, περιγράφεται η γλώσσα προγραμματισμού PHP. Γίνεται μια αναφορά στις εκδόσεις της γλώσσας, στις δυνατότητες της και στους λόγους επιλογής της για τη δημιουργία του συστήματος μας. Ακολουθεί μια αναφορά στην συνδεσιμότητα της PHP και της MySQL καθώς και ένα παράδειγμα χρήσης συναρτήσεων της PHP και κλήσεων της MySQL στον κώδικα της εργασίας μας.

Συνεχίζουμε στο κεφάλαιο 3 όπου γίνεται μια αναλυτική περιγραφή του συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων που δημιουργήθηκε. Περιγράφονται τα εργαλεία ανάπτυξης του συστήματος, η βάση δεδομένων του και οι ρυθμίσεις που έχουν γίνει προκειμένου να εξασφαλισθεί η ασφάλεια του. Καταλήγουμε με την περιγραφή των διαθέσιμων υπηρεσιών του συστήματος και τη δυνατότητα ενσωμάτωσης του σε οποιονδήποτε ιστότοπο βασίζεται στο συνδυασμό Apache, PHP και MySQL.

Ακολουθεί το κεφάλαιο 4 όπου γίνεται αναλυτική περιγραφή του διαχειριστή αρχείων που δημιουργήθηκε. Αναφέρεται ο σκοπός δημιουργίας του καθώς και οι ρυθμίσεις που έχουν γίνει προκειμένου να εξασφαλισθεί η ασφάλεια του. Συνεχίζουμε με την περιγραφή των παρεχόμενων υπηρεσιών και των δυνατοτήτων του συστήματος. Ακολουθεί η αναφορά στις ιδιαιτερότητες υλοποίησης του file manager ενώ παρουσιάζεται και ένα διάγραμμα λειτουργίας του. Τελειώνουμε με την περιγραφή κάποιων υπαρχόντων συστημάτων διαχείρισης αρχείων και την σύγκριση τους με το δικό μας σύστημα.

Τελειώνουμε με το κεφάλαιο 5 όπου γίνεται μια σύνοψη των όσων αναφέρθηκαν και περιγράφονται οι μελλοντικές προσθήκες που θα μπορούσαν να γίνουν στο σύστημα μας με στόχο την περαιτέρω ανάπτυξη του.

Ακολουθεί η βιβλιογραφία, το παράρτημα Α όπου δίνονται οδηγίες για την εγκατάσταση του συστήματος και το παράρτημα Β όπου παρουσιάζεται ο πηγαίος κώδικας του συστήματος.

Κεφάλαιο 2

Η γλώσσα προγραμματισμού PHP

2.1 Γενικά

Η PHP είναι μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού γενικού σκοπού, κατάλληλη για διαδικτυακές εφαρμογές και για την ανάπτυξη του Ιστού, η οποία μπορεί εύκολα να ενσωματωθεί σε HTML κώδικα. Πρόκειται για μια γλώσσα προγραμματισμού ανοικτού κώδικα (open source) που μπορεί κανείς εύκολα να την κατεβάσει ελεύθερα από το διαδίκτυο. Είναι εξειδικευμένη για χρήση σε εφαρμογές που αφορούν στον Παγκόσμιο Ιστό. Μπορεί να χρησιμοποιηθεί για τη δημιουργία δυναμικών ιστοσελίδων και την αλληλεπίδραση τους με βάσεις δεδομένων, για το ηλεκτρονικό εμπόριο (e-commerce) και για άλλες εφαρμογές στον Παγκόσμιο Ιστό όπως η δημιουργία ενός διαχειριστή αρχείων (file manager).

Παρά τη συχνή χρήση της για τη δημιουργία δυναμικών ιστοσελίδων, η PHP μπορεί να κάνει και πολλά άλλα πράγματα. Η σημαντική διαφορά της με άλλες γλώσσες προγραμματισμού είναι πως εκτελείται στο περιβάλλον του εξυπηρετητή (server-side) σε αντίθεση με άλλες γλώσσες (π.χ javascript) που εκτελούνται από τον browser του πελάτη. Έτσι, η PHP επικεντρώνεται κυρίως στο server-side scripting και κάνει οτιδήποτε ένα άλλο CGI πρόγραμμα μπορεί να κάνει, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο ιστοσελίδων, ή να στείλει και να πάρει cookies.

2.2 Ιστορικά στοιχεία

Η PHP (PHP : Hypertext Preprocessor) δημιουργήθηκε το 1995 και έχει παρουσιάσει αλματώδη εξέλιξη, κυρίως τα τελευταία επτά χρόνια. Τείνει πλέον να γίνει μια από τις πιο συχνά χρησιμοποιούμενες γλώσσες σε εφαρμογές που αφορούν στον παγκόσμιο Ιστό. Στη συνέχεια, παρατίθεται μια περιγραφή των διαφόρων εκδόσεων της γλώσσας, από τη δημιουργία της μέχρι και σήμερα.

▪ **PHP/FI [11]**

Δημιουργήθηκε από τον Rasmus Lerdorf το 1995, αρχικά ως ένα απλό σύνολο προγραμμάτων της γλώσσας Perl που το ονόμασε 'Personal Home Page Tools'. Σκοπός του ήταν η online πρόσβαση στα αρχεία του. Στη συνέχεια όμως, επειδή απαιτήθηκε μεγαλύτερη λειτουργικότητα, δημιούργησε μια πολύ μεγαλύτερη εφαρμογή σε γλώσσα C, η οποία προσέφερε τη δυνατότητα επικοινωνίας με βάσεις δεδομένων και επέτρεπε στους χρήστες να αναπτύξουν τις δικές τους απλές δυναμικές διαδικτυακές εφαρμογές. Αποφάσισε στη συνέχεια να δημοσιεύσει τον κώδικα έτσι ώστε να μπορεί ο καθένας να τον χρησιμοποιήσει.

Η έκδοση PHP/FI (που σημαίνει Personal Home Page / Forms Interpreter) περιελάμβανε μερικές από τις βασικές λειτουργίες της γλώσσας PHP όπως την ξέρουμε σήμερα. Περιελάμβανε Perl-like μεταβλητές και ενσωματωμένο HTML κώδικα. Η σύνταξη ήταν παρόμοια με αυτήν της Perl, αλλά πολύ πιο περιοριστική και απλή.

Το 1997, η δεύτερη έκδοση της προηγούμενης C εφαρμογής, η PHP/FI 2.0, υιοθετήθηκε από δεκάδες χιλιάδες χρηστών σε όλο τον κόσμο. Περίπου 50.000 διαδικτυακές περιοχές (domains) δήλωσαν ότι την εγκατέστησαν, που υπολογίζεται στο 1% των συνολικών περιοχών στο Internet.

▪ **PHP 3 [11]**

Η PHP 3 ήταν η πρώτη έκδοση που μοιάζει αρκετά με την PHP όπως την ξέρουμε σήμερα. Δημιουργήθηκε από τον Andi Gutmans και τον Zeev Suraski το 1997, σαν μια επέκταση της PHP/FI, αφού ανακάλυψαν πως η έκδοση αυτή αδυνατούσε να αναπτύξει μια eCommerce εφαρμογή που προσπαθούσαν να ολοκληρώσουν στα πλαίσια ενός πανεπιστημιακού project. Προσπαθώντας να συνεργαστούν και να αρχίσουν να δουλεύουν πάνω στον ήδη υπάρχοντα κώδικα της PHP/FI, ο Andi, ο Rasmus και ο Zeev αποφάσισαν να αναγγείλουν την PHP 3.0 ως επίσημο διάδοχο της PHP/FI 2.0 και σχεδόν σταμάτησαν την ανάπτυξη της PHP/FI 2.0.

Ένα από τα μεγαλύτερα πλεονεκτήματα της PHP 3.0 ήταν η δυνατότητα επέκτασής της η οποία προσέλκυσε δεκάδες προγραμματιστές να συνεργαστούν προκειμένου να δημιουργήσουν ακόμη περισσότερες web εφαρμογές βασισμένες στην PHP. Αναμφισβήτητα, αυτό ήταν το κλειδί στην τεράστια επιτυχία της PHP 3.0. Άλλα κύρια χαρακτηριστικά που εισήχθησαν στην PHP 3.0 ήταν η υποστήριξη της σύνταξης σε αντικειμενοστραφή προγραμματισμό και η ισχυρότερη και συνεπέστερη γλωσσική σύνταξη.

Ολόκληρη η νέα γλώσσα δημοσιεύτηκε με ένα νέο όνομα, που αφαίρεσε την έννοια της περιορισμένης προσωπικής χρήσης που περιελάμβανε η PHP/FI. Ονομάστηκε PHP (PHP : Hypertext Preprocessor).

Μέχρι το τέλος του 1998, δεκάδες χιλιάδες χρήστες είχαν εγκαταστήσει την PHP στα PC τους, ενώ εκατοντάδες χιλιάδες ιστοσελίδες δημιουργήθηκαν με χρήση αυτής της γλώσσας. Στο ζενίθ της, η PHP είχε εγκατασταθεί στο 10% περίπου από τους Web servers στο Internet.

Η PHP απέκτησε γρήγορα φανατικούς χρήστες και χρησιμοποιείται πλέον σε εφαρμογές στο διαδίκτυο από κάποιες από τις πιο προεξέχουσες οργανώσεις όπως οι Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson και NASA.

Τελικά, η PHP 3.0 δημοσιεύθηκε επίσημα το 1998, μετά από 9 μήνες δημόσιας δοκιμής.

▪ **PHP 4 [1,11]**

Το χειμώνα του 1998, αμέσως μόλις η PHP 3.0 παρουσιάστηκε επίσημα, ο Andi Gutmans και ο Zeev Suraski άρχισαν να επεξεργάζονται τον πυρήνα της PHP, στοχεύοντας σε επιπλέον βελτιώσεις της γλώσσας. Σκοπός τους ήταν η βελτίωση της απόδοσης των σύνθετων εφαρμογών τις οποίες η PHP 3.0 δεν μπορούσε να υποστηρίξει ικανοποιητικά. Η νέα μηχανή, που ονομάστηκε " μηχανή Zend" (Zend engine - αποτελούμενη από τα ονόματα του Zeev και του Andi), συνάντησε αυτούς τους στόχους επιτυχώς, και εισήχθη αρχικά στα μέσα του 1999. Η PHP 4.0, βασισμένη σε αυτήν την μηχανή, και συνδεμένη με ένα ευρύ φάσμα πρόσθετων νέων χαρακτηριστικών γνωρισμάτων, παρουσιάστηκε επίσημα το Μάιο του 2000, σχεδόν δύο έτη μετά από την προκάτοχό της PHP 3.0. Εκτός από την ιδιαίτερα βελτιωμένη απόδοση αυτής της έκδοσης, η PHP 4.0 περιελάμβανε και άλλα κύρια χαρακτηριστικά όπως: υποστήριξη για τους περισσότερους κεντρικούς υπολογιστές δικτύου (webservers), υποστήριξη του πρωτοκόλλου HTTP, πιο ασφαλείς τρόπους διαχείρισης των δεδομένων του χρήστη που εισήχθησαν (input) και διάφορα νέα γλωσσικά κατασκευάσματα. Σήμερα, η PHP 4.0 χρησιμοποιείται από εκατοντάδες χιλιάδες κατασκευαστές ιστοσελίδων, και σχεδόν πάνω από το 20% των περιοχών (domains) στο διαδίκτυο την έχουν εγκαταστήσει. Υπάρχει μια πολύ καλή ομάδα κατασκευαστών που εργάζονται με σκοπό τη διεύρυνση της γλώσσας και πολλές δεκάδες που εργάζονται σε συσχετιζόμενα με την PHP projects όπως το PEAR [12].

▪ PHP 5 [11]

Πρόκειται για την πιο πρόσφατη έκδοση της γλώσσας. Παρουσιάστηκε τον Ιούλιο του 2004 μετά από μακροχρόνια ανάπτυξη και διάφορες προδημοσιεύσεις. Οδηγείται κυρίως από τον πυρήνα της, τη Zend μηχανή 2.0 με ένα νέο πρότυπο αντικειμένου και δεκάδες άλλα νέα χαρακτηριστικά γνωρίσματα. Για παράδειγμα, νέες συναρτήσεις εισάγονται, υιοθετούνται νέες λέξεις- κλειδιά και άλλα νέα χαρακτηριστικά που προσδίδουν περισσότερες δυνατότητες. Η τελευταία έκδοση της PHP είναι η PHP 5.0.5 (Σεπτέμβριος 2005) ενώ υποψήφια προς έκδοση είναι και η PHP 5.1 (Release Candidate - Αύγουστος 2005).

2.3 Τι μπορεί να κάνει;

Η PHP χρησιμοποιείται συχνά για τη δημιουργία δυναμικών ιστοσελίδων. Δυναμική είναι μια ιστοσελίδα που αλληλεπιδρά με το χρήστη, έτσι ώστε ο κάθε χρήστης που την επισκέπτεται να βλέπει το περιεχόμενό της που μπορεί να αλλάζει συνεχώς, δεν πρόκειται δηλαδή για μια 'στάσιμη' ιστοσελίδα. Οι δυναμικές διαδικτυακές εφαρμογές παρουσιάζονται συνήθως σε σελίδες σχετικές με το ηλεκτρονικό εμπόριο, όπου το περιεχόμενο που επιδεικνύεται παράγεται από πληροφορίες που προσεγγίζονται σε μια βάση δεδομένων ή άλλη εξωτερική πηγή και που συνήθως εισάγονται από τους χρήστες της σελίδας. Έτσι, οι χρήστες μπορούν να εισάγουν δεδομένα στη βάση δεδομένων της ιστοσελίδας αλλά και να βλέπουν τα δεδομένα που υπάρχουν εκεί.

Παρόλο που χρησιμοποιείται πολύ συχνά για τη δημιουργία δυναμικών ιστοσελίδων, η PHP μπορεί να κάνει και πολλά άλλα πράγματα. Ακολουθεί μια αναφορά των κύριων τομέων στους οποίους μπορεί να χρησιμοποιηθεί ένα PHP script:

- **Server-side scripting**

Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Απαιτούνται τρία πράγματα. Ένας PHP μεταγλωττιστής (parser) (CGI ή server module), ένας web server (εξυπηρετητής σελίδων) και ένας web browser. Ο web server θα πρέπει να τρέχει με μια συνδεδεμένη εγκατάσταση της PHP. Τα αποτελέσματα του PHP προγράμματος μπορούμε να τα προσπελάσουμε με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.

- **Command line scripting**

Πρόκειται για την περίπτωση όπου μπορούμε να τρέχουμε ένα PHP script χωρίς server ή browser. Χρειαζόμαστε μόνο τον PHP μεταγλωττιστή. Αυτός ο τύπος είναι ιδανικός για scripts που εκτελούνται συχνά με τη χρήση της cron (σε *nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα scripts μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.

- **Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών)**

Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς client-side εφαρμογές, αλλά χρησιμοποιώντας κάποια προχωρημένα χαρακτηριστικά της, μπορεί κανείς να φτιάξει client-side εφαρμογές. Το PHP-GTK[13] προσφέρεται για αυτού του είδους τα προγράμματα και αποτελεί μια επέκταση της PHP και δίνει τη δυνατότητα δημιουργίας εφαρμογών συμβατών με κάθε είδους πλατφόρμα (cross-platform).

Με την PHP δεν είμαστε περιορισμένοι να εξάγουμε HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινιών Flash. Επιτρέπει την εξαγωγή οποιουδήποτε κειμένου όπως XHTML και οποιουδήποτε άλλου XML αρχείου. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα, καθώς και συνδεσιμότητα με τη γλώσσα προγραμματισμού Java. Επίσης, υποστηρίζει ανταλλαγή WDDX δεδομένων μεταξύ ουσιαστικά όλων των γλωσσών προγραμματισμού διαδικτύου.

Η PHP έχει εξαιρετικά χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, από την POSIX επέκταση ή τις Perl regular expressions μέχρι XML parsing αρχείων. Για τη μεταγλώττιση και την πρόσβαση αρχείων XML, υποστηρίζονται τα πρότυπα SAX και DOM.

2.4 Λόγοι χρησιμοποίησης της PHP στην εργασία μας

Επειδή η εφαρμογή μας ασχολείται με τη δημιουργία ενός διαχειριστικού συστήματος και την ενσωμάτωση του σε ιστοσελίδες, χρησιμοποιήθηκαν οι γλώσσες HTML και PHP. Ήταν απαραίτητος ένας Web Server που να εκτελεί τις λειτουργίες του

συστήματός μας μέσω των PHP scripts. Χρησιμοποιήθηκε ο Apache Web Server γιατί είναι συμβατός με τις περισσότερες πλατφόρμες. Τέλος, χρησιμοποιήθηκε το Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL Server για να είναι δυνατή η επεξεργασία των δεδομένων που είναι αποθηκευμένα στη βάση δεδομένων του συστήματος.

Στη συνέχεια, εξηγούνται αναλυτικά οι λόγοι χρήσης της PHP στην εργασία μας και ακολουθεί μια αναφορά στη MySQL και στην συνδεσιμότητα της με την PHP.

2.4.1 Απλότητα και αξιοπιστία

Η PHP προσφέρει μια απλή και καθολικά αποδεκτή λύση για τον εύκολο προγραμματισμό δυναμικών ιστοσελίδων. Οι προγραμματιστές μπορούν εύκολα να ενσωματώνουν PHP κώδικα μέσα στον HTML κώδικα. Η σύνταξη της είναι αντίστοιχη με αυτή της C και της Perl, γεγονός που καθιστά εύκολη τη χρήση της από καθέναν που έχει βασικές προγραμματιστικές γνώσεις.

Ακριβώς επειδή χρησιμοποιείται από ένα μεγάλο αριθμό χρηστών, η PHP υποστηρίζεται με τον καλύτερο δυνατό τρόπο. Επειδή πρόκειται για ένα προϊόν ανοικτού κώδικα (*open-source*), η PHP απολαμβάνει την υποστήριξη μιας μεγάλης ομάδας προγραμματιστών ανοικτού κώδικα. Παρέχεται έτσι άριστη τεχνική υποστήριξη στους χρήστες, και τα διάφορα σφάλματα εντοπίζονται και επισκευάζονται γρήγορα πράγμα που προσδίδει στα διάφορα scripts της γλώσσας ακόμη μεγαλύτερη αξιοπιστία.

2.4.2 Συνεχής ενημέρωση και επεκτασιμότητα

Η PHP ενημερώνεται και βελτιώνεται συνεχώς με σκοπό την επέκταση των δυνατοτήτων της. Νέα χαρακτηριστικά προστίθενται κατά τακτά χρονικά διαστήματα για να μπορεί να ανταπεξέλθει στις ανάγκες της αγοράς. Ο κομψός σχεδιασμός της και η απλότητα της σύνταξής της είναι αρκετά ώστε ένας PHP κώδικας να είναι σημαντικά ευκολότερος να διατηρηθεί και να ενημερωθεί από έναν συγκρίσιμο κώδικα της Perl ή της ASP. Επίσης, επειδή πρόκειται για ένα Open API προϊόν, ο κάθε χρήστης έχει τη δυνατότητα να προσθέσει easy-to-program modules για να επεκτείνει τη γλώσσα για δική του χρήση.

2.4.3 Ελευθερία επιλογής λειτουργικού συστήματος και web server

Η PHP είναι η πρώτη επιλογή των προγραμματιστών σε λειτουργικό Linux με Apache server, αλλά τρέχει εξίσου καλά και σε περιβάλλον Unix ή Windows με τον Netscape ή τον Microsoft Internet Explorer. Υποστηρίζει επίσης το πρωτόκολλο

HTTP και τα LDAP, SNMP, IMAP, COM. Μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένων και πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd και πολλούς άλλους web servers. Για την πλειοψηφία των servers η PHP έχει ένα module, για τους υπόλοιπους μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής.

Έτσι με την PHP ο χρήστης έχει την ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Η PHP είναι μια γλώσσα **cross-platform**, πράγμα που σημαίνει πως τα PHP προγράμματα μπορούν να λειτουργήσουν σε οποιονδήποτε από τους υπάρχοντες web servers. Αυτό αποτελεί και ένα βασικό πλεονέκτημα της PHP, συγκρινόμενη με άλλες γλώσσες προγραμματισμού όπως η ASP ή η ColdFusion και την καθιστά κατάλληλη για τα σημερινά ετερογενή περιβάλλοντα δικτύων.

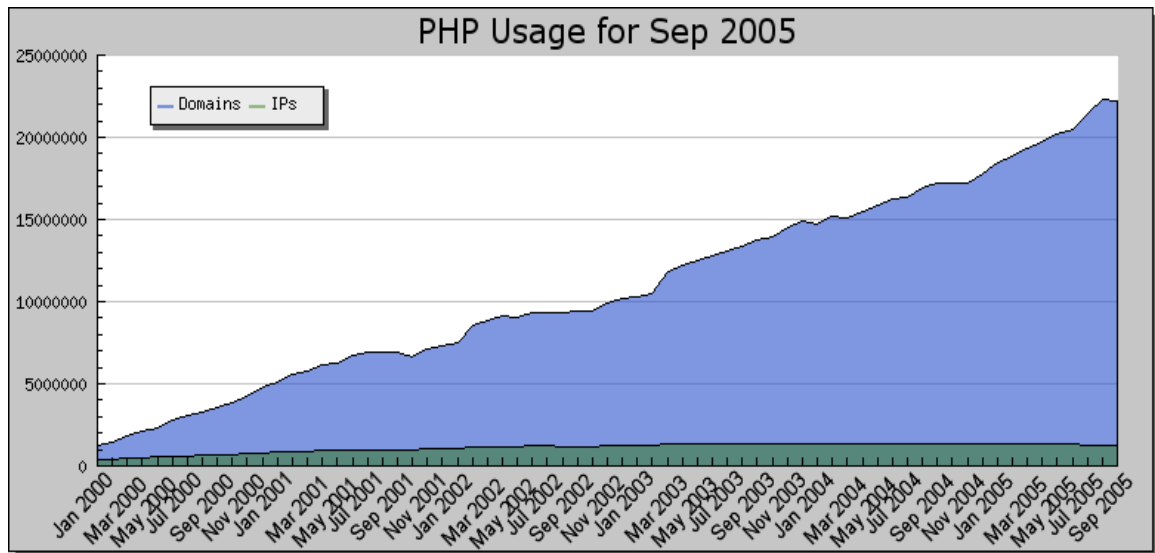
Επιπλέον, ο χρήστης έχει την ελευθερία χρησιμοποίησης συναρτησιακού (procedural) ή αντικειμενοστραφούς (object oriented) προγραμματισμού ή μιας ανάμειξης τους. Μάλιστα, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR [12]) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα

Τέλος, η PHP δεν εξαρτάται από τον browser που χρησιμοποιείται κάθε φορά (**browser-independent**) καθώς υποβάλλεται σε επεξεργασία στον server και τυπώνεται τελικά ως απλή HTML.

2.4.4 Υψηλή δημοτικότητα

Η PHP είναι σήμερα η πιο ταχεία αναπτυσσόμενη τεχνολογία για τη δημιουργία δυναμικών ιστοσελίδων. Σύμφωνα με την έρευνα Netcraft[10] σχετικά με το ποια τεχνολογία χρησιμοποιείται περισσότερο στον Ιστό, βρέθηκε πως η PHP χρησιμοποιείται σε περισσότερες από 22 εκατομμύριο περιοχές (domains), που φτάνει σε ένα ποσοστό 15% του συνόλου. Συγκεκριμένα, έχουν σημειωθεί γύρω στις 22.1670.75 περιοχές και 1.283.102 IP διευθύνσεις (βλ. Εικόνα 1). Χρησιμοποιείται στο πάνω από το 36% των Apache Web Servers, που είναι ο πιο συχνά χρησιμοποιούμενος server στον κόσμο. Αυτή η δημοτικότητα της PHP στις ιστοσελίδες με υψηλή επισκεψιμότητα, δείχνει την επιτυχία της στην κατασκευή διαδικτυακών εφαρμογών και ιστοσελίδων.

Ακολουθεί ένα διάγραμμα που επεξηγεί όσα αναφέρθηκαν παραπάνω:



Εικόνα 1: Χρήση της PHP από τον Ιανουάριο του 2000 μέχρι τον Σεπτέμβριο του 2005 (Netcraft survey [10])

Επίσης, θα πρέπει να αναφερθεί πως από το 1999 και μετά, ένας τεράστιος αριθμός βιβλίων έχουν γραφτεί από διαφορετικούς συγγραφείς για την PHP και δημοσιεύθηκαν από διαφορετικούς οίκους και σε πολλές διαφορετικές γλώσσες. Αυτός είναι και ένας ακόμη σημαντικός λόγος για την ευρεία αποδοχή της γλώσσας σε παγκόσμια κλίμακα.

2.4.5 Συνδεσιμότητα με πολλά συστήματα βάσεων δεδομένων

Η PHP προσφέρει άριστη συνδεσιμότητα με τις περισσότερες από τις κοινές βάσεις δεδομένων (συμπεριλαμβανομένων των Oracle, Sybase, MySQL και πολλών άλλων) και με διάφορες εξωτερικές βιβλιοθήκες, οι οποίες επιτρέπουν στον προγραμματιστή να κάνει τα πάντα, από την παραγωγή PDF εγγράφων μέχρι τη μεταγλώττιση XML κώδικα. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Οι εξής βάσεις δεδομένων υποστηρίζονται μέχρι στιγμής:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Υπάρχει επίσης μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που επιτρέπει στον χρήστη να χρησιμοποιεί οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) με το οποίο μπορούμε να συνδεθούμε σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο. Ειδική αναφορά στη συσχέτιση PHP και MySQL θα γίνει σε επόμενη παράγραφο.

Αυτοί είναι οι βασικοί λόγοι για τους οποίους η PHP είναι πλέον μια από τις γλώσσες που χρησιμοποιούνται πιο συχνά στον Παγκόσμιο Ιστό, για την κατασκευή ιστοσελίδων και τη δημιουργία web εφαρμογών και ο βασικός λόγος που τη χρησιμοποιήσαμε κι εμείς για την εργασία μας. Κι αυτό γιατί η εργασία είχε σαν αντικείμενο σε πρώτη φάση την κατασκευή ενός συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων που θα περιελάμβανε και μηχανή αναζήτησης και έπειτα την κατασκευή μιας εφαρμογής για τη διαχείριση των αρχείων της ομάδας του PPG μέσω των ιστοσελίδων, λειτουργίες για τις οποίες η PHP ήταν η κατάλληλη γλώσσα, όπως εξηγήθηκε παραπάνω.

2.5 PHP και MySQL

2.5.1 MySQL

Η MySQL [2,3,8] είναι ένα σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα που χρησιμοποιεί την SQL, τη δημοφιλέστερη γλώσσα για την προσθήκη, την επεξεργασία και την πρόσβαση στα στοιχεία μιας βάσης δεδομένων. Επειδή πρόκειται για σύστημα ανοικτού κώδικα (open source), καθένας μπορεί να εγκαταστήσει τη MySQL στον δικό του εξυπηρετητή και να την προσαρμόσει στις δικές του ανάγκες. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία και την ευελιξία της.

2.5.2 Λόγος επιλογής της MySQL

Ακολουθεί μία αναφορά στους λόγους επιλογής της MySQL ως το σύστημα διαχείρισης βάσεων δεδομένων της εργασίας μας.

- Η MySQL είναι πλέον η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα (open source) σε παγκόσμια κλίμακα, λόγω :

- της γρήγορης απόδοσης
 - της υψηλής αξιοπιστίας
 - και της ευκολίας χρήσης της ακόμη και για μη πεπειραμένους χρήστες.
- Χρησιμοποιείται σε περισσότερες από 6 εκατομμύρια εγκαταστάσεις, από τις μεγάλες εταιρίες ως τις πιο εξειδικευμένες ενσωματωμένες εφαρμογές σε κάθε ήπειρο στον κόσμο. Έρευνες αναφέρουν ότι χρησιμοποιείται ακόμα και σε εφαρμογές στην Ανταρκτική.
- Τρέχει σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων και των
 - Linux
 - Windows
 - OS/X
 - HP-UX
 - AIX
 - Netware
- Είναι ανοιχτού Κώδικα (Open Source) και έχει ελεύθερα δικαιώματα χρήσης (free license). Μπορεί να την κατεβάσει ελεύθερα ο οποιοσδήποτε από την ιστοσελίδα της MySQL.
- Είναι η βάση δεδομένων που επιλέγει μια καινούρια σειρά εφαρμογών οι οποίες στηρίζονται στο μοντέλο LAMP stack, που περιγράφεται ως εξής:
 - Λειτουργικό σύστημα :Linux
 - Web server :Apache
 - Σύστημα διαχείρισης βάσης δεδομένων :MySQL
 - Γλώσσα προγραμματισμού : PHP ή Perl ή Python

Δηλαδή η PHP, η MySQL και ο Apache αποτελούν μια ομάδα σε πολλές εφαρμογές. Μάλιστα αναφέρεται ότι στις περισσότερες web εφαρμογές που χρησιμοποιούν την PHP, το Σ.Δ.Β.Δ είναι η MySQL και ο server είναι ο Apache. Έτσι, με τη βοήθεια της PHP μπορούμε να κατασκευάσουμε ένα σύστημα διαχείρισης αρχείων, με την MySQL να αποθηκεύσουμε τα αρχεία αυτά σε μια βάση δεδομένων και στη συνέχεια, να επεξεργαζόμαστε τα αρχεία μέσω της ιστοσελίδας μας αποκτώντας πρόσβαση στη βάση του συστήματος. Το μοντέλο αυτό πλησιάζει αρκετά σε αυτό που διαπραγματεύεται η εργασία μας. Αυτός ήταν ένας από τους βασικούς λόγους που αναπτύξαμε κι εμείς το σύστημα μας με αυτά τα εργαλεία.

Ακολουθεί ένα άρθρο ενός ερευνητή σχετικά με την εξέλιξη των βάσεων ανοικτού κώδικα που συνοψίζει τα παραπάνω:

“ Η MySQL συνεχίζει να χρησιμοποιείται ευρύτατα αλλά στερείται μερικά κύρια χαρακτηριστικά.

Συνεχίζει να έχει ένα πολύ υψηλό ποσοστό υιοθέτησης από διάφορες βιομηχανίες και είναι γνωστή για την αξιοπιστία της, την ευκολία χρήσης και την απόδοση της. Πρόκειται για ένα πλήρες, σχεσιακό DBMS που υποστηρίζει την ανίχνευση αδιεξόδων. Εντούτοις, στερείται μερικά βασικά χαρακτηριστικά γνωρίσματα των DBMS, όπως τα triggers, stored procedures και views τα οποία είναι πιθανό να είναι διαθέσιμα στην έκδοση 5.0 της MySQL. Αυτά τα χαρακτηριστικά γνωρίσματα είναι πιθανό να ωθήσουν τη MySQL σε ακόμη υψηλότερα ποσοστά υιοθέτησης από εταιρίες. Αυτή τη στιγμή, υπάρχουν πολλές μεγάλες επιχειρήσεις που τρέχουν τη MySQL για να υποστηρίξουν mission-critical αλλά και non-mission-critical εφαρμογές.”

Έρευνα Forrester : “ Open Source Databases Come of Age”

Από τον Noel Yuhanna. [14]

2.5.3 Αλληλεπίδραση PHP και MySQL

Για πολλούς ανθρώπους, ο βασικός λόγος για την εκμάθηση μιας γλώσσας προγραμματισμού όπως η PHP είναι η αλληλεπίδραση που προσφέρει με πολλές βάσεις δεδομένων. Είναι πραγματικά εκπληκτικό πόσο χρήσιμη μπορεί να είναι μια βάση δεδομένων σε μια ιστοσελίδα. Υπάρχει μια τεράστια ποικιλία πραγμάτων που μπορεί να κάνει κάποιος με τη χρήση της PHP και της MySQL, από την επίδειξη απλών λιστών δεδομένων μέχρι το τρέξιμο ολόκληρης ιστοσελίδας από μια βάση δεδομένων. Κάποια παραδείγματα αλληλεπίδρασης PHP και MySQL είναι τα εξής:

1) Δημιουργία δυναμικών ιστοσελίδων που στηρίζονται σε βάσεις δεδομένων:

Χαρακτηριστικό παράδειγμα αποτελούν οι ιστοσελίδες που παίρνουν όλες τις πληροφορίες τους από μια βάση δεδομένων. Το μόνο που χρειάζεται είναι η χρήση των κατάλληλων συναρτήσεων της MySQL ώστε να επιτευχθεί η πρόσβαση στον server και μετά στη βάση δεδομένων για την κατάλληλη επεξεργασία των περιεχομένων της. Σε αυτή την κατηγορία ανήκει και το πρώτο τμήμα της εργασίας μας όπου σκοπός είναι η πρόσβαση (μέσω του συστήματος μας) στη βάση με τις διάφορες εργασίες, η τροποποίησή της, η εκτέλεση ερωτημάτων σε αυτή, η ανάκτηση των δεδομένων της και η εμφάνισή τους στην ιστοσελίδα μας.

- 2) **Αλλαγή μορφής ιστοσελίδας:** Εάν κάποιος θελήσει να αλλάξει τη μορφή της ιστοσελίδας του, είναι πολύ πιθανό πως θα σπαταλήσει έναν αρκετά πολύτιμο χρόνο για να αλλάξει όλες τις σελίδες του website του. Με την PHP και τη MySQL, όλο το website θα μπορούσε να ήταν ένα ή 2 script αρχεία. Αυτά θα αποκτούσαν πρόσβαση σε μια βάση δεδομένων MySQL για να πάρουν πληροφορίες για τις σελίδες. Για να ενημερωθεί η μορφή του website αρκεί να αλλαχτεί μία μόνο σελίδα.
- 3) **Forums και mailing lists:** Εκατοντάδες τέτοιοι πίνακες μηνυμάτων [15] στο διαδίκτυο οργανώνονται χρησιμοποιώντας PHP και MySQL. Αυτοί προσφέρουν μια ευρεία ποικιλία επιλογών και είναι πολύ πιο επαρκείς από άλλα συστήματα που δημιουργούν μια σελίδα για κάθε μήνυμα. Όλες οι σελίδες στο φόρουμ μπορούν να ενημερωθούν με την αλλαγή ενός και μόνο script. Επίσης πολλές mailing lists δημιουργούνται με τη χρήση της PHP, δίνοντας τη δυνατότητα στους χρήστες να διαχειρίζονται τα mail τους, να τα διαγράφουν, να στέλνουν νέα μηνύματα κ.ο.κ.

2.5.4 Παράδειγμα χρήσης MySQL μέσω της PHP

Στη συνέχεια, θα αναφέρουμε κάποια παραδείγματα χρήσης συναρτήσεων της PHP και κλήσεων της MySQL στον κώδικα της εργασίας μας, για να φανεί καλύτερα η συνδεσιμότητα της PHP με την MySQL, όπως αυτή αναφέρθηκε προηγουμένως και η χρήση της MySQL για την επεξεργασία των περιεχομένων της βάσης του συστήματος.

- ```
/* Σύνδεση με τον server του συστήματος */
$mid = @mysql_connect("agatha.cs.uoi.gr", $username, $password)
 or die("Unable to connect to database server!");
```
- ```
/* Σύνδεση με τη βάση του συστήματος */
@mysql_select_db($database,$mid)
or die( "Unable to select required database!");
```
- ```
/* Εκτέλεση ερωτήματος στη βάση του συστήματος */
$query="DELETE FROM $tablename WHERE id='$id'";
mysql_query($query) or die("Could not remove the publication!\n");
```
- ```
/* Επιστροφή του αριθμού των εγγράφων του αποτελέσματος ενός ερωτήματος*/
$rows=mysql_numrows($query);
```

- /* Επιστροφή μιας εγγραφής από τον πίνακα της βάσης */

```
$query = "SELECT * FROM $tablename WHERE id='$id'";  
$result = mysql_query($query);  
if (!$result) die("Cannot retrieve the publication to be  
modified.\n");  
$row = mysql_fetch_array($result);
```
- /* Κλείσιμο της σύνδεσης με τον server του συστήματος */

```
mysql_close();
```

Κεφάλαιο 3

Διαχείριση βιβλιογραφίας και δημοσιεύσεων

3.1 Γενική περιγραφή

Σκοπός αυτού του τμήματος της πτυχιακής εργασίας είναι η δημιουργία ενός συστήματος για τη διαχείριση βιβλιογραφικών δεδομένων. Οι δυνατότητες που απαιτούνται είναι οι εξής:

- Προβολή λίστας με τις υπάρχουσες εργασίες της ερευνητικής ομάδας.
- Εισαγωγή νέων εργασιών.
- Προσθήκη των εργασιών και σε ηλεκτρονική μορφή (διαδικασία upload).
- Τροποποίηση των υπαρχόντων εργασιών.
- Διαγραφή των υπαρχόντων εργασιών.

Το σύστημα που δημιουργήθηκε είναι **γενικό**. Μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση όπου απαιτείται η ύπαρξη ενός συστήματος για τη διαχείριση μιας ομάδας εργασιών. Η υλοποίηση του συστήματος όμως έγινε για την Ομάδα Παράλληλης Επεξεργασίας του Πανεπιστημίου των Ιωαννίνων, χωρίς αυτό να απαγορεύει την επέκτασή του για την επεξεργασία οποιασδήποτε άλλης ομάδας εργασιών. Ελάχιστη τροποποίηση κάποιων παραμέτρων του συστήματος αρκεί για να εφαρμοστεί αυτόσιο σε οποιοδήποτε άλλο περιβάλλον.

Το σύστημα μας παράγει ιστοσελίδες όπου είναι δυνατή η προβολή μιας λίστας με τις εργασίες που υπάρχουν αποθηκευμένες στο σύστημα. Για κάθε εργασία, εμφανίζονται κάποιες χρήσιμες πληροφορίες για αυτή όπως, ο τίτλος της, ο συγγραφέας, ο τύπος της, η ημερομηνία έκδοσής της, το συνέδριο που δημοσιεύθηκε κ.τ.λ. Επίσης, είναι δυνατή και η πρόσβαση στο κείμενο της εργασίας σε ηλεκτρονική μορφή. Έτσι, κάθε επισκέπτης της ιστοσελίδας μπορεί να δει τη λίστα των εργασιών και να σώσει στον υπολογιστή του αυτές που επιθυμεί. Επίσης, μπορεί αν θέλει να δει μόνο ένα συγκεκριμένο τύπο εργασιών ή και να αναζητήσει κάποια λέξη-κλειδί μέσα στη λίστα των εργασιών. Ακόμη, εισάγοντας τον κατάλληλο κωδικό και αφού αυτός γίνει δεκτός από το σύστημα, μπορεί να αποκτήσει πρόσβαση στις υπόλοιπες διαθέσιμες υπηρεσίες που είναι : Η εισαγωγή νέων εργασιών και η τροποποίηση / διαγραφή υπαρχόντων εργασιών. Υπάρχουν επτά τύποι εργασιών. Σε περίπτωση επιλογής εισαγωγής νέας εργασίας, καλείται να επιλέξει τον τύπο της εργασίας που

θέλει να εισάγει, να δώσει τις κατάλληλες πληροφορίες για αυτή (όπως τίτλο κ.τ.λ) και να την προσφέρει σε ηλεκτρονική μορφή (αν αυτή είναι διαθέσιμη). Επιπλέον υπάρχει και η δυνατότητα προσθήκης μέχρι δύο συνδέσμων (links) που θα αναφέρονται σε αυτήν την εργασία. Διαφορετικά, σε περίπτωση επιλογής τροποποίησης της εργασίας, ο χρήστης αλλάζει τα στοιχεία της εργασίας που αυτός επιθυμεί και αποθηκεύει τις αλλαγές. Τέλος σε περίπτωση επιλογής διαγραφής, ο χρήστης επιλέγει την εργασία που θέλει να διαγράψει και καλείται να επιβεβαιώσει τη διαγραφή της.

3.2 Εργαλεία ανάπτυξης του συστήματος

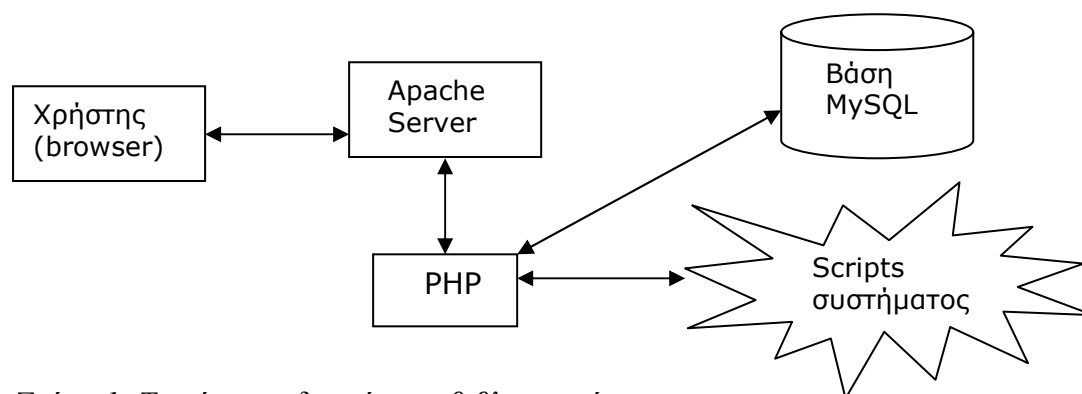
Επειδή το σύστημα μας παράγει ιστοσελίδες, χρησιμοποιήθηκαν οι γλώσσες HTML και PHP. Το Σύστημα Διαχείρισης Βάσης Δεδομένων που χρησιμοποιήθηκε είναι ο MySQL Server. Οι λόγοι επιλογής της PHP και της MySQL έχουν ήδη αναφερθεί στο κεφάλαιο 2. Ήταν απαραίτητος ένας Web Server που να εκτελεί τις λειτουργίες του συστήματος μας μέσω των PHP scripts. Επιλέχθηκε ο APACHE Web Server. Οι λόγοι επιλογής του περιγράφονται παρακάτω:

- Είναι συμβατός με τις περισσότερες πλατφόρμες, καθώς τρέχει το ίδιο καλά τόσο σε Unix όσο και σε Windows, αλλά και στα περισσότερα λειτουργικά συστήματα.
- Οι περισσότερες ιστοσελίδες που δημιουργούνται με την PHP λειτουργούν έχοντας ως σύστημα διαχείρισης δεδομένων την MySQL, ως γλώσσα προγραμματισμού την PHP και ως διαχειριστή τους τον Apache.
- Είναι ο δημοφιλέστερος web server. Δημιουργήθηκε τον Δεκέμβριο του 1995 και ένα χρόνο αργότερα είχε ήδη καθιερωθεί ως ο νούμερο ένα διαχειριστής στο διαδίκτυο, θέση την οποία κατέχει μέχρι σήμερα σύμφωνα με την έρευνα Netcraft[16]. Η έρευνα αυτή εκτελείται σε τακτά χρονικά διαστήματα από το 1996 μέχρι σήμερα και δείχνει πάντα την υψηλή δημοτικότητα του Apache. Η πιο πρόσφατη δημοσιεύθηκε τον Φεβρουάριο του 2005 και φανερώνει πως το 69% των ιστοσελίδων στο Internet χρησιμοποιούν τον Apache, που μεταφράζεται σε 49598424 domains.

Ακολουθεί ένα διάγραμμα (βλ. Σχήμα 1) όπου περιγράφονται τα βασικά σημεία του συστήματος βιβλιογραφίας που υλοποιήθηκε. Ο χρήστης επικοινωνεί με το σύστημα μέσω του browser του και έρχεται σε επαφή με τον Apache server του συστήματος. Έπειτα, μέσω των PHP scripts του συστήματος, γίνεται η σύνδεση με τη βάση δεδομένων της MySQL προκειμένου να εκτελεστούν τα ερωτήματα του χρήστη.

Αναλυτικότερα, η διαδικασία που ακολουθείται προκειμένου να εκτελεστεί μια λειτουργία που επιθυμεί ο χρήστης είναι η εξής:

- Αρχικά ο χρήστης μέσω του browser του συνδέεται με τον server του συστήματος και αποκτά τη δυνατότητα θέασης του συστήματος και των δυνατοτήτων του.
- Έπειτα, επιλέγει τη λειτουργία που επιθυμεί να εκτελέσει.
- Τα scripts του συστήματος (μέσω της PHP), προσπαθούν να συνδεθούν με τον Apache server για να μπορέσουν έπειτα να συνδεθούν και με τη βάση του συστήματος.
- Απομένει η εκτέλεση του ερωτήματος πάνω στη βάση και το κλείσιμο της σύνδεσης με τον Apache server.
- Ο χρήστης μπορεί να επιλέξει την επόμενη λειτουργία που θέλει να εκτελέσει.



Σχήμα 1: Το σύστημα διαχείρισης βιβλιογραφίας

3.3 Η βάση δεδομένων του συστήματος

Θα πρέπει να αναφερθεί ότι η βάση δεδομένων του συστήματος ονομάζεται *parapubsubdb* και αποτελείται από ένα μόνο πίνακα τον *publications* όπου και αποθηκεύονται οι εργασίες του συστήματος. Ακολουθεί ένα παράδειγμα του ερωτήματος της MySQL που χρησιμοποιήσαμε για τη δημιουργία του πίνακα, όπου φαίνονται και τα διάφορα πεδία του:

```
mysql_query('CREATE TABLE publication (id int(6) NOT NULL  
auto_increment,Bibkey varchar(30),Type varchar(30),Title  
varchar(128),Authors varchar(100),Journal varchar(100),Volume  
varchar(100),Number varchar(100),Conference varchar(100),Address  
varchar(100),Booktitle varchar(100),Editors varchar(100),Publisher  
varchar(100),Month varchar(100),Year varchar(100),Pages varchar  
(25),Info varchar(200),Keywords varchar(250),File1 varchar(100),File2  
varchar(100),File3 varchar(100),File4 varchar(100),File5
```

```
varchar(100),Type1 varchar(100),Type2 varchar(100),Type3
varchar(100),Type4 varchar(100),Type5 varchar(100),PRIMARY
KEY(id),UNIQUE id (id))').
```

Αναλυτικά τα 28 πεδία του πίνακα φαίνονται στο παρακάτω σχήμα:

Id	Bibkey	Type	Title	Authors	Journal
Volume	Number	Conference	Address	Booktitle	Editors
Publisher	Month	Year	Pages	Info	Keywords
File1	File2	File3	File4	File5	Type1
Type2	Type3	Type4	Type5		

Σχήμα 2: Τα πεδία του πίνακα του συστήματος

Το πρωτεύον κλειδί του πίνακα είναι το id. Έχουμε αυτόματη αρίθμηση, οπότε η τιμή του ξεκινά από το 1 και κάθε φορά που εισάγεται μια καινούρια εργασία, αυξάνεται κατά ένα.

Κάθε φορά που εισάγεται, διαγράφεται ή τροποποιείται μια εργασία, ενημερώνονται τα κατάλληλα πεδία του πίνακα publications, ανάλογα με τον τύπο της εργασίας. Η διαδικασία που ακολουθείται είναι η εξής: Αρχικά, μέσω των κατάλληλων εντολών της MySQL, συνδεόμαστε με τον web server κι έπειτα με τη βάση δεδομένων του συστήματος. Το μόνο που μένει είναι η εκτέλεση του ερωτήματος στον πίνακα της βάσης.

Υπάρχουν επτά τύποι εργασιών. Τα πεδία για κάθε τύπο επιλέχθηκαν έτσι ώστε να έχουμε μια ικανοποιητική περιγραφή αλλά και μια πλήρη εικόνα των εργασιών του κάθε τύπου. Προσπαθήσαμε να χρησιμοποιήσουμε όσα πεδία θα ήταν αρκετά για μια σωστή περιγραφή των εργασιών κάθε τύπου. Γι' αυτό και θα παρατηρήσουμε κάποια κοινά αλλά και κάποια διαφορετικά πεδία μεταξύ των επτά τύπων. Ακολουθεί μια αναλυτική περιγραφή των πεδίων των τύπων που υποστηρίζει το σύστημα:

- **Journal article**
id-Bibkey-Title-Authors-Journal-Volume-Number-Month-Year-Pages-Keywords
- **Conference article:**
id- Bibkey-Title-Authors-Conference-Address-Month-Year-Pages-Keywords
- **Technical report**
id-Bibkey-Title-Authors-Number-Address-Month-Year-Info-Keywords
- **Book chapter**
id-Bibkey-Title-Authors-Booktitle-Editors-Publisher-Year-Pages-Keywords

- **Book**
id-Bibkey-Authors-Booktitle-Publisher-Year-Keywords
- **Thesis**
id-Bibkey-Authors-Number-Booktitle-Address-Month-Year-Info-Keywords
- **Other**
id-Bibkey-Title-Authors-Info-Keywords

3.4 Η ασφάλεια του συστήματος

3.4.1 Πρόσβαση στην ‘Ιδιωτική Περιοχή’ του συστήματος

Πολλή σημαντική είναι η ασφάλεια που θα πρέπει να προσφέρει το σύστημα μας καθώς και η εξασφάλιση της αποφυγής της πρόσβασης σε αυτό από μη εξουσιοδοτημένα άτομα. Για την επίτευξη αυτού του σκοπού χρησιμοποιήθηκε η ιδιότητα: **Apache Basic Authentication**[17] που παρέχει ο Apache server του συστήματος μας.

Στην ‘Ιδιωτική περιοχή’ του συστήματος υπάρχουν οι διαθέσιμες υπηρεσίες: εισαγωγή, τροποποίηση, διαγραφή των εργασιών, που αποτελούν και τις λειτουργίες του χρήστη-διαχειριστή. Για τη χρήση τους απαιτείται η εισαγωγή ενός κατάλληλου μυστικού κωδικού. Μόνο ο χρήστης που θα δώσει το σωστό username και password θα μπορέσει να χρησιμοποιήσει αυτές τις υπηρεσίες. Σε οποιονδήποτε άλλο απαγορεύεται η χρήση τους, οπότε και θα αρκεστεί στην εμφάνιση της λίστας των τρεχόντων εργασιών, χωρίς να μπορεί να τις τροποποιήσει.

Ουσιαστικά, θα πρέπει να καθοριστούν τα έγκυρα ζεύγη από usernames και passwords για την εισαγωγή στην ‘Ιδιωτική Περιοχή’ του συστήματος. Τα βήματα για να το επιτύχουμε αυτό είναι τα εξής:

- 1) Δημιουργία ενός password file και
- 2) Καθορισμός του περιεχομένου του

Στον server του συστήματος δημιουργούμε ένα αρχείο που περιέχει τα αποδεκτά usernames και passwords. Επειδή το αρχείο αυτό περιέχει πολύ σημαντικές πληροφορίες, θα πρέπει να τοποθετηθεί έξω από τον κατάλογο ‘Protected’, του οποίου τα αρχεία θέλουμε να μην είναι ορατά από τους χρήστες που δεν δίνουν έγκυρο κωδικό. Τα passwords σε αυτό το αρχείο δεν θα είναι ορατά καθώς θα είναι σε encrypted μορφή. Για να δημιουργηθεί το αρχείο αυτό θα πρέπει να χρησιμοποιήσουμε την εντολή htpasswd του Apache. Η σύνταξή της εντολής για τη δημιουργία ενός νέου username και password είναι η εξής:

```
a. #htpasswd -c /usr/apache/passwd/passwd user1
```

Έπειτα θα μας ζητηθεί να δώσουμε το password για το username που μόλις πληκτρολογήσαμε. Για παράδειγμα

```
b. #htpasswd -c /usr/apache/passwd/passwd user1
New password: mypassword
Re-type new password: mypassword
Adding password for user user1
```

Έτσι, στον κατάλογο passwd του Apache δημιουργείται το password file με το όνομα passwd που θα περιέχει ένα χρήστη με το όνομα user1. Έπειτα αλλάζουμε τα δικαιώματά του αρχείου ώστε μόνο ο web server να μπορεί να το διαβάσει και μόνο ο root να γράφει σε αυτό:

```
c. chown root.nogroup /usr/apache/passwd/passwd
chmod 640 /usr/apache/passwd/passwd
```

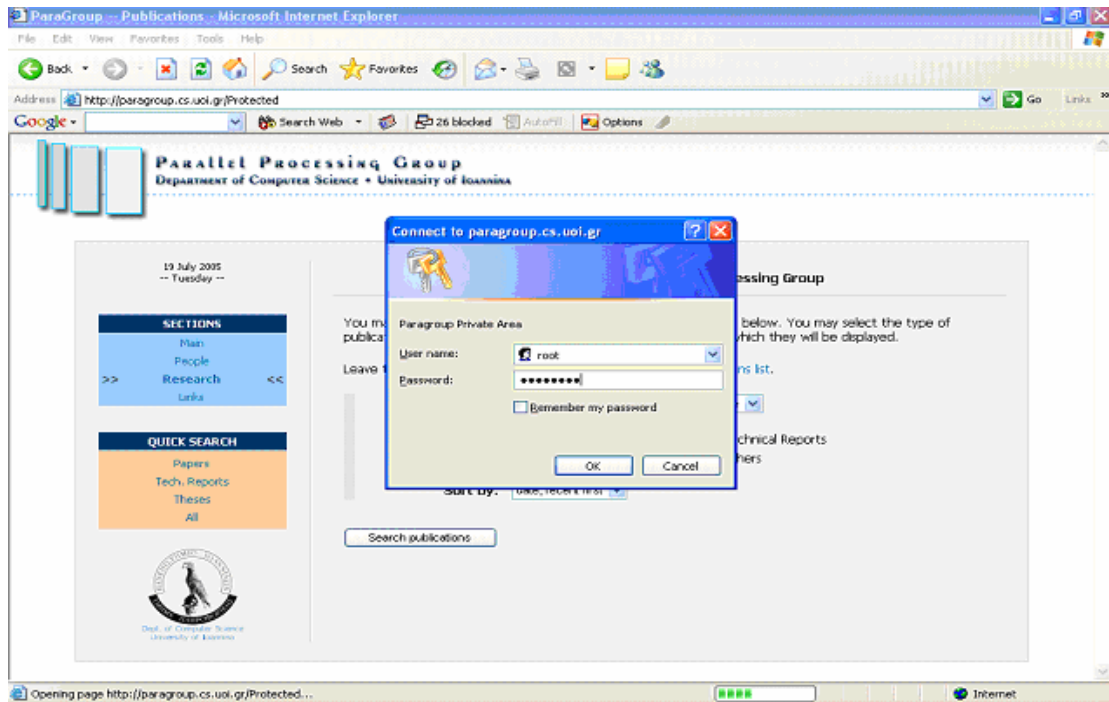
Αν στη συνέχεια θελήσουμε να προσθέσουμε και έναν νέο χρήστη, αρκεί να εκτελέσουμε την ίδια εντολή αλλά χωρίς την `-c` παράμετρο γιατί αυτή χρησιμοποιείται μόνο στην περίπτωση που το αρχείο δημιουργείται για πρώτη φορά.

Έτσι, όταν ο χρήστης προσπαθήσει να εισέλθει στην Ιδιωτική Περιοχή, θα εμφανιστεί ένα παράθυρο για την είσοδο του κωδικού του. Το μήνυμα που εμφανίζεται στο παράθυρο καθορίζεται στο configuration file του server. Εκεί έχουν προστεθεί και τα αποδεκτά usernames. Ένα παράδειγμα του κώδικα που προστίθεται στο configuration file είναι το εξής:

```
d. AuthType Basic
AuthName "Paragroup Private Area"
AuthUserFile /usr/apache/passwd/passwd
Require user user1
```

Ο ίδιος κώδικας τοποθετείται και στο **.htaccess** αρχείο που δημιουργούμε. Αυτό το αρχείο τοποθετείται στον κατάλογο του οποίου τα δεδομένα θέλουμε να προστατεύσουμε. Έτσι, σύμφωνα με τον κώδικα αυτό, το μήνυμα που θα εμφανίζεται στο παράθυρο είναι το "Paragroup Private Area" (βλ. Εικόνα 2). Ο κωδικός που δίνει ο χρήστης εξετάζεται αν ανήκει στα αποδεκτά ζεύγη username και password. Αν ναι, τότε θα μπορέσει να συνεχίσει. Διαφορετικά, έχει τη δυνατότητα να δώσει τον κωδικό του ακόμη δύο φορές και αν και πάλι δεν γίνει δεκτός, θα του απαγορευθεί η είσοδος και θα εμφανιστεί κατάλληλο μήνυμα.

Στη συνέχεια παρατίθεται μια εικόνα που δείχνει την προσπάθεια εισόδου στην Ιδιωτική Περιοχή ενός επισκέπτη της ιστοσελίδας.



Εικόνα 2: Προσπάθεια εισόδου στην 'Ιδιωτική Περιοχή' του συστήματος

3.4.2 Πρόσβαση στη βάση δεδομένων του συστήματος

Είναι πολύ σημαντικό να εξασφαλίσουμε πως μόνο τα εξουσιοδοτημένα άτομα θα μπορούν να έχουν πρόσβαση στη βάση του συστήματος. Κάθε φορά που θέλουμε να εκτελέσουμε ένα ερώτημα στον πίνακα της βάσης, θα πρέπει να ακολουθήσουμε τα παρακάτω βήματα:

- 1) Σύνδεση με τον server του συστήματος.
- 2) Σύνδεση με την βάση του συστήματος.
- 3) Εκτέλεση του ερωτήματος στον πίνακα της βάσης.

Ακολουθεί ένα παράδειγμα τμήματος κώδικα ενός PHP script που χρησιμοποιείται για τη σύνδεση με τη βάση του συστήματος.

```

/* Σύνδεση με τη βάση δεδομένων του συστήματος */
$mid = mysql_connect("agatha.cs.uoi.gr", $username, $password)
    or die("Unable to connect to database server!");
@mysql_select_db($database, $mid)
    or die( "Unable to select required database!");

```

Για να συνδεθούμε με τον server, απαιτείται η γνώση ενός κατάλληλου μυστικού κωδικού που κρατείται σε ένα αρχείο του συστήματος και δεν είναι ορατό μέσω της ιστοσελίδας. Η προεπιλεγμένη επιλογή του συστήματος είναι η εξής: username=root, password=" ". Θεωρήσαμε σκόπιμη την αλλαγή αυτών των τιμών ούτως ώστε να μην μπορεί κανείς να συνδεθεί στη βάση εάν δεν ξέρει τον σωστό κωδικό. Έτσι, με

χρήση των κατάλληλων εντολών του περιβάλλοντος της MySQL, οι τιμές αυτές διαφοροποιήθηκαν από την αρχική τους τιμή, εξασφαλίζοντας την πρόσβαση στη βάση μόνο των ατόμων που έχουν γνώση του σωστού κωδικού.

3.5 Διαθέσιμες υπηρεσίες

Στη συνέχεια περιγράφονται οι υπηρεσίες που προσφέρει το σύστημα στους χρήστες της και επισημαίνονται κάποιες ιδιαιτερότητες της υλοποίησης.

3.5.1 Λειτουργίες διαχειριστή

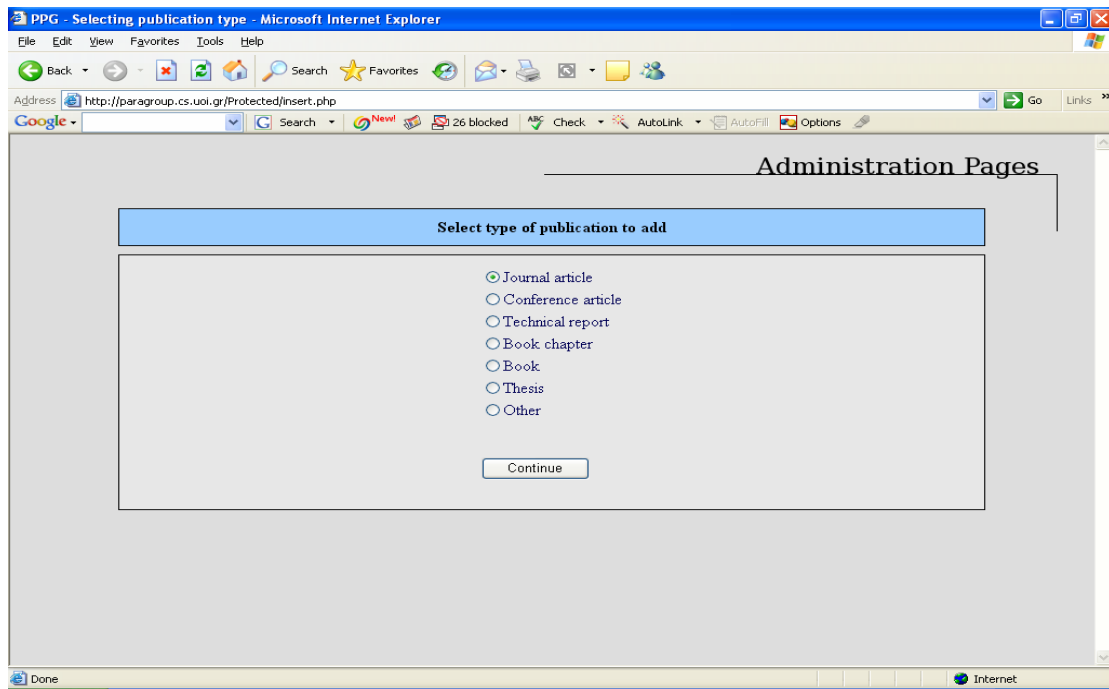
Εάν κάποιος θελήσει να μπει στην ‘Ιδιωτική Περιοχή’ του συστήματος, θα πρέπει να δώσει τον κατάλληλο κωδικό και αφού αυτός γίνει δεκτός θα μπορέσει να αποκτήσει πρόσβαση στις λειτουργίες του διαχειριστή και να χρησιμοποιήσει τις διαθέσιμες υπηρεσίες. Υπάρχουν δύο δυνατές επιλογές: εισαγωγή νέας εργασίας και τροποποίηση / διαγραφή εργασιών.

3.5.1.1 Εισαγωγή νέας εργασίας

Ο χρήστης καλείται να επιλέξει τον τύπο της εργασίας που θέλει να εισάγει (βλ. Εικόνα 3). Ανάλογα με την επιλογή του εμφανίζεται και κατάλληλη φόρμα με τα πεδία της εργασίας που καλείται να τα συμπληρώσει. Υπάρχουν οι εξής δυνατότητες :

- Προσθήκη (upload) μέχρι και τριών διαφορετικών τύπων αρχείων που αναφέρονται στη συγκεκριμένη εργασία, όπως για παράδειγμα η εργασία σε μορφή pdf ή poscript.
- Προσθήκη μέχρι και δύο διαφορετικών συνδέσμων (links) που αναφέρονται στη συγκεκριμένη εργασία, όπως για παράδειγμα σύνδεσμος προς το περιοδικό που δημοσίευσε την εργασία.

Όπως έχει ήδη αναφερθεί, υπάρχουν επτά τύποι εργασιών που μπορεί να εισάγει ο χρήστης. Αρχικά, καλείται να επιλέξει τον τύπο της εργασίας που θέλει να εισάγει και έπειτα να συμπληρώσει την φόρμα που προβάλλεται με τα κατάλληλα στοιχεία της εργασίας. Ακολουθεί μια εικόνα που δείχνει τους τύπους των εργασιών που μπορούν να προστεθούν.



Εικόνα 3: Επιλογή του τύπου της εργασίας που θα εισαχθεί στο σύστημα

Έχει δημιουργηθεί μια κατάλληλη συνάρτηση ούτως ώστε αν υπάρχει ο χαρακτήρας “'” σε κάποιο από τα πεδία που συμπληρώνει ο χρήστης, να προστεθεί ο χαρακτήρας “\” πριν από αυτόν. Αυτό γίνεται για να μην δημιουργηθεί πρόβλημα στη σύνταξη του ερωτήματος εισαγωγής της εργασίας στη βάση του συστήματος.

Στην περίπτωση προσθήκης της εργασίας και σε ηλεκτρονική μορφή, δίνεται η δυνατότητα στο χρήστη να καθορίσει τον τρόπο με τον οποίο αυτή θα εμφανίζεται στη λίστα των εργασιών. Αν δεν συμπληρωθεί το πεδίο ‘**Type**’ (βλ. Εικόνα 4), το σύστημα θα πάρει αυτόματα τον τύπο του αρχείου (π.χ pdf ή image/jpeg), κρίνοντας από την κατάληξη του. Διαφορετικά, στη λίστα των εργασιών η κάθε εργασία θα εμφανίζεται με τον τύπο που έδωσε ο χρήστης στο πεδίο Type. Με αυτόν τον τρόπο, **οποιοσδήποτε τύπος αρχείου** μπορεί να προστεθεί στο σύστημα.

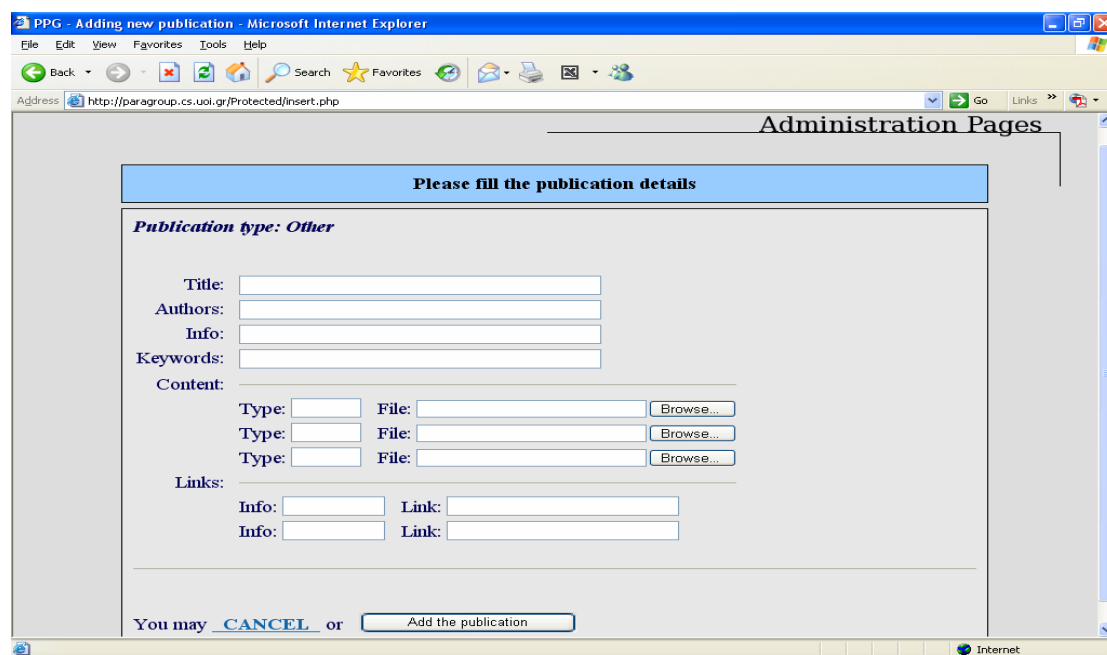
Ο χρήστης μπορεί να εισάγει και συνδέσμους (links) και να καθορίσει το όνομα του κάθε συνδέσμου συμπληρώνοντας το πεδίο ‘**Info**’. Αν το πεδίο αυτό δεν συμπληρωθεί, η εργασία θα εμφανίζεται στη λίστα των εργασιών με το όνομα που έχει δώσει ο χρήστης για τον σύνδεσμο στο πεδίο ‘Link’. Οι σύνδεσμοι αυτοί μπορεί να είναι κάποιες διευθύνσεις ιστοσελίδων που αναφέρονται στη συγκεκριμένη εργασία ή κάποιων σχετικών ιστοσελίδων.

Τέλος, θα πρέπει να σημειωθεί πως η συμπλήρωση του πεδίου ‘**Authors**’ είναι υποχρεωτική. Διαφορετικά, δεν θα μπορέσει να ολοκληρωθεί η εισαγωγή της εργασίας στη βάση δεδομένων του συστήματος.

Αν όλα έχουν γίνει σωστά, η εργασία θα προστεθεί στο σύστημα, μια νέα εγγραφή θα δημιουργηθεί στον πίνακα της βάσης μας και τα uploaded αρχεία θα προστεθούν στον κατάλογο όπου φυλάσσονται όλα τα αρχεία που προστίθενται από τους χρήστες, στον κατάλογο Publications. Επειδή υπάρχει η περίπτωση στον κατάλογο αυτό να υπάρχει ήδη ένα αρχείο με το ίδιο όνομα, μπροστά από το όνομα του κάθε αρχείου που προσθέτουμε με τη διαδικασία upload μπαίνει ένας αύξων αριθμός. Ο μετρητής αυτός κρατείται σε ένα αρχείο που έχουμε δημιουργήσει. Έχει αρχικά την τιμή 1 και κάθε φορά που προστίθεται ένα καινούριο αρχείο στον κατάλογο Publications, το αρχείο με το μετρητή ανοίγεται και η τιμή του αυξάνεται κατά ένα. Αφού προστεθεί η εργασία, ο χρήστης έχει δύο επιλογές :

1. Να επιστρέψει στη λίστα με τις διαθέσιμες υπηρεσίες ή
2. Να περιμένει 5 δευτερόλεπτα οπότε και το σύστημα θα τον επιστρέψει αυτόματα στην επιλογή τύπου εργασίας και στην εισαγωγή μιας νέας εργασίας.

Ακολουθεί ένα παράδειγμα εισαγωγής εργασίας τύπου Other.



The screenshot shows a web browser window titled "PPG - Adding new publication - Microsoft Internet Explorer". The address bar shows "http://paragroup.cs.uoi.gr/Protected/insert.php". The page content includes a header "Administration Pages" and a main form area with the title "Please fill the publication details". The form is for a "Publication type: Other" and contains the following fields and controls:

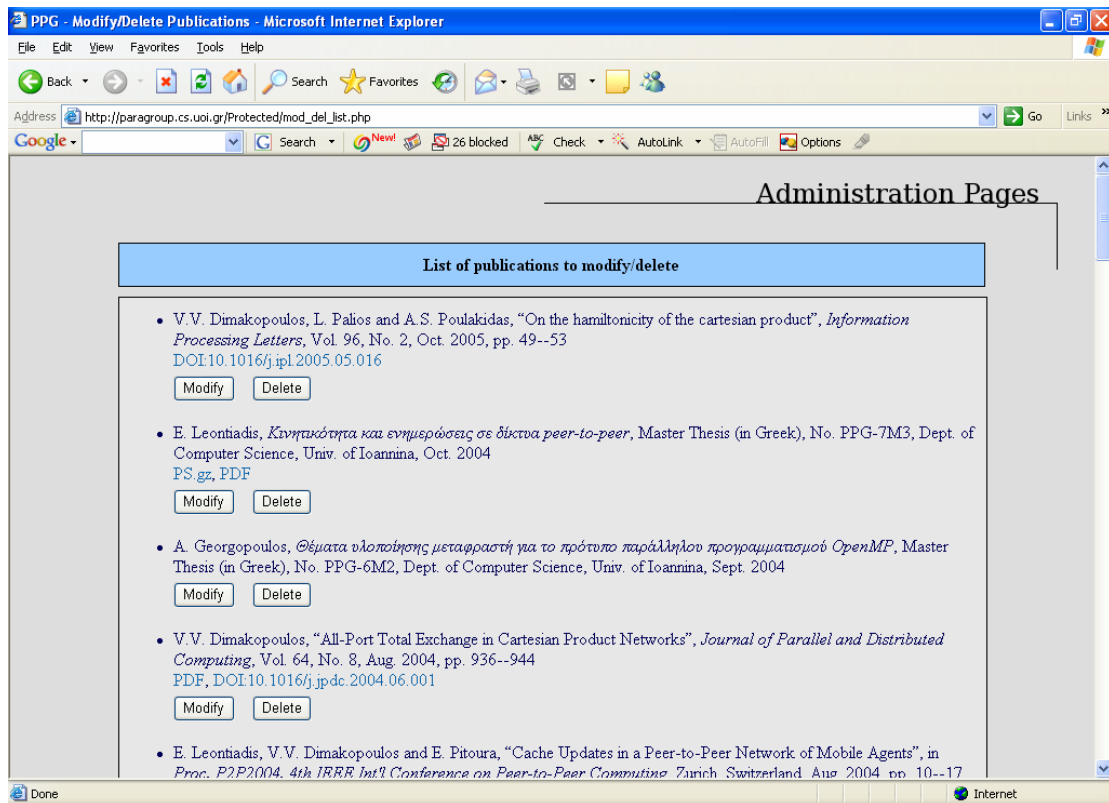
- Title:
- Authors:
- Info:
- Keywords:
- Content:
 - Type: File:
 - Type: File:
 - Type: File:
- Links:
 - Info: Link:
 - Info: Link:

At the bottom of the form, there is a message: "You may [CANCEL](#) or

Εικόνα 4: Παράδειγμα εισαγωγής εργασίας

3.5.1.2 Τροποποίηση / Διαγραφή των εργασιών

Αν ο χρήστης επιλέξει τη διαγραφή ή την τροποποίηση των εργασιών, του εμφανίζεται η λίστα με τις υπάρχουσες εργασίες όπου δίπλα σε κάθε εργασία υπάρχουν τα Modify και Delete buttons (βλ. Εικόνα 5).



Εικόνα 5: Η λίστα των εργασιών με τις δυνατότητες τροποποίησης και διαγραφής

Αν ο χρήστης πατήσει το Delete button, θα καλεστεί να επιβεβαιώσει ότι επιθυμεί τη διαγραφή της συγκεκριμένης εργασίας. Αν δεν επιβεβαιώσει τη διαγραφή της, επιστρέφει πάλι πίσω στη λίστα των εργασιών για να επιλέξει την επόμενη εργασία που θέλει να επεξεργαστεί. Διαφορετικά, η εργασία θα διαγραφεί από τη λίστα των εργασιών με ταυτόχρονη διαγραφή της αντίστοιχης εγγραφής από τη βάση του συστήματος. Αν η εργασία υπάρχει και σε ηλεκτρονική μορφή, τότε όλα τα αντίτυπά της θα διαγραφούν από τον κατάλογο του συστήματος όπου φυλάσσονται τα uploaded αρχεία. Ο χρήστης καλείται έπειτα να επιλέξει την επόμενη του κίνηση είτε να επιστρέψει στη λίστα των εργασιών είτε να επιστρέψει στη λίστα των διαθέσιμων υπηρεσιών.

Αν ο χρήστης πατήσει το Modify button, εμφανίζεται μια φόρμα με όλα τα ήδη αποθηκευμένα στοιχεία της επιλεγμένης εργασίας, μαζί με τα τρία uploaded αρχεία και τους δύο συνδέσμους. Ο χρήστης μπορεί να τροποποιήσει τα πεδία που επιθυμεί και να πατήσει το MODIFY button, οπότε και η αντίστοιχη εγγραφή στον πίνακα του συστήματος ενημερώνεται κατάλληλα. Αν δεν επιθυμεί την αποθήκευση των αλλαγών του, αρκεί να ακολουθήσει τον σύνδεσμο CANCEL για να επιστρέψει πίσω στη λίστα των εργασιών. Θα πρέπει να σημειωθεί ότι και στην περίπτωση της τροποποίησης, το πεδίο '**Authors**' θα πρέπει να μην είναι κενό. Σκοπός της ύπαρξης της δυνατότητας τροποποίησης από το χρήστη είναι:

- Να διορθώσει κάποιο από τα πεδία της εργασίας που ίσως είχε συμπληρώσει λανθασμένα.
- Να διορθώσει τον τύπο του αρχείου που υπάρχει στο πεδίο **Type**.
- Να προσθέσει κάποιο αρχείο, στην περίπτωση που κατά την αρχική εισαγωγή της εργασίας, αυτή δεν συνοδεύταν από κάποιο αρχείο.
- Να προσθέσει κάποια από τα τρία αρχεία που δεν είχε κάνει upload προηγουμένως.
- Να διορθώσει το όνομα του συνδέσμου που υπάρχει στο πεδίο **Link**.
- Να προσθέσει κάποιο σύνδεσμο, στην περίπτωση που κατά την αρχική εισαγωγή της εργασίας, αυτή δεν συνοδεύταν από κάποιο σύνδεσμο.
- Να προσθέσει ένα νέο σύνδεσμο που δεν είχε προσθέσει προηγουμένως..

3.5.2 Η λίστα των εργασιών

3.5.2.1 Αναζήτηση ανάμεσα στις εργασίες

Οποιοσδήποτε επισκέπτης της ιστοσελίδας του συστήματος μπορεί να δει ολόκληρη τη λίστα των εργασιών και να κατεβάσει στον υπολογιστή του όποιες από αυτές επιθυμεί. Μπορεί όμως να επιλέξει να δει κάποιες μόνο από τις υπάρχουσες εργασίες (βλ. Εικόνα 6). Η εμφάνιση κάποιων μόνο από των υπάρχοντων εργασιών επιτυγχάνεται ως εξής:

1) Αναζήτηση λέξης :

Ο χρήστης μπορεί να ψάξει για μια συγκεκριμένη λέξη της αρεσκείας του και να εμφανίσει μόνο τις εργασίες που περιέχουν αυτή τη λέξη στα στοιχεία τους. Έχει τη δυνατότητα να επιλέξει σε ποιο πεδίο των στοιχείων της εργασίας θέλει να ψάξει τη λέξη: Στον τίτλο, στον συγγραφέα, στο έτος, στη λέξη-κλειδί ή και παντού.

2) Επιλογή τύπου :

Ο χρήστης μπορεί να επιλέξει ποιους τύπους εργασιών θέλει να εμφανίσει (Articles, Technical reports, Theses, Others) αρκεί να τσεκάρει τα αντίστοιχα checkboxes.

3) Ταξινόμηση:

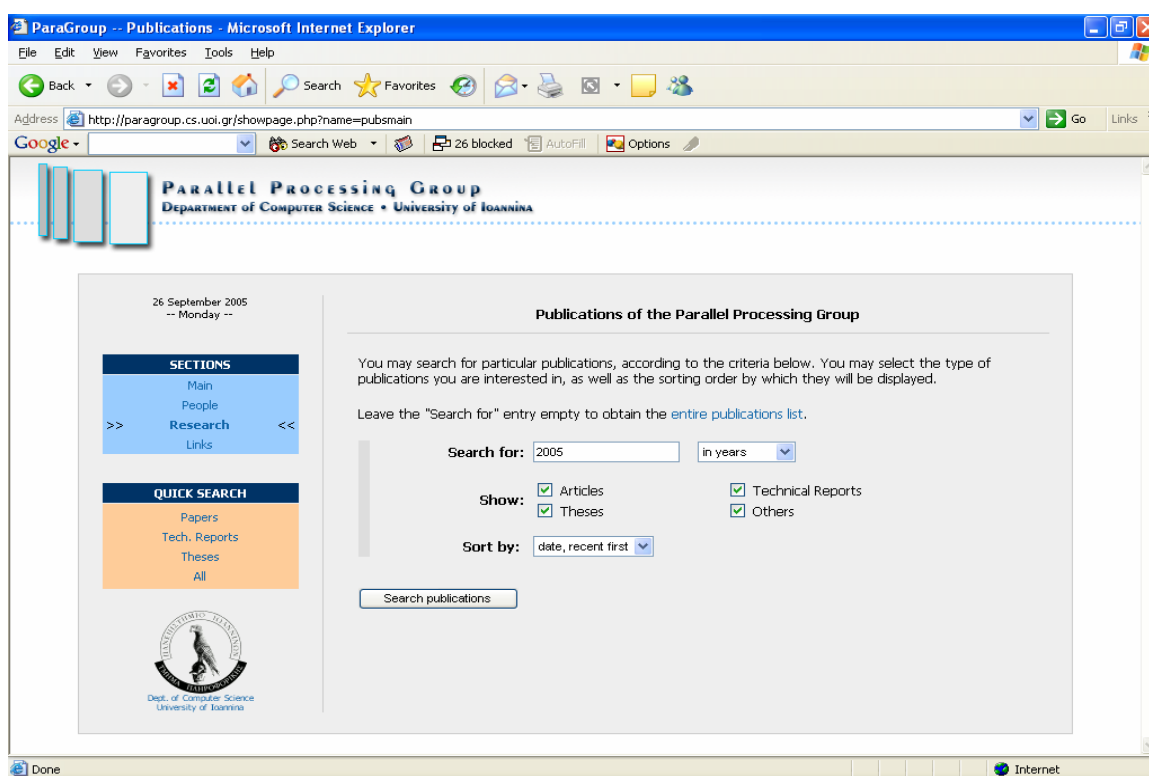
Τα αποτελέσματα της αναζήτησης εμφανίζονται ταξινομημένα με βάση την ημερομηνία. Υπάρχουν δύο επιλογές : Αύξουσα και φθίνουσα ταξινόμηση. Η προεπιλεγμένη ρύθμιση είναι η εμφάνιση σε φθίνουσα χρονολογική σειρά.

Εκτός από αυτόν τον τρόπο επιλογής εργασιών, είναι διαθέσιμη και η επιλογή της γρήγορης αναζήτησης 'QUICK SEARCH' όπου ο επισκέπτης μπορεί να επιλέξει τον

τύπο των εργασιών που θέλει να δει. Ανάλογα με την επιλογή του χρήστη, εμφανίζονται όλες οι εργασίες μόνο αυτού του τύπου. Υπάρχουν τέσσερις επιλογές:

- Papers
- Technical Reports
- Theses
- All

Αν επιλέξει All, τότε θα εμφανιστούν όλες οι εργασίες που υπάρχουν στο σύστημα, ανεξάρτητα από τον τύπο τους. Ακολουθεί ένα παράδειγμα επιλογής εργασιών με βάση την αναζήτηση λέξης σε αυτές, όπου εμφανίζονται όλες οι εργασίες που δημοσιεύθηκαν το 2005 :



Εικόνα 6: Κριτήρια με βάση τα οποία επιλέγονται οι εργασίες που θα εμφανιστούν

3.5.2.2 Αλλαγή του τρόπου εμφάνισης των εργασιών

Ας σημειωθεί ότι ανάλογα με τον τύπο της κάθε εργασίας του συστήματος, τα διάφορα στοιχεία της (π.χ τίτλος, συγγραφέας, έτος δημοσίευσης κ.τ.λ) εμφανίζονται και με διαφορετικό τρόπο. Για παράδειγμα, μπορεί να έχει καθοριστεί πως τα ονόματα των τίτλων όλων των εργασιών τύπου Journal Article θα εμφανίζονται με πλάγια γράμματα. Αυτές οι ρυθμίσεις έχουν καθοριστεί από τον δημιουργό του συστήματος σε ένα css αρχείο. Με αυτόν τον τρόπο παρέχεται μια επιπλέον δυνατότητα: Αλλάζοντας το css αρχείο μπορούμε να αλλάξουμε την εμφάνιση της

ιστοσελίδας και τον τρόπο που εμφανίζονται οι εργασίες στη λίστα των εργασιών χωρίς να χρειαστεί να τροποποιήσουμε τον αντίστοιχο κώδικα που είναι υπεύθυνος για τη μορφή της λίστας των εργασιών.

3.6 Φορητότητα συστήματος

Θα πρέπει να τονισθεί πως το σύστημα διαχείρισης βιβλιογραφίας και δημοσιεύσεων που περιγράψαμε, παρ' όλο που είναι ενσωματωμένο στις ιστοσελίδες του PPG, δεν παύει να είναι αυτόνομο και ικανό να ενσωματωθεί σε οποιονδήποτε ιστότοπο που βασίζεται στο συνδυασμό Apache, PHP και MySQL. Και αυτό γιατί:

- Όλες οι οπτικές ρυθμίσεις βρίσκονται στο css αρχείο. Αρκεί η αλλαγή των ρυθμίσεων αυτών προκειμένου να προσαρμοστεί το σύστημα μας στις ανάγκες του εκάστοτε χρήστη.
- Όλες οι ρυθμίσεις που αφορούν την πρόσβαση στον server και κατόπιν στη βάση του συστήματος, όπως το username και το password, βρίσκονται σε ένα μόνο αρχείο. Η αλλαγή των μεταβλητών αυτού του αρχείου είναι αρκετή για να αποκτήσει το σύστημα πρόσβαση στον server του νέου ιστότοπου.
- Στο ίδιο αρχείο βρίσκονται και οι μεταβλητές που καθορίζουν τον όνομα της βάσης και του πίνακα του συστήματος. Έτσι, όταν κάποιος θελήσει να ενσωματώσει το σύστημα μας στον ιστότοπο του, αρκεί να δημιουργήσει τη βάση και τον πίνακα με πεδία ίδια με τον δικό μας, και στη συνέχεια να τροποποιήσει τις παραπάνω μεταβλητές ώστε αυτές να ανταποκρίνονται στα ονόματα της βάσης και του πίνακα του νέου συστήματος.
- Στο ίδιο αρχείο βρίσκεται και η μεταβλητή που καθορίζει τον κατάλογο στον οποίο αποθηκεύονται οι ηλεκτρονικές μορφές των αρχείων που προστίθενται από τους χρήστες του συστήματος. Η τιμή της μεταβλητής μπορεί να αλλάξει ανάλογα με τον νέο κατάλογο που επιθυμούμε να μεταφέρονται οι εργασίες.

Συμπεραίνουμε λοιπόν πως με λίγες απλές αλλαγές κάποιων ρυθμίσεων μπορούμε να επιτύχουμε τη μεταφορά του συστήματος μας σε οποιοδήποτε ιστότοπο, με την προϋπόθεση βέβαια ότι είναι διαθέσιμα τα εργαλεία Apache, Php και MySQL.

Κεφάλαιο 4

Περιγραφή του File Manager

4.1 Σκοπός δημιουργίας

Το δεύτερο τμήμα της εργασίας περιλαμβάνει τη δημιουργία ενός Web File Manager, με στόχο τη διαχείριση των αρχείων ενός αρχικού (root) καταλόγου και όλων των υποκαταλόγων του. Πρόκειται για ένα κοινόχρηστο χώρο που μπορεί να χρησιμοποιηθεί για να τοποθετεί εκεί μια ερευνητική ομάδα κοινόχρηστα αρχεία όπως ερευνητικά άρθρα, βιβλία, τεχνικές αναφορές, δημοσιεύσεις σε συνέδρια και πολλούς άλλους τύπους αρχείων, ακόμα και τραγούδια ή αρχεία εικόνας. Απαραίτητη είναι η χρήση ενός μυστικού κωδικού ώστε να μην επιτρέπεται η είσοδος στα μη εξουσιοδοτημένα άτομα. Οποιοσδήποτε επισκέπτης της ιστοσελίδας του συστήματος θα μπορεί, αφού έχει γίνει δεκτός ο κωδικός του, να εισέλθει στον κοινόχρηστο αυτό χώρο, να δει τα περιεχόμενα του αρχικού καταλόγου, να εκτελέσει σε αυτά διάφορες λειτουργίες και να μεταφορτώσει (upload) τα δικά του αρχεία αν το επιθυμεί. Βέβαια, ο συγκεκριμένος web file manager μπορεί να γενικευτεί και να τοποθετηθεί σε οποιοδήποτε κατάλογο ενός συστήματος, τα περιεχόμενα του οποίου θέλουμε να διαχειριστούμε.

4.2 Η ασφάλεια του συστήματος

Είναι πολύ σημαντική η ασφάλεια που θα πρέπει να προσφέρει το σύστημα διαχείρισης αρχείων που δημιουργήσαμε. Θα πρέπει να εξασφαλίσουμε την αποφυγή πρόσβασης στον κοινόχρηστο χώρο από μη εξουσιοδοτημένα άτομα. Για την επίτευξη αυτού του σκοπού χρησιμοποιήθηκε η ιδιότητα: **Apache Basic Authentication** που παρέχει ο Apache server του συστήματος μας, όπως και στην περίπτωση του συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων. Ένας καινούριος μυστικός κωδικός δημιουργήθηκε. Έτσι, όταν ο χρήστης προσπαθήσει να εισέλθει στον κοινόχρηστο χώρο θα πρέπει να εισάγει τον κωδικό στο παράθυρο εισαγωγής κωδικού που θα εμφανιστεί. Στην περίπτωση εισόδου σωστού κωδικού θα εισέλθει στον κοινόχρηστο χώρο αλλιώς θα του απαγορευτεί η είσοδος.

Μια ακόμη ρύθμιση που έχει γίνει προκειμένου να παραμένει ασφαλές το σύστημα είναι η εξής: **απαγόρευση εισόδου σε καταλόγους πάνω του αρχικού**. Όλες οι λειτουργίες του συστήματος θα πρέπει να εκτελούνται πάνω σε αρχεία και υποκαταλόγους του αρχικού καταλόγου του κοινόχρηστου χώρου. Δεν θα πρέπει να υπάρξει πρόσβαση σε καταλόγους πάνω από τον αρχικό. Η μεταβλητή στην οποία αποθηκεύεται ο τρέχων κατάλογος ελέγχεται για να βρεθεί αν περιέχει το αλφαριθμητικό ". .". Αν ναι, τότε απαγορεύεται η είσοδος σε αυτό τον κατάλογο γιατί θα βρίσκεται πάνω από τον αρχικό. Ίδιος έλεγχος γίνεται και στη μεταβλητή όπου αποθηκεύεται το αρχείο που έχει επιλέξει ο χρήστης για να εκτελέσει κάποια λειτουργία. Αν υπάρχει το ". .", τότε απαγορεύεται η εκτέλεση της λειτουργίας στο αρχείο που έχει επιλεγεί γιατί ανήκει σε κάποιον κατάλογο πάνω από τον αρχικό.

4.3 Παρεχόμενες υπηρεσίες

4.3.1 Σύντομη περιγραφή

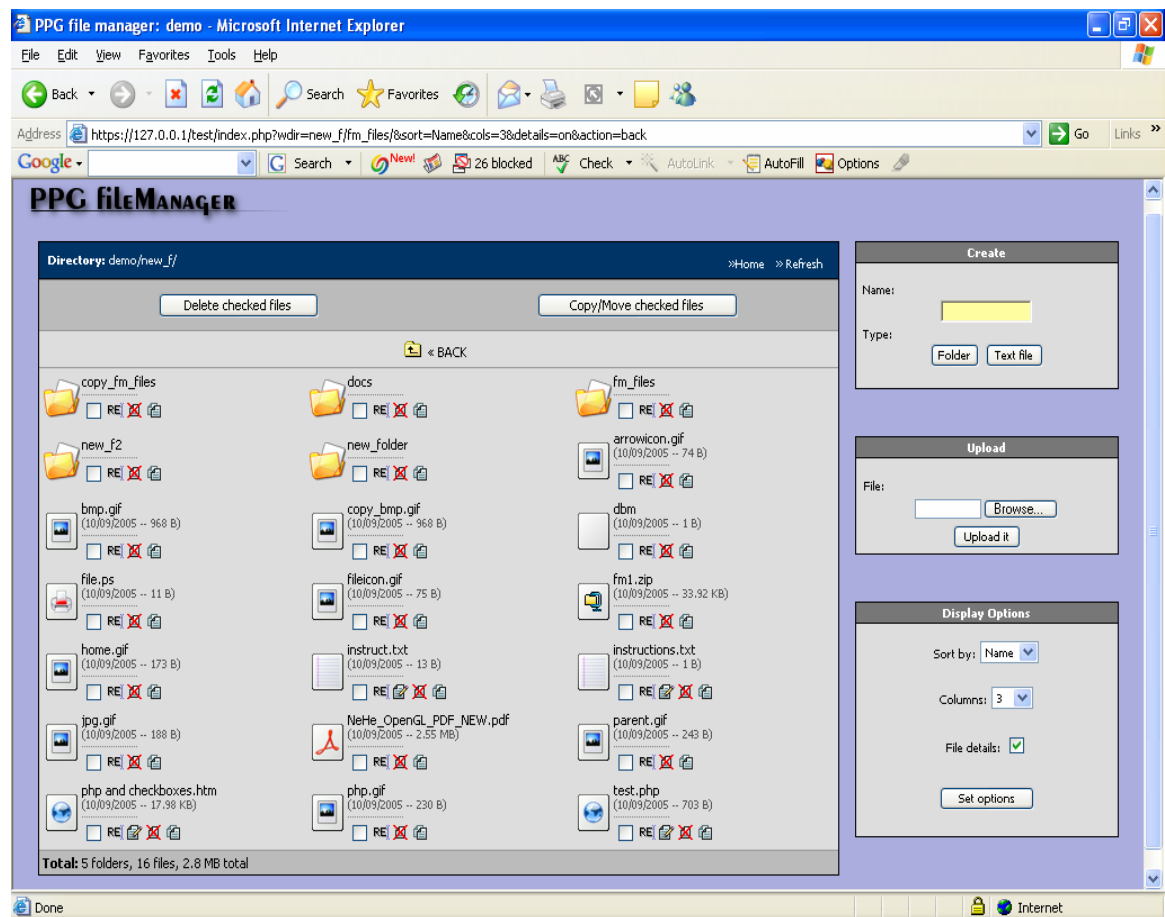
Ο file manager που δημιουργήθηκε παρέχει τις παρακάτω υπηρεσίες:

- Δημιουργία ενός νέου αρχείου στον τρέχοντα κατάλογο.
- Δημιουργία ενός νέου καταλόγου.
- Φόρτωση (upload) ενός αρχείου οποιουδήποτε τύπου.
- Μετονομασία ενός αρχείου ή καταλόγου και αλλαγή του τύπου του.
- Τροποποίηση ενός αρχείου.
- Ατομική / Ομαδική διαγραφή αρχείων και καταλόγων.
- Ατομική / Ομαδική αντιγραφή αρχείων και καταλόγων.
- Ατομική / Ομαδική μετακίνηση αρχείων και καταλόγων.
- Εμφάνιση των περιεχομένων ενός καταλόγου.
- Άνοιγμα ενός αρχείου, θέαση των περιεχομένων του και κατέβασμά του.
- Επιλογή εμφάνισης των περιεχομένων ενός καταλόγου σε στήλες που ο αριθμός τους μπορεί να ποικίλλει: από 1 έως 10.
- Εμφάνιση των δικαιωμάτων των περιεχομένων του τρέχοντος καταλόγου.
- Εμφάνιση κάποιων επιπλέον χαρακτηριστικών των περιεχομένων του τρέχοντος καταλόγου, όπως το μέγεθος και την ημερομηνία τελευταίας τροποποίησης.
- Επιλογή ταξινόμησης των περιεχομένων ενός καταλόγου κατά:
 - Αλφαβητική σειρά
 - Μέγεθος
 - Ημερομηνία τελευταία τροποποίησης
 - Τύπο αρχείου

- Ύπαρξη διαφορετικού εικονιδίου για κάθε τύπο αρχείων.
- Εμφάνιση του συνολικού αριθμού των υποκαταλόγων και των αρχείων του τρέχοντος καταλόγου.
- Εμφάνιση διαφωτιστικών μηνυμάτων σε περίπτωση λάθους.
- Εμφάνιση του τρέχοντος καταλόγου στην κορυφή του file manager.
- Επιστροφή στον αρχικό κατάλογο.
- Δυνατότητα 'Refresh'.

4.3.2 Λειτουργίες πάνω στα αρχεία και τους καταλόγους

Ακολουθεί μια εικόνα όπου παρουσιάζονται τα περιεχόμενα ενός καταλόγου σε τρεις στήλες και όλες οι δυνατότητες που παρέχει ο διαχειριστής αρχείων πάνω σε αυτά.



Εικόνα 7: Εμφάνιση των περιεχομένων ενός καταλόγου και όλων των δυνατών λειτουργιών επί αυτών

Ακολουθεί μια αναλυτικότερη περιγραφή των παρεχόμενων υπηρεσιών που αναφέρθηκαν προηγουμένως και συγκεκριμένα των δυνατοτήτων που παρέχονται στο χρήστη σχετικά με τις λειτουργίες που μπορεί να εκτελέσει πάνω στα αρχεία.

1) Δημιουργία ενός νέου αρχείου στον τρέχοντα κατάλογο

Αρχικά, ο χρήστης θα πρέπει να πληκτρολογήσει ολόκληρο το όνομα του αρχείου (μαζί με την επέκτασή του) και να πατήσει το 'Text file' button (βλ. Εικόνα7). Στη φόρμα που εμφανίζεται, θα πρέπει να δώσει το περιεχόμενό του αρχείου και να πατήσει το Save button.

Ακολουθεί ο έλεγχος του ονόματος που έχει δοθεί. Αρχικά, ελέγχεται αν το όνομα του αρχείου είναι κενό. Σε αυτήν την περίπτωση, εμφανίζεται σχετικό μήνυμα και το αρχείο δε δημιουργείται. Διαφορετικά, από το δοθέν όνομα αρχείου αφαιρούνται κενά που μπορεί να υπάρχουν στην αρχή και στο τέλος καθώς και όλα τα HTML και PHP tags του, με τη χρήση κατάλληλων συναρτήσεων της PHP (όπως την trim και την strip_tags). Επίσης, υπάρχουν κάποιοι χαρακτήρες σε ένα όνομα που θέλουμε να τους αφαιρούμε από αυτό, αντικαθιστώντας τους με τον χαρακτήρα ".". Για παράδειγμα, σε ένα δοθέν όνομα, όπου συναντάται κάποιος από τους παρακάτω χαρακτήρες: ("%","'", "+", "\\", "/", "#", ".", "!", ":", ";", "?", "*", "~") αντικαθιστάται από τον χαρακτήρα ".", δηλαδή αφαιρείται από το όνομα. Έχοντας εκτελέσει όλες τις παραπάνω λειτουργίες, το όνομα είναι πλέον έτοιμο και προχωράμε τη διαδικασία δημιουργίας του αρχείου με αυτό το όνομα.

Αφού τελειώσουμε με τον έλεγχο του ονόματος, προχωράμε στον παρακάτω έλεγχο: εξετάζεται αν το αρχείο ανήκει στην κατηγορία των 'BLOCKED' αρχείων, τα οποία δεν μπορούν να μετονομαστούν, να διαγραφούν, να αντιγραφούν ή να φορτωθούν (upload) στον τρέχοντα κατάλογο. Αν ανήκει, εμφανίζεται σχετικό μήνυμα και το αρχείο δεν δημιουργείται.

Ακόμη, υπάρχει μια ομάδα καταλόγων που δεν επιτρέπουν τη δημιουργία ή τη φόρτωση αρχείων σε αυτά μέσω της διαδικασίας upload. Ελέγχεται λοιπόν εάν ο τρέχων κατάλογος ανήκει σε αυτή την ομάδα των καταλόγων. Αν ναι, η δημιουργία του νέου αρχείου δεν είναι δυνατή.

Σε περίπτωση που όλοι οι έλεγχοι που έχουν γίνει επιτρέπουν τη δημιουργία του αρχείου, προχωράμε σε έναν τελευταίο έλεγχο: ελέγχεται αν το αρχείο που θέλουμε να δημιουργήσουμε υπάρχει ήδη στον κατάλογο. Αν υπάρχει, εμφανίζεται σχετικό μήνυμα λάθους και δεν ολοκληρώνεται η διαδικασία δημιουργίας του αρχείου, αλλιώς προχωράμε στη δημιουργία του αρχείου.

Θα πρέπει να σημειωθεί πως μία από τις ρυθμίσεις που υπάρχουν στο script που υλοποιεί τον διαχειριστή αρχείων είναι και ο καθορισμός της τιμής μιας μεταβλητής που καθορίζει αν είναι δυνατή η εμφάνιση της 'Text file' επιλογής. Αν αυτή η μεταβλητή έχει τιμή 1, είναι δυνατή η εμφάνιση του 'Text file' button, διαφορετικά το 'Text file' button δεν εμφανίζεται στο Create μενού του διαχειριστή αρχείων.

2) Δημιουργία ενός νέου καταλόγου

Παρόμοιοι έλεγχοι εκτελούνται και κατά τη δημιουργία νέου καταλόγου. Υπάρχει πάλι μια καθορισμένη ομάδα καταλόγων που δεν επιτρέπουν τη δημιουργία νέου καταλόγου. Αν ο τρέχων κατάλογος ανήκει σε αυτούς, δεν εμφανίζεται η επιλογή 'Folder' στο Create μενού του διαχειριστή αρχείων (βλ. Εικόνα 7), με αποτέλεσμα να μην είναι δυνατή η δημιουργία νέου καταλόγου.

Διαφορετικά, η επιλογή είναι δυνατή και προχωράμε στην επεξεργασία του ονόματος που δίνεται από τον χρήστη. Αρχικά, ελέγχεται αν το όνομα καταλόγου είναι κενό. Σε αυτήν την περίπτωση, εμφανίζεται σχετικό μήνυμα και ο κατάλογος δε δημιουργείται. Έπειτα, αφαιρούνται και πάλι από το όνομα του καταλόγου τα κενά και οι ειδικοί χαρακτήρες όπως και πριν. Στη συνέχεια, ελέγχουμε αν ανήκει στα 'BLOCKED' ονόματα, αν ο τρέχων κατάλογος επιτρέπει τη δημιουργία νέου αρχείου και αν υπάρχει ήδη ένας κατάλογος με το ίδιο όνομα οπότε και δεν δημιουργείται. Σε διαφορετική περίπτωση, δημιουργείται ένας νέος κενός κατάλογος με το δοθέν όνομα και με τα δικαιώματα 777.

3) Φόρτωση (upload) ενός αρχείου οποιουδήποτε τύπου

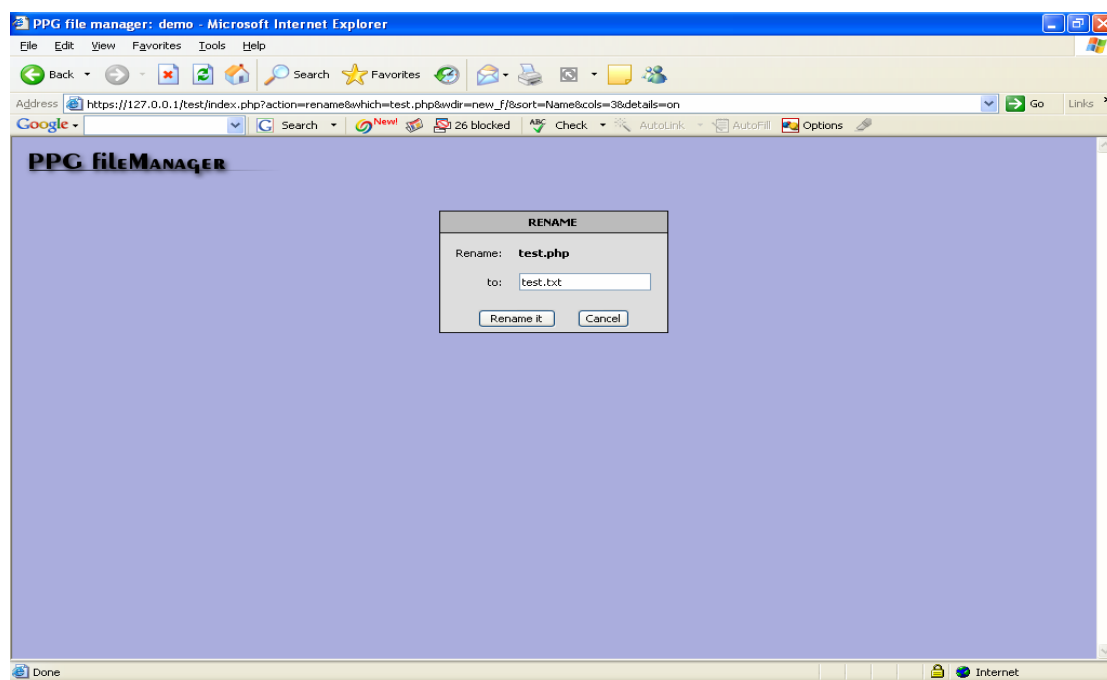
Αν ο τρέχων κατάλογος δεν επιτρέπει τη φόρτωση αρχείων, δεν είναι δυνατή η επιλογή 'Upload'. Σε διαφορετική περίπτωση, ελέγχεται αν το όνομα αρχείου που έχει δοθεί είναι κενό οπότε και εμφανίζεται σχετικό μήνυμα στον χρήστη που τον ενημερώνει πως πρέπει πρώτα να επιλέξει το αρχείο που θέλει να φορτώσει και μετά να πατήσει το 'Upload' button. Ακολουθεί η αφαίρεση κάποιων ειδικών χαρακτήρων από το όνομα του αρχείου που θέλει να φορτώσει ο χρήστης και ελέγχεται αν το μέγεθος του αρχείου είναι μεγαλύτερο του μηδενός και μικρότερο του μέγιστου επιτρεπόμενου μεγέθους αρχείου που μπορεί να φορτωθεί. Ακολουθεί ο έλεγχος αν η προσθήκη του νέου αρχείου στον κατάλογο θα ξεπεράσει το μέγιστο επιτρεπόμενο μέγεθος του καταλόγου και αν ανήκει στην κατηγορία των 'BLOCKED' αρχείων. Σε κάθε μια από αυτές τις περιπτώσεις εμφανίζεται κατάλληλο μήνυμα λάθους και δεν ολοκληρώνεται η διαδικασία φόρτωσης του αρχείου. Διαφορετικά, το αρχείο προστίθεται τελικά στον τρέχοντα κατάλογο.

4) Μετονομασία ενός αρχείου ή καταλόγου και αλλαγή του τύπου του

Δίπλα σε κάθε αρχείο ή κατάλογο υπάρχει το 'RE' εικονίδιο (βλ. Εικόνα 7) με το οποίο μας προσφέρεται η δυνατότητα μετονομασίας του. Ο χρήστης επιλέγει το αρχείο ή τον κατάλογο που θέλει να μετονομάσει οπότε και καλείται να δώσει το νέο

όνομα του. Γίνεται έλεγχος αν το όνομα που έχει δοθεί είναι κενό και στη συνέχεια επεξεργάζεται το δοθέν όνομα, ούτως ώστε να αφαιρεθούν από αυτό τα κενά και κάποιοι ειδικοί χαρακτήρες. Έπειτα, ελέγχεται αν το προς μετονομασία αρχείο ανήκει στην κατηγορία των 'BLOCKED' αρχείων οπότε και δεν μπορεί να μετονομαστεί. Σε κάθε τέτοια περίπτωση, εμφανίζεται κατάλληλο μήνυμα λάθους και δεν ολοκληρώνεται η διαδικασία μετονομασίας του αρχείου ή του καταλόγου. Διαφορετικά, γίνεται ο τελευταίος έλεγχος, όπου στην περίπτωση του αρχείου ελέγχεται αν υπάρχει ήδη ένα αρχείο ίδιου ονόματος και τύπου και στην περίπτωση του καταλόγου αν υπάρχει ένας κατάλογος με το ίδιο όνομα. Αν δεν υπάρχει, η διαδικασία μετονομασίας ολοκληρώνεται επιτυχώς.

Θα πρέπει να σημειωθεί πως στη διαδικασία μετονομασίας ενός αρχείου, συμπεριλαμβάνεται και η διαδικασία αλλαγής του τύπου του. Μπορεί κάποιος να αλλάξει και το όνομα του αρχείου αλλά και τον τύπο του ταυτόχρονα, απλά δίνοντας το νέο όνομα του μαζί με τη νέα επέκτασή του. Ακολουθεί ένα παράδειγμα που επεξηγεί τα παραπάνω, όπου το αρχείο 'test.php' μετονομάζεται στο αντίστοιχο 'test.txt'.

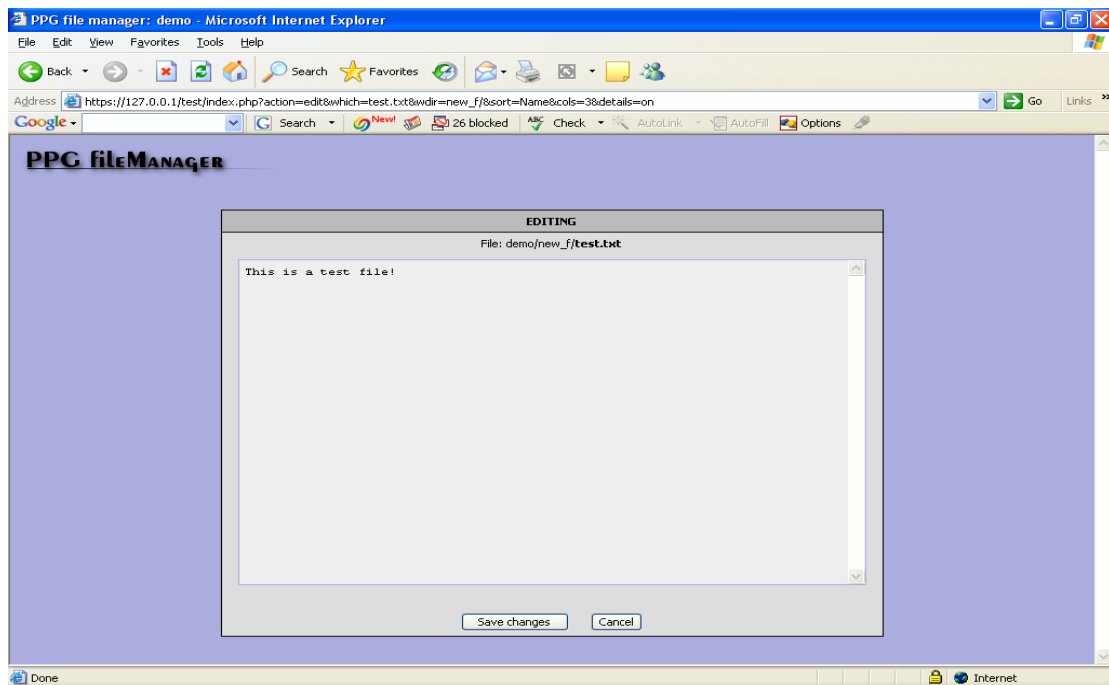


Εικόνα 8: Μετονομασία και ταυτόχρονη αλλαγή του τύπου ενός αρχείου

5) Τροποποίηση ενός αρχείου

Η επιλογή αυτή είναι δυνατή μόνο για τα αρχεία και όχι για τους καταλόγους. Για να μπορεί να εμφανίζεται το εικονίδιο με την edit επιλογή στη λίστα των δυνατών επιλογών ενός αρχείου (βλ. Εικόνα 7), θα πρέπει ο τύπος του να ανήκει στη λίστα με

τους τύπους αρχείων που επιτρέπεται να τροποποιηθούν, η οποία αποτελεί μια από τις ρυθμίσεις του file manager κι έχει καθοριστεί αρχικά από το συγγραφέα του script κατά τη δημιουργία του. Ο χρήστης επιλέγει το αρχείο που θέλει να τροποποιήσει και του εμφανίζεται η φόρμα με το παλιό περιεχόμενο του αρχείου. Μπορεί τώρα να τροποποιήσει το περιεχόμενό του και να πατήσει το 'Save changes' button για να αποθηκεύσει τις αλλαγές (βλ. Εικόνα 9). Έπειτα, με τη χρήση της συνάρτησης stripslashes (PHP function), αφαιρούνται από το νέο περιεχόμενο του αρχείου οι χαρακτήρες '\' ενώ ο χαρακτήρας '\\' μετατρέπεται στον '\'. Αν το αρχείο δεν επιτρέπει την εγγραφή σε αυτό, εμφανίζεται μήνυμα που δηλώνει πως δεν υπάρχει το δικαίωμα εγγραφής και η διαδικασία τροποποίησής του αρχείου σταματά.. Διαφορετικά, το αρχείο τροποποιείται επιτυχώς. Ακολουθεί ένα παράδειγμα τροποποίησης ενός αρχείου:



Εικόνα 9: Παράδειγμα τροποποίησης ενός αρχείου

6) Ατομική / Ομαδική διαγραφή αρχείων και καταλόγων

Προσφέρονται δύο δυνατές επιλογές για τη διαγραφή των αρχείων και των καταλόγων:

6.α) Ατομική διαγραφή

Στις δυνατές επιλογές που εμφανίζονται για κάθε αρχείο ή κατάλογο, εκτός από τα εικονίδια μετονομασίας και τροποποίησης, υπάρχει και το εικονίδιο της διαγραφής

(βλ. Εικόνα 7). Έτσι, ο χρήστης μπορεί να ξεκινήσει τη διαδικασία διαγραφής ενός αντικειμένου, επιλέγοντας το αντίστοιχο εικονίδιο της διαγραφής. Στη συνέχεια καλείται να επιβεβαιώσει ότι επιθυμεί τη διαγραφή του ολοκληρώνοντας έτσι τη διαδικασία διαγραφής ή να την ακυρώσει σε περίπτωση λανθασμένης επιλογής.

6.β) Ομαδική διαγραφή

Στην περίπτωση που υπάρχει η ανάγκη διαγραφής δύο ή περισσότερων αντικειμένων, προσφέρεται η δυνατότητα της ομαδικής διαγραφής των αρχείων και καταλόγων. Ο χρήστης μπορεί να επιλέξει τα αρχεία ή τους καταλόγους που θέλει να διαγράψει, τσεκάροντας το αντίστοιχο 'checkbox', και πατώντας το 'Delete checked files' button (βλ. Εικόνα 7) για να ολοκληρώσει τη διαδικασία διαγραφής. Καλείται στη συνέχεια να επιβεβαιώσει τη διαγραφή των επιλεγμένων αντικειμένων ή να την ακυρώσει σε περίπτωση λανθασμένης επιλογής. Θα πρέπει να τονιστεί πως θα διαγραφούν όλα τα επιλεγμένα αντικείμενα. Μάλιστα, στην περίπτωση ενός καταλόγου, θα διαγραφούν πρώτα όλα τα αρχεία και οι υποκατάλογοί του και μετά ο ίδιος ο κατάλογος. Στην περίπτωση ενός αρχείου, ελέγχεται αν αυτό είναι 'BLOCKED' οπότε και η διαδικασία διαγραφής του σταματά εμφανίζοντας κατάλληλο μήνυμα.

Σε περίπτωση που πατηθεί το 'Delete checked files' button χωρίς να έχει επιλεγεί κάποιο αρχείο προηγουμένως, εμφανίζεται σχετικό πληροφοριακό μήνυμα στο χρήστη που τον ενημερώνει πως πρέπει πρώτα να επιλέξει τα αντικείμενα που θέλει να διαγράψει και μετά να πατήσει το 'Delete checked files' button.

7) Ατομική / Ομαδική αντιγραφή αρχείων και καταλόγων

Προσφέρονται δύο δυνατές επιλογές για την αντιγραφή των αρχείων και των καταλόγων:

7.α) Ατομική αντιγραφή

Στις δυνατές επιλογές που εμφανίζονται για κάθε αρχείο ή κατάλογο, εκτός από τα εικονίδια μετονομασίας, τροποποίησης και διαγραφής, υπάρχει και το εικονίδιο της αντιγραφής/μετακίνησης (βλ. Εικόνα 7). Έτσι, ο χρήστης μπορεί να ξεκινήσει τη διαδικασία αντιγραφής ενός αντικειμένου, επιλέγοντας το αντίστοιχο εικονίδιο της αντιγραφής/μετακίνησης. Στη συνέχεια καλείται να επιλέξει αν επιθυμεί την αντιγραφή ή τη μετακίνηση του αντικειμένου και τον κατάλογο προορισμού

ψάχνοντας μέσα σε μια λίστα με όλους τους καταλόγους στους οποίους έχει τη δυνατότητα να αντιγράψει αρχεία.

Από αυτή τη λίστα (βλ. Εικόνα 10) εξαιρούνται οι κατάλογοι που δεν επιτρέπουν την αντιγραφή ή τη μεταφορά αρχείων σε αυτούς, όπως αυτοί έχουν καθοριστεί από το δημιουργό του script. Πρόκειται για αυτούς που δεν επιτρέπουν τη φόρτωση αρχείων σε αυτούς. Αν βρισκόμαστε στον root κατάλογο, η λίστα περιλαμβάνει όλους τους υποκαταλόγους του μαζί με τον ίδιο τον root κατάλογο, ενώ αν βρισκόμαστε μέσα σε κάποιον από τους υποκαταλόγους του root, η λίστα περιλαμβάνει ότι και πριν δηλαδή όχι μόνο τους υποκαταλόγους του τρέχοντος καταλόγου αλλά όλους τους υποκαταλόγους του root, ακόμα και αυτούς που βρίσκονται πάνω από τον τρέχοντα κατάλογο. Ας σημειωθεί ότι στη λίστα των καταλόγων προορισμού, ο τρέχων κατάλογος εμφανίζεται με μπλε χρώμα για να ξεχωρίζει από τους υπόλοιπους.

Έτσι, ο χρήστης πατώντας το 'Do it' button (βλ. Εικόνα 10) ολοκληρώνει τη διαδικασία αντιγραφής του αντικείμενου στον κατάλογο προορισμού. Σε κάθε περίπτωση, το νέο αντικείμενο που δημιουργείται έχει τα ίδια δικαιώματα με το αρχικό αντικείμενο. Στην περίπτωση που ο χρήστης έχει επιλέξει να αντιγράψει έναν κατάλογο, ένας νέος κατάλογος δημιουργείται στον κατάλογο προορισμού που θα έχει το ίδιο όνομα με τον προς αντιγραφή κατάλογο. Αρχικά, ο κατάλογος αυτός δημιουργείται με τα δικαιώματα 0777. Σε αυτόν αντιγράφονται όλα τα αρχεία και όλοι υποκατάλογοι του προς αντιγραφή καταλόγου. Έπειτα και αφού έχει καθοριστεί το περιεχόμενο του, αλλάζουμε τα δικαιώματα του για να είναι ίδια με του αρχικού καταλόγου. Στην περίπτωση που ο χρήστης έχει επιλέξει να αντιγράψει ένα αρχείο, ελέγχεται αν αυτό είναι 'BLOCKED' οπότε και εμφανίζεται σχετικό μήνυμα και το αρχείο δεν αντιγράφεται στον κατάλογο προορισμού. Διαφορετικά, ένα νέο αρχείο δημιουργείται στον κατάλογο προορισμού με το όνομα και τα δικαιώματα του προς αντιγραφή αρχείου.

Μια ιδιαιτερότητα της υλοποίησης της διαδικασίας αντιγραφής ενός αντικείμενου είναι η εξής: Αν το αντικείμενο που θέλουμε να αντιγράψουμε υπάρχει ήδη μέσα στον κατάλογο προορισμού, το νέο αντικείμενο που δημιουργείται ονομάζεται copy_filename όπου filename το όνομα του παλιού αντικείμενου. Δηλαδή, κάθε φορά που δημιουργείται ένα αντίγραφο ενός ήδη υπάρχοντος αντικείμενου, προστίθεται στην αρχή του ονόματος του το αλφαριθμητικό 'copy_', ώστε να αποφύγουμε την αντικατάσταση του παλιού αρχείου από το νέο αντίγραφο του. Με τον τρόπο αυτό, προσφέρεται και η δυνατότητα αντιγραφής ενός αρχείου μέσα στον ίδιο τον κατάλογο που ανήκει.

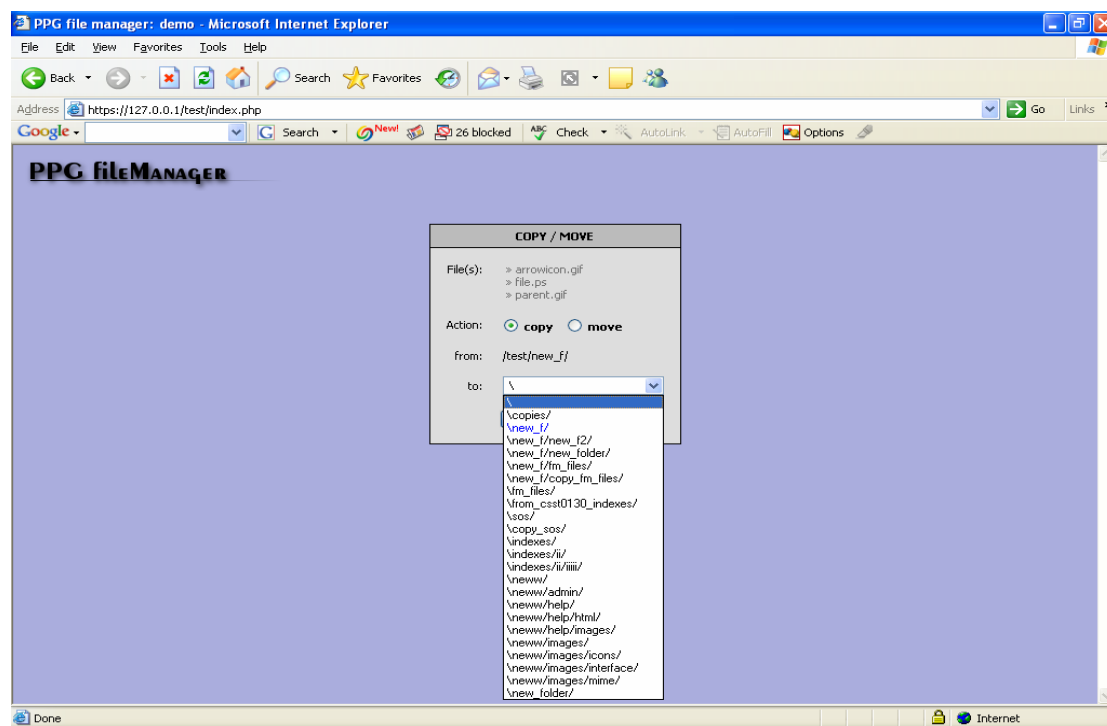
Μια δεύτερη ιδιαιτερότητα της διαδικασίας αντιγραφής είναι η εξής: Δεν επιτρέπεται ο κατάλογος προορισμού να είναι ίδιος με τον προς αντιγραφή κατάλογο.

Για την αποφυγή της αναδρομής αυτής ελέγχεται αν ο κατάλογος που θέλουμε να αντιγράψουμε είναι ο ίδιος με αυτόν που έχουμε επιλέξει στη λίστα των καταλόγων προορισμού. Αν ισχύει κάτι τέτοιο, δεν αντιγράφεται ο επιλεγμένος κατάλογος. Σε κάθε περίπτωση εμφανίζεται διαφωτιστικό μήνυμα που ενημερώνει τον χρήστη για το αποτέλεσμα της διαδικασίας αντιγραφής

7.β) Ομαδική αντιγραφή

Ο χρήστης μπορεί να επιλέξει τα αρχεία ή τους καταλόγους που θέλει να αντιγράψει, τσεκάροντας το αντίστοιχο 'checkbox', και πατώντας το 'Copy/Move checked files' button για να ολοκληρώσει τη διαδικασία αντιγραφής. Καλείται στη συνέχεια να επιλέξει αν θέλει να εκτελέσει την 'Copy' ή την 'Move' λειτουργία καθώς και τον κατάλογο προορισμού με τον ίδιο τρόπο που περιγράφηκε και στην ατομική αντιγραφή. Θα πρέπει να τονιστεί πως θα αντιγραφούν όλα τα επιλεγμένα αντικείμενα. Ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα αρχεία και να τα αντιγράψει όλα στον κατάλογο προορισμού.

Σε περίπτωση που πατηθεί το 'Copy/Move checked files' button (βλ. Εικόνα7) χωρίς να έχει επιλεγεί κάποιο αρχείο προηγουμένως, εμφανίζεται σχετικό πληροφοριακό μήνυμα στο χρήστη που τον ενημερώνει πως πρέπει πρώτα να επιλέξει τα αντικείμενα που θέλει να αντιγράψει ή να μετακινήσει και μετά να πατήσει το 'Copy/Move checked files' button.



Εικόνα 10: Παράδειγμα ομαδικής αντιγραφής αρχείων

8) *Ατομική / Ομαδική μετακίνηση αρχείων και καταλόγων*

Προσφέρονται δύο δυνατές επιλογές για τη μετακίνηση των αρχείων και των καταλόγων:

8.α) Ατομική μετακίνηση

Ο χρήστης μπορεί να ξεκινήσει τη διαδικασία μετακίνησης ενός αντικειμένου, επιλέγοντας το αντίστοιχο εικονίδιο της αντιγραφής/μετακίνησης. Στη συνέχεια καλείται να επιλέξει αν επιθυμεί την αντιγραφή ή τη μετακίνηση του αντικειμένου και τον κατάλογο προορισμού ψάχνοντας μέσα σε μια λίστα με όλους τους καταλόγους στους οποίους έχει τη δυνατότητα να μεταφέρει αρχεία. Η λίστα είναι η ίδια που εμφανίζεται και στην περίπτωση της αντιγραφής

Έτσι, ο χρήστης πατώντας το 'Do it' button (βλ. Εικόνα 10) ολοκληρώνει τη διαδικασία μετακίνησης του αντικειμένου στον κατάλογο προορισμού. Θα πρέπει να τονιστεί πως θα μετακινηθούν όλα τα προεπιλεγμένα αντικείμενα. Σε κάθε περίπτωση, το νέο αντικείμενο που δημιουργείται έχει τα ίδια δικαιώματα με το αρχικό αντικείμενο. Στην περίπτωση που ο χρήστης έχει επιλέξει να μετακινήσει έναν κατάλογο, ένας νέος κατάλογος δημιουργείται στον κατάλογο προορισμού όπου και αντιγράφονται όλα τα αρχεία και όλοι υποκατάλογοι του προς μεταφορά καταλόγου και διαγράφονται έπειτα από τον τρέχοντα κατάλογο. Στην περίπτωση ενός αρχείου, ελέγχεται αν αυτό είναι 'BLOCKED' οπότε και εμφανίζεται σχετικό μήνυμα και το αρχείο δεν αντιγράφεται στον κατάλογο προορισμού. Διαφορετικά το αρχείο μπορεί να μετακινηθεί, οπότε αντιγράφεται στον προορισμό αρχικά κι έπειτα διαγράφεται από τον κατάλογο προέλευσης.

Μια ιδιαιτερότητα της υλοποίησης της διαδικασίας μεταφοράς ενός αντικειμένου είναι η εξής: Αν το αντικείμενο που θέλουμε να μεταφέρουμε υπάρχει ήδη μέσα στον κατάλογο προορισμού, το νέο αντικείμενο που δημιουργείται ονομάζεται `copy_filename` όπου `filename` το όνομα του παλιού αντικειμένου, ώστε να αποφύγουμε την αντικατάσταση του παλιού αρχείου από το νέο αντίγραφο του που μεταφέρθηκε στον κατάλογο προορισμού. Σε κάθε περίπτωση εμφανίζεται διαφωτιστικό μήνυμα που ενημερώνει τον χρήστη για το αποτέλεσμα της διαδικασίας μετακίνησης του αντικειμένου.

8.β) Ομαδική μετακίνηση

Ο χρήστης μπορεί να επιλέξει τα αρχεία ή τους καταλόγους που θέλει να μετακινήσει, τσεκάροντας το αντίστοιχο 'checkbox', και πατώντας το 'Copy/Move checked files' button για να ολοκληρώσει τη διαδικασία μεταφοράς τους. Καλείται

στη συνέχεια να επιλέξει αν θέλει να εκτελέσει την ‘Copy’ ή την ‘Move’ λειτουργία καθώς και τον κατάλογο προορισμού με τον ίδιο τρόπο που περιγράφηκε και στην ατομική μετακίνηση. Θα πρέπει να τονιστεί πως θα μεταφερθούν όλα τα επιλεγμένα αντικείμενα. Ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα αρχεία και να τα μεταφέρει όλα στον κατάλογο προορισμού.

Σε περίπτωση που πατηθεί το ‘Copy/Move checked files’ button χωρίς να έχει επιλεγεί κάποιο αρχείο προηγουμένως, εμφανίζεται σχετικό πληροφοριακό μήνυμα στο χρήστη που τον ενημερώνει πως πρέπει πρώτα να επιλέξει τα αντικείμενα που θέλει να μετακινήσει και μετά να πατήσει το ‘Copy/Move checked files’ button.

9) *Εμφάνιση των περιεχομένων ενός καταλόγου*

Το όνομα κάθε υποκαταλόγου είναι link που οδηγεί στα περιεχόμενα του. Όταν λοιπόν ο χρήστης πατήσει πάνω στο όνομα ενός υποκαταλόγου, ανιχνεύουμε όλα τα αρχεία και τους καταλόγους του, με τη χρήση της συνάρτησης `readdir` της PHP, και τα εμφανίζουμε στην οθόνη του file manager.

10) *Ανοιγμα ενός αρχείου, θέαση των περιεχομένων του και κατέβασμά του*

Το όνομα κάθε αρχείου είναι ένα link, που μας οδηγεί στην εμφάνιση των περιεχομένων του αρχείου. Υπάρχουν 19 τύποι αρχείων που υποστηρίζονται από το file manager μας και περιγράφονται στον παρακάτω πίνακα:

html	htm	css	asp	php	jpg	jpeg
bmp	gif	png	txt	pdf	ps	Eps
zip	rar	tar	gz	Other		

Αν ένα αρχείο δεν ανήκει σε κανένα από τους 18 παραπάνω τύπους, θεωρείται τύπου ‘Other’. Σε κάθε περίπτωση, πατώντας το link εμφανίζεται το περιεχόμενο του αρχείου από εκεί και πέρα μπορούμε απλά να δούμε το περιεχόμενό του αρχείου ή και να το αποθηκεύσουμε στον υπολογιστή μας εάν θέλουμε. Στην περίπτωση ενός αρχείου τύπου Other, εμφανίζεται ένα κατάλληλο ‘File Download’ μενού όπου ο χρήστης καλείται να επιλέξει αν θέλει να ανοίξει το αρχείο ή να το σώσει τοπικά στον υπολογιστή του.

Σε όλες αυτές τις περιπτώσεις, η διαδικασία που ακολουθείται είναι η ίδια. Χρησιμοποιώντας κατάλληλες συναρτήσεις της PHP, παίρνουμε κάποιες πληροφορίες για το αρχείο, όπως το μέγεθος και τον τύπο του (χρήση της

συνάρτησης `stat()`) κι έπειτα διαβάζουμε το αρχείο και τυπώνουμε το περιεχόμενό του στην οθόνη (χρήση της συνάρτησης `readfile()`).

4.3.3 Επιλογές για την εμφάνιση των αρχείων

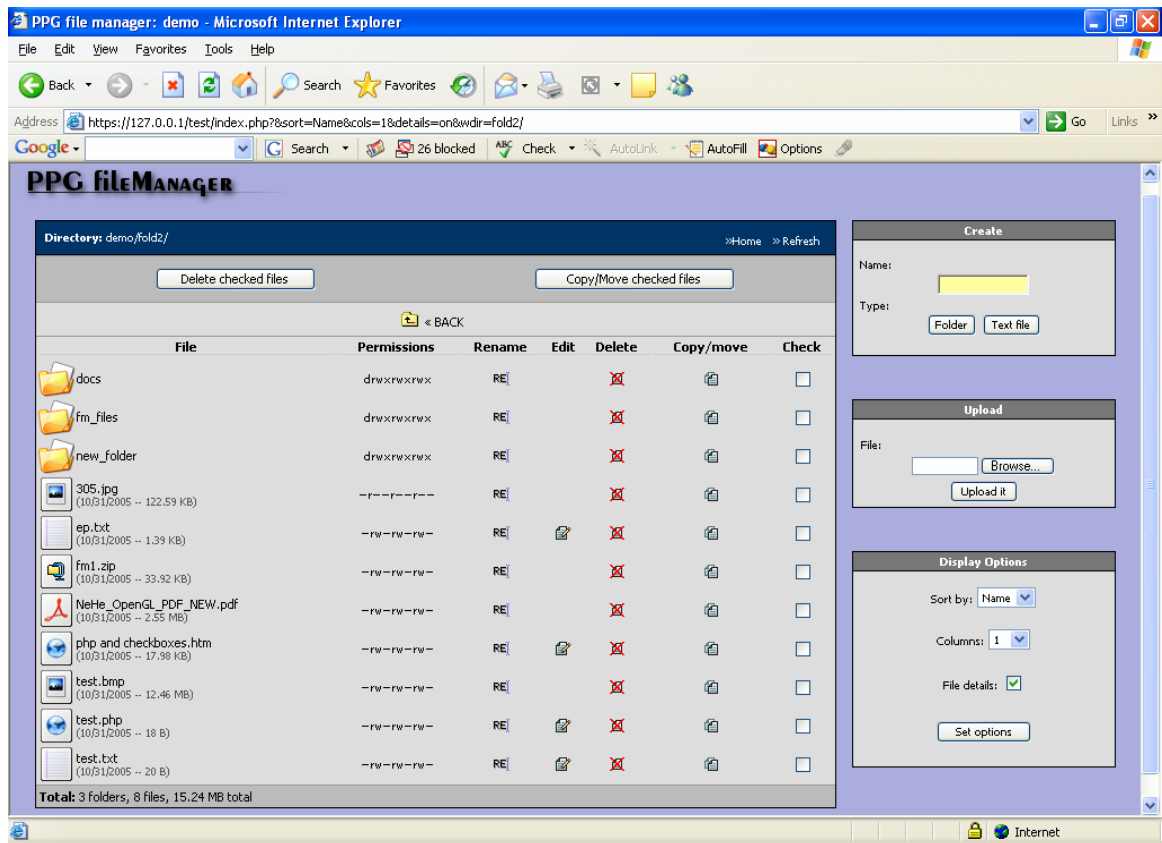
Στη συνέχεια ακολουθεί μια περιγραφή των τρόπων με τους οποίους ο χρήστης μπορεί να αλλάξει τον τρόπο εμφάνισης των περιεχομένων του τρέχοντος καταλόγου. Όλες αυτές οι δυνατές επιλογές βρίσκονται στο *'Display options'* μενού του διαχειριστή αρχείων.

1) *Επιλογή εμφάνισης των περιεχομένων ενός καταλόγου σε στήλες που ο αριθμός τους μπορεί να ποικίλλει: από 1 έως 10*

Όταν μπαίνουμε σε έναν κατάλογο εμφανίζονται όλα τα περιεχόμενά του, πρώτα οι κατάλογοι και μετά τα αρχεία, ταξινομημένα κατά αλφαβητική σειρά και σε πίνακα των 3 στηλών. Αυτές είναι οι προεπιλεγμένες ρυθμίσεις τις οποίες όμως ο χρήστης έχει τη δυνατότητα να τις αλλάξει. Στο *'Display Options'* μενού του file manager (βλ. Εικόνα 7) υπάρχει μια επιλογή που επιτρέπει στον χρήστη να επιλέξει τον αριθμό των στηλών εμφάνισης. Μπορεί να επιλέξει οποιαδήποτε αριθμό από το 1 έως το 10. Σε περίπτωση επιλογής του αριθμού 1, έχουμε εμφάνιση σε μία στήλη δηλαδή εμφάνιση σε μορφή κατακόρυφης λίστας (βλ. Εικόνα 11).

Έτσι λοιπόν, θα μπορούσαμε να πούμε ότι ένα σημαντικό πλεονέκτημα του file manager μας είναι πως επιτρέπει στον χρήστη να βλέπει τα περιεχόμενα ενός καταλόγου σε μορφή λίστας αλλά και σε μορφή πίνακα, δίνοντας του τη δυνατότητα να επιλέγει αυτός τον τρόπο εμφάνισης κάθε φορά, ανάλογα με το πλήθος των περιεχομένων του κάθε φακέλου. Αν για παράδειγμα ένας φάκελος έχει λίγα αρχεία, ο χρήστης θα μπορούσε να τα βλέπει σε μορφή λίστας ή σε μικρό αριθμό στηλών ενώ αν ο αριθμός των αρχείων είναι μεγάλος, θα ήταν πιο λειτουργική η εμφάνιση τους σε μεγάλο αριθμό στηλών. Και αυτό γιατί αν είχε επιλέξει την εμφάνιση σε μορφή κατακόρυφης λίστας, δεν θα μπορούσε εύκολα να διαχειριστεί τα αρχεία καθώς η λίστα θα ήταν μακροσκελής και άρα μη λειτουργική.

Ακολουθεί ένα παράδειγμα εμφάνισης των περιεχομένων ενός καταλόγου σε μορφή λίστας, όπου δίπλα σε κάθε αντικείμενο υπάρχουν τα εικονίδια με τις δυνατές λειτουργίες που μπορεί να εκτελέσει ο χρήστης πάνω σε αυτά.



Εικόνα 11: Εμφάνιση των περιεχομένων ενός καταλόγου σε κατακόρυφη λίστα

2) *Εμφάνιση των δικαιωμάτων των περιεχομένων του τρέχοντος καταλόγου*

Όταν ο χρήστης επιλέξει εμφάνιση σε μορφή λίστας, εκτός από τα αρχεία και τις δυνατές επιλογές πάνω σε αυτά, εμφανίζεται και μία επιπλέον στήλη (η Permissions) με τα δικαιώματα τους (βλ. Εικόνα 11). Σε περίπτωση καταλόγου, ο πρώτος χαρακτήρας που εμφανίζεται στα δικαιώματα του είναι ο 'd' με τον οποίο δηλώνεται ότι πρόκειται για κατάλογο. Αντιθέτως, σε περίπτωση αρχείου ο πρώτος χαρακτήρας είναι ο '-' με τον οποίο δηλώνεται ότι πρόκειται για ένα κανονικό αρχείο. Υπάρχουν και κάποιοι άλλοι χαρακτήρες που χρησιμοποιούνται όπως ο 'l' που δηλώνει έναν συμβολικό σύνδεσμο, ο 'b' για ένα αρχείο τύπου block special, ο 's' για ένα socket, ο 'c' για ένα αρχείο τύπου character special, ο 'p' για ένα fifo pipe και τέλος ο 'u' για ένα αρχείο τύπου unknown.

3) *Εμφάνιση κάποιων επιπλέον χαρακτηριστικών των περιεχομένων του τρέχοντος καταλόγου, όπως το μέγεθος και την ημερομηνία τελευταίας τροποποίησης*

Μία ακόμη από τις διαθέσιμες επιλογές είναι και αυτή της εμφάνισης κάποιων λεπτομερειών των περιεχομένων ενός καταλόγου. Η προεπιλεγμένη ρύθμιση είναι η

μη εμφάνιση αυτών των λεπτομερειών. Ο χρήστης αν επιθυμεί την εμφάνισή τους, αρκεί να τσεκάρει το αντίστοιχο 'File details' checkbox (βλ. Εικόνα 11) και θα του παρουσιαστούν τα μεγέθη αλλά και η ημερομηνία τελευταίας τροποποίησης των αντικειμένων. Στη βάση του file manager, όπου εμφανίζεται ο συνολικός αριθμός των καταλόγων και αρχείων του τρέχοντος καταλόγου, θα εμφανιστεί και μια επιπλέον πληροφορία σχετικά με το συνολικό μέγεθος του τρέχοντος καταλόγου. Αν στη συνέχεια ο χρήστης θελήσει να κρύψει αυτές τις λεπτομέρειες, αρκεί να αποεπιλέξει το 'File details' checkbox.

- 4) *Επιλογή ταξινόμησης των περιεχομένων ενός καταλόγου κατά:*
- a. *Αλφαβητική σειρά*
 - b. *Μέγεθος*
 - c. *Ημερομηνία τελευταία τροποποίησης*
 - d. *Τύπο αρχείου*

Τα περιεχόμενα ενός καταλόγου είναι ταξινομημένα κατά αύξουσα αλφαβητική σειρά. Αυτή είναι η προεπιλεγμένη ρύθμιση, την οποία ο χρήστης μπορεί να αλλάξει. Όποια και να είναι η επιλογή του, πάντα ταξινομούνται οι κατάλογοι πρώτα και μετά τα αρχεία. Σε όλες τις περιπτώσεις έχουμε αύξουσα ταξινόμηση, π.χ από το μικρότερο μέγεθος στο μεγαλύτερο ή από την παλαιότερη ημερομηνία τροποποίησης στην πιο πρόσφατη.

Επιλέγοντας ταξινόμηση κατά αλφαβητική σειρά, τα αντικείμενα ταξινομούνται με βάση το όνομά τους, χωρίς να γίνεται διάκριση πεζών-κεφαλαίων, πρόκειται δηλαδή για case-sensitive ταξινόμηση. Αυτό σημαίνει πως κατά τη διάρκεια της ταξινόμησης, τα κεφαλαία γράμματα μετατρέπονται στα αντίστοιχα μικρά τους και μετά γίνεται η σύγκριση των ονομάτων των αντικειμένων και η ταξινόμησή τους κατά αλφαβητική σειρά. Έτσι, δεν εμφανίζονται πρώτα όλα τα ονόματα με τα κεφαλαία γράμματα και μετά τα ονόματα με τα μικρά, αλλά όλα μαζί. Για παράδειγμα, τρεις κατάλογοι με τα ονόματα pnuke, Protected και new θα ταξινομηθούν ως εξής: new, pnuke, Protected και όχι Protected, new, pnuke.

Επιλέγοντας ταξινόμηση με βάση τον τύπο του αρχείου, δημιουργούμε ομάδες αρχείων με βάση τον τύπο τους π.χ πρώτα τα τύπου bmp, μετά τα τύπου txt κ.τ.λ. Θα πρέπει να αναφέρουμε πως σε αυτήν την περίπτωση, οι ομάδες αρχείων δημιουργούνται αφού πρώτα έχει γίνει αύξουσα αλφαβητική ταξινόμηση των ονομάτων των τύπων των αρχείων. Έτσι, δεν υπάρχει περίπτωση να εμφανιστεί για παράδειγμα ένα zip αρχείο πριν από ένα css αρχείο, καθώς αλφαβητικά το css προηγείται του zip. Αυτό γίνεται απλά για να υπάρχει μια σειρά στην εμφάνιση των ομάδων των αρχείων ώστε να μην εμφανίζονται με διαφορετική σειρά κάθε φορά.

Θα πρέπει να τονίσουμε πως όταν έχουμε ταξινόμηση κατά τύπο, **ακολουθεί και δεύτερη ταξινόμηση**, κατά αλφαβητική σειρά αυτή τη φορά, στα αρχεία του ίδιου τύπου.








Αξίζει να σημειωθεί πως όταν θελήσουμε να χρησιμοποιήσουμε τον file manager, και αφού θα έχει εισαχθεί ο σωστός κωδικός, θα εμφανιστούν τα περιεχόμενα του αρχικού καταλόγου σε 3 στήλες, ταξινομημένα κατά αλφαβητική σειρά και χωρίς τις επιπλέον λεπτομέρειες (date και size), καθώς αυτές είναι οι προεπιλεγμένες ρυθμίσεις. Αν στη συνέχεια τροποποιηθούν κάποιες από αυτές τις ρυθμίσεις και θελήσουμε να μπούμε σε έναν άλλο κατάλογο, τότε τα περιεχόμενα του καταλόγου θα εμφανιστούν σύμφωνα με τις τρέχουσες ρυθμίσεις και όχι σύμφωνα με τις αρχικές προεπιλεγμένες ρυθμίσεις. Το ίδιο ισχύει και στην περίπτωση που είμαστε μέσα σε έναν υποκατάλογο και θελήσουμε να επιστρέψουμε στον προηγούμενο κατάλογο. Το σύστημα μας δηλαδή φροντίζει για την **διατήρηση των τρεχουσών ρυθμίσεων** κατά τη μετακίνηση μας ανάμεσα στους καταλόγους του συστήματος.

4.3.4 Επιπλέον δυνατότητες που παρέχει το σύστημα

Ακολουθεί η περιγραφή κάποιων επιπλέον δυνατοτήτων που παρέχει το σύστημα, που βοηθούν το χρήστη στην εκτέλεση των λειτουργιών που επιθυμεί και στην πλοήγηση μέσα στον file manager.

1) *Υπαρξη διαφορετικού εικονιδίου για κάθε τύπο αρχείων*

Κατά την εμφάνιση των αρχείων του τρέχοντος καταλόγου, καθένα τους συνοδεύεται από μια εικόνα που δηλώνει τον τύπο του αρχείου. Υπάρχουν επτά εικόνες που χρησιμοποιούνται για την περιγραφή των αρχείων και περιγράφονται στη συνέχεια:

- i. Για αρχεία τύπου html-htm-php-css-asp: 
- ii. Για αρχεία τύπου bmp-jpg-jpeg-gif-png: 
- iii. Για αρχεία τύπου txt: 
- iv. Για αρχεία τύπου pdf: 
- v. Για αρχεία τύπου ps-eps: 
- vi. Για αρχεία τύπου zip-rar-gz-tar: 
- vii. Για αρχεία τύπου Other: 

Στην περίπτωση ενός καταλόγου, η εικόνα που χρησιμοποιείται είναι η εξής: 

2) *Εμφάνιση του συνολικού αριθμού των υποκαταλόγων και των αρχείων του τρέχοντος καταλόγου*

Στη βάση του διαχειριστή αρχείων, υπάρχει ένα μικρό μενού όπου εμφανίζεται ο συνολικός αριθμός των αρχείων και των καταλόγων του τρέχοντος καταλόγου. Έτσι, δίνεται η δυνατότητα στον χρήστη να γνωρίζει σε κάθε χρονική στιγμή το πλήθος των αντικειμένων του καταλόγου στον οποίο βρίσκεται.

3) *Εμφάνιση διαφωτιστικών μηνυμάτων σε περίπτωση λάθους*

Σε καθεμία από τις προσφερόμενες λειτουργίες, υπάρχει η πιθανότητα λάθους χειρισμού από την πλευρά του χρήστη. Για παράδειγμα μπορεί να προσπαθήσει να δημιουργήσει ένα αρχείο που υπάρχει ήδη μέσα στον τρέχοντα κατάλογο. Προκειμένου να βοηθηθεί ο χρήστης και να χρησιμοποιήσει τον file manager στον καλύτερο δυνατό βαθμό, εμφανίζονται διαφωτιστικά μηνύματα που τον ενημερώνουν σχετικά με το λάθος του και τον βοηθούν να διορθώσει τις λανθασμένες ενέργειες του.

4) *Εμφάνιση του τρέχοντος καταλόγου στην κορυφή του file manager*

Στην κορυφή του file manager εμφανίζεται το μονοπάτι που δηλώνει σε ποιον κατάλογο βρίσκεται ο χρήστης σε κάθε χρονική στιγμή. Έτσι, του προσφέρεται η δυνατότητα να γνωρίζει τον τρέχοντα κατάλογο ούτως ώστε να διευκολύνεται στη μετακίνηση του μέσα στους διάφορους καταλόγους του file manager.

5) *Επιστροφή στον αρχικό κατάλογο*

Όταν ο χρήστης βρίσκεται μέσα σε έναν κατάλογο, μπορεί να επιστρέψει στον προηγούμενο κατάλογο ακολουθώντας τον BACK σύνδεσμο αλλά και να επιστρέψει στον κύριο κατάλογο, αν το επιθυμεί. Στην κορυφή του file manager, υπάρχει ένας σύνδεσμος, το Home link (βλ. Εικόνα 11). Ακολουθώντας τον, ο χρήστης μπορεί να επιστρέψει στον αρχικό κατάλογο. Έτσι, σε οποιοδήποτε κατάλογο και αν βρίσκεται, του παρέχεται η δυνατότητα απευθείας επιστροφής στον αρχικό. Ας σημειωθεί ότι κατά την επιστροφή στον αρχικό κατάλογο, διατηρούνται όλες οι τρέχουσες ρυθμίσεις του τελευταίου καταλόγου στον οποίο βρισκόταν ο χρήστης.

6) *Δυνατότητα 'Refresh'*

Δίπλα από την επιλογή επιστροφής στον αρχικό κατάλογο, υπάρχει η δυνατότητα ανανέωσης του περιεχομένου του file manager, η οποία επιτυγχάνεται με τη χρήση του 'Refresh' link.

4.4 Ιδιαιτερότητες υλοποίησης

Στην παράγραφο αυτή θα συζητηθούν κάποιες ιδιαιτερότητες της υλοποίησης μας οι οποίες προσέδωσαν αρκετά επιπλέον χαρακτηριστικά στον διαχειριστή αρχείων που δημιουργήθηκε. Μέσα στον κώδικα του αρχείου που υλοποιεί τον διαχειριστή υπάρχουν κάποιες μεταβλητές που παίζουν σημαντικό ρόλο στον καθορισμό των ιδιαιτεροτήτων του διαχειριστή. Η αλλαγή των τιμών τους, μπορεί να αλλάξει πολλές από τις δυνατότητες που προσφέρονται, κάποιες να τις ενισχύσει και άλλες να τις εξαλείψει εντελώς. Ακολουθεί μια αναλυτική περιγραφή αυτών των ιδιαιτεροτήτων και τονίζονται οι επιπλέον δυνατότητες που αυτές προσέφεραν στο διαχειριστή αρχείων που δημιουργήθηκε:

- Μέγιστο μέγεθος αρχείου και καταλόγου

Μέσα στον κώδικα του αρχείου που υλοποιεί τον file manager, υπάρχει η μεταβλητή `$UploadLimit` που καθορίζει το μέγιστο μέγεθος ενός αρχείου που μπορεί να φορτωθεί στον τρέχοντα κατάλογο. Δεν επιτρέπεται να φορτωθεί αρχείο μεγέθους μεγαλύτερου του `$UploadLimit`. Η τρέχουσα τιμή του είναι 524288 bytes.

Ακόμη, για τον καθορισμό του μέγιστου επιτρεπόμενου μεγέθους ενός καταλόγου χρησιμοποιείται η μεταβλητή `$DirsizeLimit`. Η τρέχουσα τιμή της είναι 128 Mbytes. Όποτε προσπαθήσουμε να φορτώσουμε ένα αρχείο στον τρέχοντα κατάλογο που θα τον αναγκάσει να ξεπεράσει το μέγιστο μέγεθος του, εμφανίζεται διαφωτιστικό μήνυμα και το αρχείο δεν προστίθεται τελικά.

Και οι δύο αυτές μεταβλητές μπορούν να αλλάξουν τιμή για να προσαρμοστούν στις ανάγκες της ιστοσελίδας στην οποία θα τοποθετηθεί ο file manager και όπου θα χρησιμοποιείται για την επεξεργασία των δεδομένων της.

- 'Κρυφά' αρχεία

Πρόκειται για μία ακόμη ιδιαιτερότητα του προγράμματος. Υπάρχει ένας πίνακας (ο `$HiddenFiles`) που περιέχει τα ονόματα των αρχείων και των καταλόγων που δεν θέλουμε να εμφανίζονται. Στον πίνακα αυτό τοποθετούνται συνήθως αρχεία που θέλουμε να προστατεύσουμε, όπως για παράδειγμα το `.htaccess` αρχείο ή το `.css` αρχείο του file manager μας. Μπορούμε να προσθέσουμε νέα ονόματα αλλά και να αφαιρέσουμε κάποια από τα ήδη υπάρχοντα αν θέλουμε αυτά να πάντων να είναι κρυφά. Έτσι, κατά την εμφάνιση των περιεχομένων του τρέχοντος καταλόγου τα hidden αρχεία δεν θα εμφανίζονται.

- Δημιουργία καταλόγου

Υπάρχει μια μεταβλητή, η `$MkdirAllowed`, που καθορίζει αν ο file manager επιτρέπει τη δημιουργία καταλόγου. Αρχικά, αυτή η μεταβλητή έχει τιμή 1 που σημαίνει ότι επιτρέπεται η δημιουργία νέου καταλόγου. Όποτε θέλουμε να απαγορεύσουμε αυτή τη λειτουργία αρκεί να κάνουμε την εξής απλή κίνηση: να θέσουμε τη μεταβλητή σε 0.

- Κατάλογοι που δεν επιτρέπουν τη δημιουργία νέου καταλόγου

Υπάρχει ένας πίνακας που περιέχει όλους τους καταλόγους που δεν επιτρέπουν τη δημιουργία νέου καταλόγου σε αυτούς. Για παράδειγμα, μια τιμή του πίνακα θα μπορούσε να ήταν η εξής: `$NoCreateDirs = array("docs/")`. Βέβαια, στον πίνακα αυτό μπορούμε να τοποθετήσουμε όσους καταλόγους θέλουμε να έχουν αυτή την ιδιαιτερότητα. Έτσι, όταν ο χρήστης εισέλθει στον κατάλογο docs/, θα διαπιστώσει πως το 'Folder' button στο Create μενού λείπει και έχει αντικατασταθεί από το εξής ενημερωτικό μήνυμα: [Directory creation not permitted here].

- Κατάλογοι που δεν επιτρέπουν την φόρτωση, αντιγραφή ή μετακίνηση αρχείων στο εσωτερικό τους

Παρόμοια, υπάρχουν κατάλογοι που δεν επιτρέπουν τη φόρτωση αρχείων σε αυτούς μέσω της διαδικασίας upload. Για παράδειγμα, μια τιμή του πίνακα θα μπορούσε να ήταν η εξής: `$NoUploadDirs = array("docs/")`. Βέβαια, στον πίνακα αυτό μπορούμε να τοποθετήσουμε όσους καταλόγους θέλουμε να έχουν αυτή την ιδιαιτερότητα. Έτσι, όταν ο χρήστης εισέλθει στον κατάλογο docs/, θα διαπιστώσει πως το Upload μενού θα λείπει.

Επιπλέον, οι κατάλογοι αυτοί δεν επιτρέπουν ούτε την αντιγραφή ή τη μετακίνηση αρχείων σε αυτούς. Έτσι, δεν εμφανίζονται καν στη λίστα των καταλόγων προορισμού κατά τη διαδικασία αντιγραφής ή μετακίνησης των αντικειμένων ενός καταλόγου.

- Δημιουργία νέου αρχείου

Υπάρχει μια μεταβλητή, η `$MkfileAllowed`, που καθορίζει αν ο file manager επιτρέπει τη δημιουργία νέου αρχείου. Αρχικά, αυτή η μεταβλητή έχει τιμή 1 που σημαίνει ότι επιτρέπεται η δημιουργία νέου αρχείου. Όποτε θέλουμε να

απαγορεύσουμε αυτή τη λειτουργία αρκεί να κάνουμε την εξής απλή κίνηση: να θέσουμε τη μεταβλητή σε 0, οπότε και εμφανίζεται το εξής μήνυμα: [New File Creation not permitted here].

- Επεκτάσεις των αρχείων που επιτρέπεται να τροποποιηθούν

Δεν είναι δυνατή η τροποποίηση όλων των αρχείων, υπάρχουν συγκεκριμένοι τύποι αρχείων που επιτρέπεται να τροποποιηθούν. Αυτοί καθορίζονται στον πίνακα `$EditExtensions` ο οποίος αρχικά έχει την τιμή: `$EditExtensions = array("htm", "html", "txt", "php", "css", "c", "h")`. Αν θελήσει να χρησιμοποιήσει περισσότερες επεκτάσεις αρκεί να τις προσθέσει στον πίνακα. Αλλά και αν δεν θελήσει να υπάρχουν κάποιες από τις ήδη διαθέσιμες τιμές μπορεί να τις αφαιρέσει από τον πίνακα. Έτσι, για όλα τα αρχεία που ο τύπος τους ανήκει στον πίνακα `$EditExtensions`, εμφανίζεται το edit εικονίδιο δίπλα τους ενώ για όλα τα άλλα δεν είναι δυνατή η εμφάνιση του.

- Τα 'BLOCKED' αρχεία

Υπάρχει ο `$ModifyBlock` πίνακας που περιέχει όλα τα 'BLOCKED' αρχεία και τους καταλόγους. Πρόκειται για αρχεία που δεν επιτρέπεται να προστεθούν (upload) ή να αντιγραφούν σε έναν κατάλογο, να μετονομαστούν και να διαγραφούν. Ο πίνακας μπορεί να αλλάξει για να περιέχει όλα τα αρχεία (ή τους καταλόγους) που επιθυμούμε κάθε φορά.

- Χαρακτήρες που αφαιρούνται από τα ονόματα των αρχείων

Όταν δημιουργούμε ένα αρχείο, το σύστημα επεξεργάζεται το όνομα που δίνει ο χρήστης για το νέο αρχείο. Υπάρχουν κάποιοι χαρακτήρες σε ένα όνομα που θέλουμε να τους αφαιρούμε από αυτό, αντικαθιστώντας τους με τον χαρακτήρα `"`. Οι χαρακτήρες αυτοί ανήκουν στον πίνακα `$snr` που έχει τις τιμές: `$snr=array("%", "'", "+", "\\", "/", "#", ".", "!", '"', '|', '?', '*', '~');` Για παράδειγμα, σε ένα δοθέν όνομα, όπου συναντάται κάποιος από τους παραπάνω χαρακτήρες, αντικαθιστάται από τον χαρακτήρα `"`, δηλαδή αφαιρείται από το όνομα. Το ίδιο συμβαίνει και κατά τη δημιουργία νέου καταλόγου όπως και κατά τη φόρτωση νέου αρχείου.

- Επιλογές ταξινόμησης

Υπάρχει ο πίνακας `$arr` που περιέχει όλους τους δυνατούς τρόπους ταξινόμησης. Οι τρέχουσες τιμές του είναι `$arr= array("Name", "Size", "Date", "Type");`. Ο πίνακας μπορεί να αλλάξει για να περιέχει και άλλους τρόπους ταξινόμησης. Η προεπιλεγμένη ρύθμιση ταξινόμησης είναι η 'Name' και κρατείται στη μεταβλητή `$defaultSortby`. Η τιμή της μπορεί να αλλάξει προκειμένου να καθορίσουμε την προεπιλεγμένη ρύθμιση ταξινόμησης που επιθυμούμε.

- Στήλες εμφάνισης

Υπάρχει ο `$arr` πίνακας που περιέχει τις στήλες στις οποίες επιτρέπεται να εμφανίζονται τα περιεχόμενα ενός καταλόγου. Οι τρέχουσες τιμές του είναι: `$arr= array(1,2,3,4,5,6,7,8,9,10)`, καθώς θέλουμε εμφάνιση από 1 έως 10 στήλες. Τα περιεχόμενα του πίνακα μπορούν να αλλάξουν ανάλογα με τις ανάγκες μας κάθε φορά. Ο αρχικός αριθμός εμφάνισης στηλών είναι 3 και κρατείται στη μεταβλητή `$defaultColumns`. Η τιμή της μπορεί να αλλάξει προκειμένου να καθορίσουμε το νέο προεπιλεγμένο αριθμό στηλών εμφάνισης.

- Εμφάνιση λεπτομερειών των αρχείων

Υπάρχει μια μεταβλητή που καθορίζει αν θα εμφανίζονται οι λεπτομέρειες των αρχείων. Η αρχική τιμή της μεταβλητής είναι `FALSE` (`$defaultDetails=FALSE`) καθώς την πρώτη φορά που μπαίνουμε στον `root` κατάλογο δεν επιθυμούμε να εμφανίζονται οι λεπτομέρειες των αρχείων. Η τιμή της μπορεί να αλλάξει και να μετατραπεί σε `TRUE`, αν επιθυμούμε αρχικά να εμφανίζονται οι λεπτομέρειες των αρχείων δηλαδή το μέγεθος και η ημερομηνία τελευταίας τροποποίησης τους αλλά και το συνολικό μέγεθος του τρέχοντος καταλόγου.

- Προσθήκη του "copy_" μπροστά από τα ονόματα των αρχείων και καταλόγων

Στην περίπτωση της αντιγραφής ή μετακίνησης αρχείων, αν υπάρχει ήδη κάποιο αρχείο με το ίδιο όνομα (αλλά και τον ίδιο τύπο) στον κατάλογο προορισμού, στην αρχή του ονόματος του προς αντιγραφή αρχείου προστίθεται το αλφαριθμητικό "copy_" για να αποφύγουμε την αντικατάσταση του παλιού αρχείου από αυτό. Το ίδιο συμβαίνει και κατά τη μετακίνηση των αρχείων. Παρόμοιες ρυθμίσεις έχουν γίνει και για την αντιγραφή/μετακίνηση των καταλόγων. Με αυτό τον τρόπο, προσφέρεται και η δυνατότητα αντιγραφή ενός αρχείου μέσα στον ίδιο κατάλογο που ανήκει.

Συνοψίζοντας, θα μπορούσαμε να πούμε πως όλες οι μεταβλητές που αναφέρθηκαν και που περιγράφουν τις ιδιαιτερότητες της υλοποίησης μας, μπορούν να αλλάξουν τιμές μελλοντικά για να τροποποιήσουν τις ρυθμίσεις του file manager και να του προσδώσουν νέες δυνατότητες.

4.5 Διάγραμμα λειτουργίας

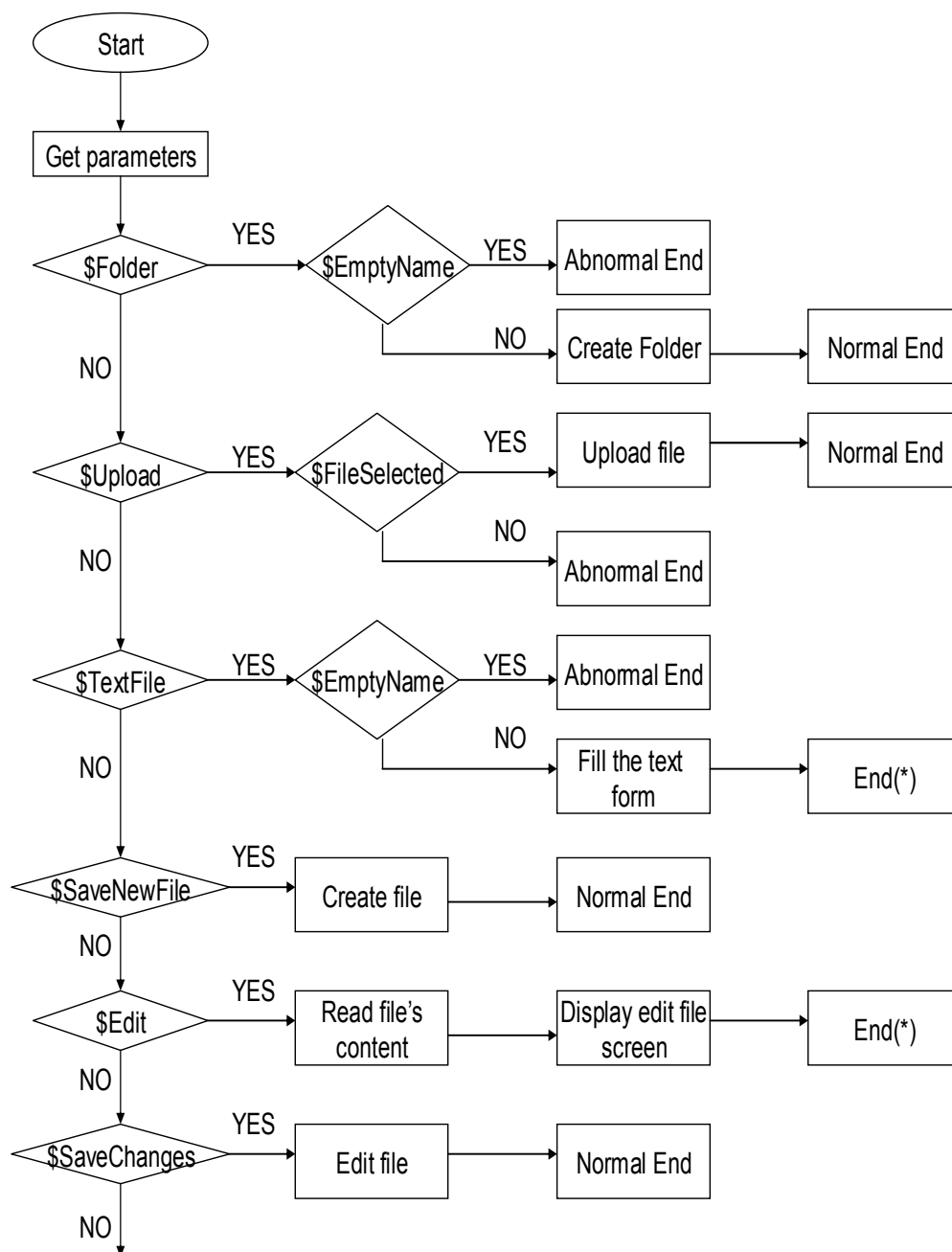
Όταν εισερχόμαστε σε έναν κατάλογο, αντικρίζουμε τη λίστα (ή τον πίνακα) των περιεχομένων του μαζί με τις λειτουργίες που μπορούμε να εκτελέσουμε πάνω σε αυτά. Σε κάποιες από τις λειτουργίες, όπως για παράδειγμα σε αυτή της δημιουργίας νέου καταλόγου ή της φόρτωσης αρχείου ακολουθείται η εξής διαδικασία: μετά από το πάτημα του κατάλληλου button από τον χρήστη, καλούνται οι κατάλληλες συναρτήσεις στον κώδικα του file manager για την εκτέλεση της λειτουργίας που επέλεξε ο χρήστης. Δεν μεταφερόμαστε σε κάποια άλλη σελίδα αλλά παραμένουμε στην τρέχουσα που περιέχει τη λίστα των εργασιών. Απλά, στο τέλος της λίστας εμφανίζεται κατάλληλο μήνυμα στον χρήστη που τον ενημερώνει σχετικά με το αποτέλεσμα της λειτουργίας που είχε επιλέξει.

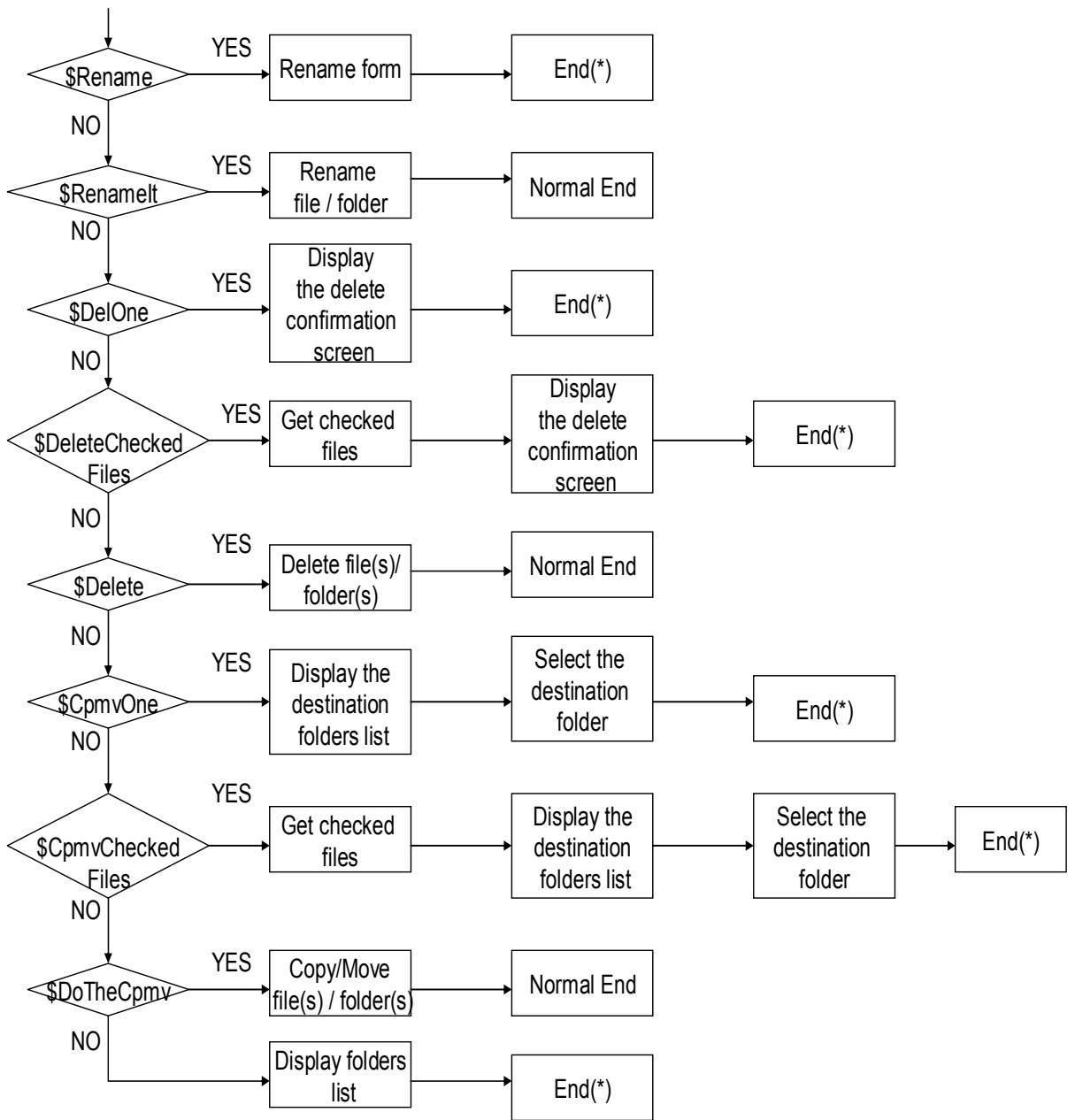
Σε κάποιες άλλες λειτουργίες, όπως αυτή της δημιουργίας νέου αρχείου, της μετονομασίας, της τροποποίησης, της διαγραφής και της αντιγραφής/μετακίνησης ακολουθείται ελαφρώς διαφορετική διαδικασία. Όταν ο χρήστης επιλέξει κάποια λειτουργία πατώντας το κατάλληλο button ή ακολουθώντας τον σύνδεσμο της λειτουργίας, δεν θα παραμείνει στην τρέχουσα σελίδα με τη λίστα των εργασιών και το μενού του file manager, αλλά θα μεταφερθεί σε μια άλλη σελίδα για να συνεχίσει εκεί την εκτέλεση της λειτουργίας του. Για παράδειγμα, κατά τη διαδικασία διαγραφής, θα αντικρίσει ένα μενού επιβεβαίωσης διαγραφής, κατά τη διαδικασία τροποποίησης θα αντικρίσει μία φόρμα με το περιεχόμενο του αρχείου που θέλει να τροποποιήσει και κατά τη διαδικασία αντιγραφής θα καλεστεί να επιλέξει τον κατάλογο προορισμού ψάχνοντας μέσα στη λίστα των διαθέσιμων καταλόγων. Έπειτα, με το πάτημα των κατάλληλων κουμπιών (π.χ Save button) μπορεί να ολοκληρώσει τη λειτουργία του, οπότε και θα επιστρέψει πάλι στην προηγούμενη σελίδα με τη λίστα των αρχείων, όπου και θα του εμφανιστεί κατάλληλο ενημερωτικό μήνυμα σχετικά με το αποτέλεσμα της λειτουργίας του.

Η διαφορά δηλαδή των δύο ομάδων λειτουργιών που περιγράφηκαν είναι η εξής: Στις πρώτες, με την επιλογή εκτέλεσης μιας λειτουργίας από τον χρήστη, παραμένουμε στην τρέχουσα σελίδα με την διαφορά ότι προστίθεται και ένα μήνυμα στο τέλος της. Ενώ στις υπόλοιπες λειτουργίες, μεταφερόμαστε σε μία άλλη σελίδα για να ολοκληρώσουμε την λειτουργία που θέλουμε να εκτελέσουμε και μετά

επιστρέφουμε στην αρχική σελίδα, όπου έχει προστεθεί ένα μήνυμα στο τέλος της. Και στις δύο περιπτώσεις 'τρέχει' πάντα το ίδιο αρχείο, αυτό που υλοποιεί τον file manager. Ο file manager καλεί συνεχώς τον εαυτό του, με διαφορετικές παραμέτρους κάθε φορά, ανάλογα με τη λειτουργία που έχει επιλεγεί.

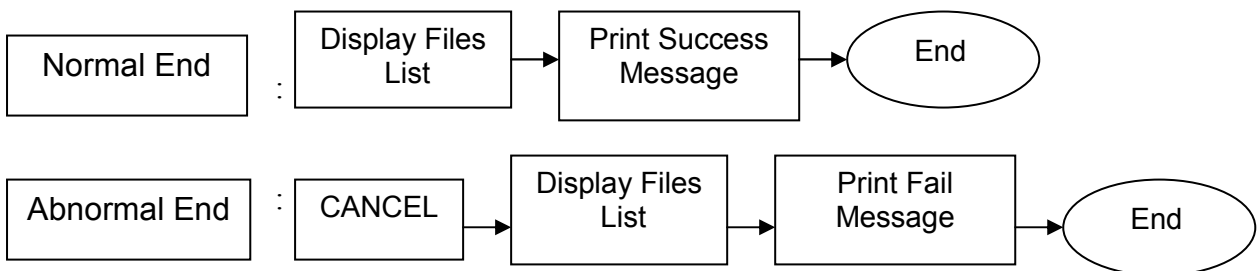
Στη συνέχεια παρατίθεται ένα διάγραμμα της λειτουργίας του συστήματος. Περιγράφεται ο αλγόριθμος που ακολουθείται, ανάλογα με τη λειτουργία που επιλέγει ο χρήστης κάθε φορά.





Εικόνα 12: Διάγραμμα λειτουργίας του διαχειριστή αρχείων

Ακολουθεί η επεξήγηση κάποιων τμημάτων του διαγράμματος:



End(*)

: Δεν ολοκληρώνεται η λειτουργία αλλά καλείται πάλι το αρχείο που υλοποιεί τον file manager με τις κατάλληλες παραμέτρους αυτή τη φορά, ούτως ώστε να κληθεί η κατάλληλη συνάρτηση για να ολοκληρωθεί η λειτουργία που έχει επιλέξει ο χρήστης. Στην ουσία, επιστρέφουμε πάλι στο start του διαγράμματος.

4.6 Σχετικά υπάρχοντα συστήματα διαχείρισης αρχείων

Αρκετές είναι οι εφαρμογές τύπου 'file manager' που έχουν δημιουργηθεί με τη χρήση της γλώσσα PHP. Πολλές από αυτές είναι απλοί διαχειριστές αρχείων [18,19,20,21] που λειτουργούν σαν FTP εφαρμογές, με σκοπό την επεξεργασία των δεδομένων ενός ή περισσότερων καταλόγων. Συνήθως αποτελούν ένα πακέτο αρχείων που αν τοποθετηθούν στον κατάλληλο αρχικό (root) κατάλογο, επιτρέπουν την πρόσβαση και την επεξεργασία των περιεχομένων αυτού του καταλόγου και όλων των εσωτερικών υποκαταλόγων του μέσω του web. Συνήθως, οι περισσότεροι file managers τοποθετούνται σε κάποιον κατάλογο μιας ιστοσελίδας και δίνουν στους χρήστες τη δυνατότητα να διαχειρίζονται τα δεδομένα της ιστοσελίδας, με τη βοήθεια του script που υλοποιεί τον file manager.

Υπάρχουν κάποιες βασικές λειτουργίες που τις υλοποιούν οι περισσότεροι file managers. Και αυτές δεν είναι άλλες από τις: view /edit/ rename file και delete/copy /move/upload files. Υπάρχουν βέβαια και file managers που έχουν αρκετές επιπλέον δυνατότητες, όπως τις δυνατότητες

- ✓ δημιουργίας νέου αρχείου ή καταλόγου
- ✓ ταξινόμησης με βάση το μέγεθος, τον τύπο ή την ημερομηνία τελευταίας τροποποίησης
- ✓ μετακίνησης ή αντιγραφής αρχείου μέσα στο ίδιο τον κατάλογο που ανήκει
- ✓ εμφάνισης και αλλαγής των δικαιωμάτων ενός αρχείου ή καταλόγου

Στη συνέχεια, παραθέτουμε μια παρουσίαση κάποιων από τα υπάρχοντα συστήματα διαχείρισης αρχείων και προχωράμε σε αναλυτική σύγκριση τους με τον δικό μας διαχειριστή αρχείων.

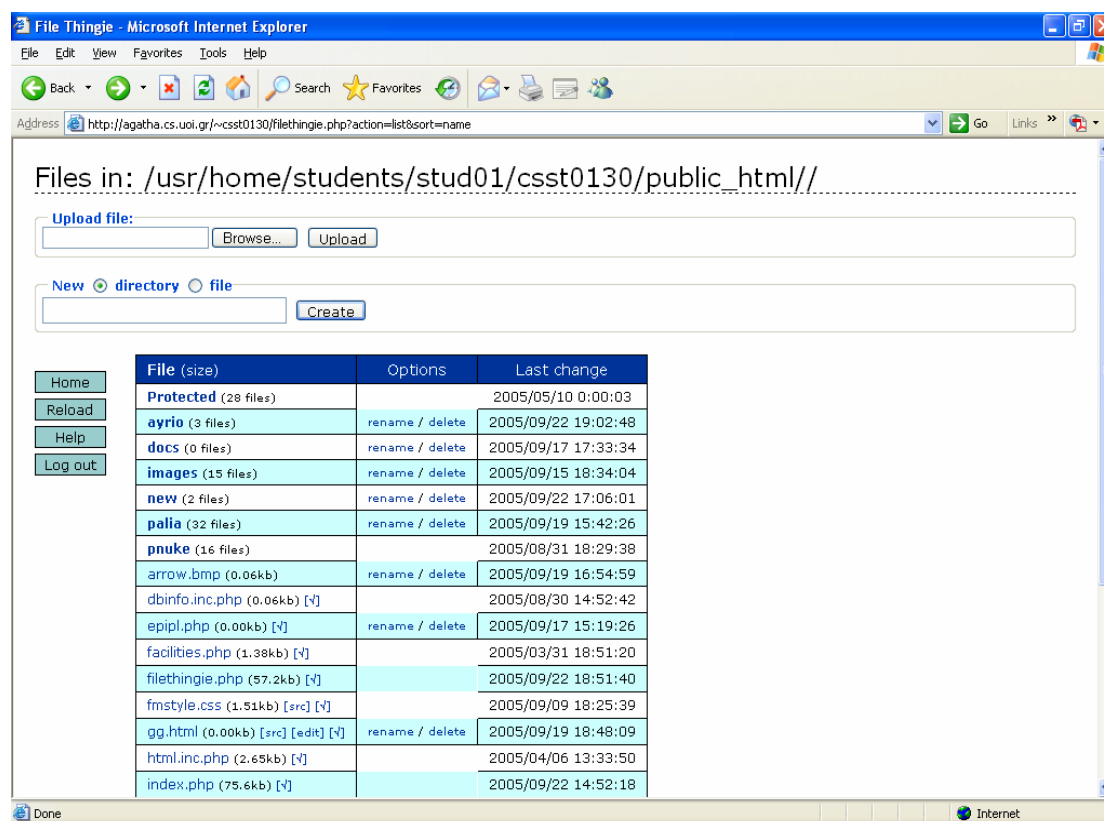
4.6.1 Αναλυτική σύγκριση

Θα πρέπει να σημειωθεί πως σε κάθε περίπτωση, ο διαχειριστής αρχείων που δημιουργείται, καλείται να εκπληρώσει τις προσδοκίες του χρήστη του. Οπότε μπορεί να χρειαστεί να έχει κάποιες μόνο από τις δυνατότητες που αναφέρθηκαν και όχι απαραίτητα όλες. Υπάρχουν πολλές γλώσσες προγραμματισμού που ασχολούνται με τη δημιουργία τέτοιων συστημάτων όπως η PHP, CGI, PERL, JAVA, ASP και άλλες

αλλά θα μπορούσαμε να πούμε πως με την PHP και την JAVA έχουν δημιουργηθεί οι περισσότεροι διαχειριστές αρχείων. Ας σημειωθεί, ότι τα συστήματα που θα περιγραφούν έχουν δημιουργηθεί με την γλώσσα δημιουργίας του δικού μας συστήματος, την PHP.

- **FILE THINGIE [22]**

Πρόκειται για ένα απλό file manager που δημιουργήθηκε το 2003 και μπορεί να κάνει ότι κάνουν και οι περισσότεροι διαχειριστές αρχείων. Είναι αρκετά δημοφιλής στους χρήστες που θέλουν απλά να βλέπουν τα αρχεία ενός καταλόγου και να εκτελούν κάποιες βασικές λειτουργίες πάνω σε αυτά. Εμφανίζει τα περιεχόμενα του καταλόγου στον οποίο τοποθετείται σε μορφή κατακόρυφης λίστας με κάποιες επιπλέον πληροφορίες για το καθένα όπως το μέγεθος, την ημερομηνία τελευταίας τροποποίησης αλλά και τον αριθμό των αρχείων που περιέχει (σε περίπτωση που είναι κατάλογος). Ακόμη, δίνει στο χρήστη τη δυνατότητα δημιουργίας νέου καταλόγου ή αρχείου καθώς και φόρτωσης νέου αρχείου. Τέλος, επιτρέπει τη μετονομασία, την τροποποίηση αλλά και τη διαγραφή αρχείων και καταλόγων. Στη συνέχεια παρατίθεται ένα παράδειγμα χρήσης του file thingie.



Εικόνα 13: Παράδειγμα χρήσης του File Thingie

Σύγκριση:

Παρατηρούμε ότι ο file thingie δεν παρέχει τίποτα επιπλέον από όσα παρέχει ο δικός μας file manager. Πολλές λειτουργίες που υπάρχουν στο δικό μας σύστημα, εδώ δεν είναι δυνατές όπως:

1. Ομαδική διαγραφή αρχείων και καταλόγων: Παρατηρούμε πως είναι διαθέσιμη μόνο η ατομική διαγραφή ενός αρχείου ή καταλόγου και όχι η ομαδική όπως στο δικό μας σύστημα.
2. Αντιγραφή / Μετακίνηση αρχείων και καταλόγων: Δεν υπάρχει καν μια τέτοια επιλογή διαθέσιμη σε αντίθεση με το σύστημα μας όπου είναι δυνατή η ατομική αλλά και η ομαδική αντιγραφή/μετακίνηση.
3. Διαγραφή καταλόγου: Παρατηρούμε πως ενώ μπορούμε να διαγράψουμε αρχεία, δεν μπορούμε να διαγράψουμε έναν κατάλογο, εκτός και αν αυτός είναι άδειος. Κατάλογοι με πάνω από ένα αρχεία δεν μπορούν να διαγραφούν. Θα πρέπει πρώτα να διαγραφούν όλα τα αρχεία τους και μετά οι ίδιοι. Κάτι το οποίο δεν συμβαίνει στο δικό μας file manager καθώς παρέχει τη δυνατότητα διαγραφής ενός καταλόγου με ταυτόχρονη διαγραφή όλων των δεδομένων του.
4. Αλλαγή τύπου ενός αρχείου: Ενώ μπορεί να μετονομαστεί ένα αρχείο, δεν επιτρέπεται να αλλαχτεί και η επέκτασή του και άρα και ο τύπος του.
5. Ταξινόμηση κατά τύπο αρχείου: Ενώ είναι δυνατή η αύξουσα ταξινόμηση κατά όνομα, μέγεθος και ημερομηνία τελευταίας τροποποίησης, δεν παρέχεται η δυνατότητα ταξινόμησης κατά τύπο αρχείου.
6. Δυνατότητα τροποποίησης για περιορισμένο τύπο αρχείων: Επιτρέπει να τροποποιηθούν μόνο τα αρχεία με επεκτάσεις ".txt", ".php", ".html", ενώ στο δικό μας file manager επιτρέπεται η τροποποίηση των αρχείων με επεκτάσεις : "htm", "html", "txt", "php", "css", "c", "h".
7. Εμφάνιση των αρχείων σε στήλες: Τα περιεχόμενα ενός καταλόγου εμφανίζονται μόνο σε μορφή κατακόρυφης λίστας, το ένα κάτω από το άλλο. Αυτός ο τρόπος εμφάνισης είναι λειτουργικός στην περίπτωση που έχουμε λίγα αρχεία οπότε και μπορούμε εύκολα να τα διαχειριστούμε όλα σε μορφή λίστας. Σε περίπτωση περισσότερων αρχείων όμως, η λίστα γίνεται αρκετά μεγάλη και άρα μη λειτουργική. Σε αυτή τη περίπτωση μας συμφέρει η επιλογή εμφάνισης των αρχείων σε μορφή πίνακα, δυνατότητα που δεν την προσφέρει ο File Thingie σε αντίθεση με τον δικό μας. Δεν υπάρχει λοιπόν η δυνατότητα εμφάνισης των αρχείων σε περισσότερες από μία στήλες πράγμα που δυσκολεύει τη διαχείριση τους. Αν εκτός από τη δυνατότητα εμφάνισης σε λίστα υπήρχε και η δυνατότητα εμφάνισης σε στήλες, θα χωρούσαν

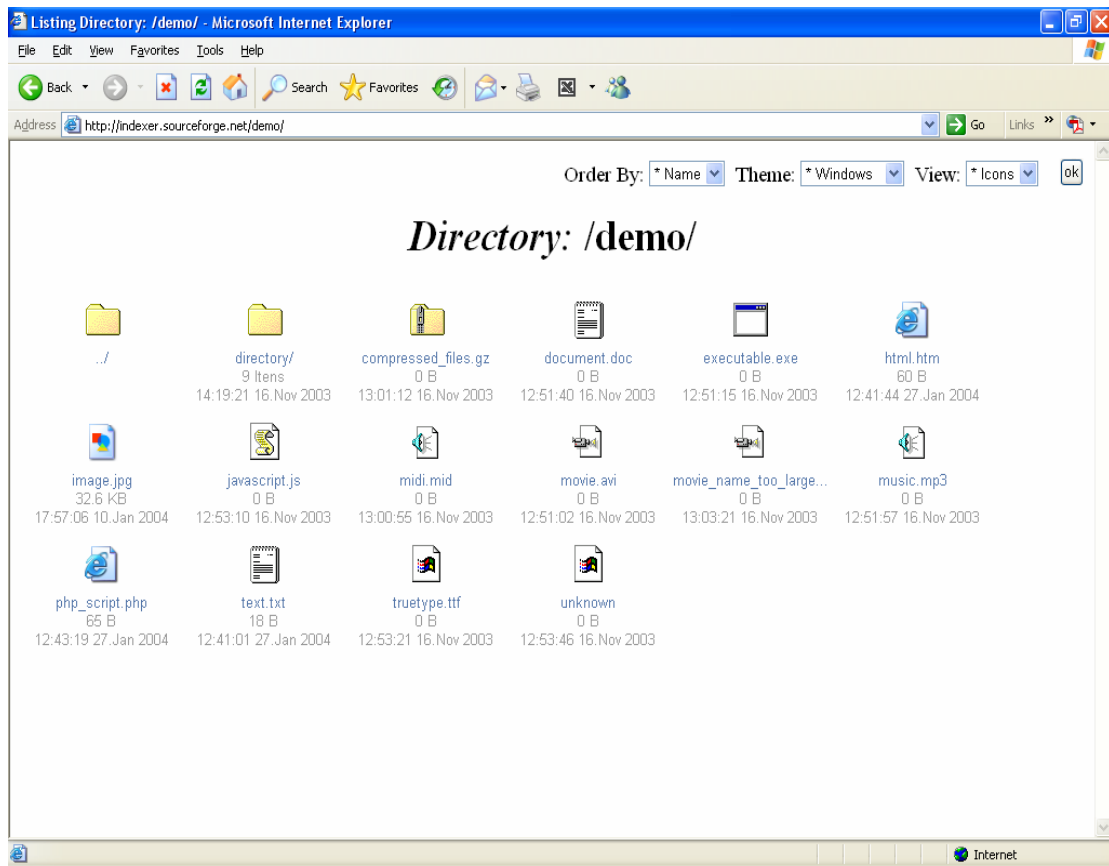
περισσότερα αρχεία σε κάθε γραμμή του πίνακα πράγμα που θα έδινε στο χρήστη τη δυνατότητα εξοικονόμησης χώρου.

8. Εμφάνιση /Απόκρυψη των λεπτομερειών των αρχείων: Για κάποια στοιχεία των αρχείων, όπως το μέγεθος ή την ημερομηνία δημιουργίας, δεν υπάρχει η δυνατότητα απόκρυψής τους ούτως ώστε να εμφανίζονται μόνο αν το επιθυμεί ο χρήστης. Στον file manager μας, η απόκρυψη τους σε συνδυασμό με την επιλογή εμφάνισης σε μεγάλο αριθμό στηλών, έχει σαν αποτέλεσμα την εξοικονόμηση αρκετού χώρου και την εμφάνιση των αρχείων σε λιγότερες γραμμές από ότι αν χρησιμοποιούσαμε εμφάνιση σε μορφή λίστας όπως στον file thingie.
9. Εμφάνιση των δικαιωμάτων: Ο file manager μας επιτρέπει την εμφάνιση των δικαιωμάτων των περιεχομένων του τρέχοντος καταλόγου, σε περίπτωση που ο χρήστης επιλέξει εμφάνιση σε κατακόρυφη λίστα. Η επιλογή αυτή δεν είναι διαθέσιμη στον file thingie.

- **PHP Directory Indexer [23]**

Ο PHP Directory Indexer είναι ένας PHP file manager που επιτρέπει την επιλογή της εμφάνισης των αρχείων ενός καταλόγου σε μορφή πίνακα με 6 στήλες ή σε μορφή λίστας όπου τα αρχεία συνοδεύονται από την εμφάνιση κάποιων επιπλέον στοιχείων για αυτά, όπως τον τύπο του αρχείου. Πρώτα εμφανίζονται οι κατάλογοι και μετά τα αρχεία. Επιτρέπεται η ταξινόμηση των αρχείων κατά αλφαβητική σειρά, κατά ημερομηνία τελευταίας τροποποίησης και κατά μέγεθος. Η προεπιλεγμένη ρύθμιση είναι η ταξινόμηση κατά ημερομηνία με τα πιο πρόσφατα αρχεία να εμφανίζονται τελευταία. Για κάθε κατάλογο, εμφανίζεται ο αριθμός των αρχείων (ή καταλόγων) που περιέχει και η ημερομηνία τελευταίας τροποποίησης του. Για κάθε αρχείο, εμφανίζεται το μέγεθός του σε KBytes μαζί με την ημερομηνία τελευταίας τροποποίησης του. Όταν μεταπηδούμε από εμφάνιση σε μορφή παραθύρου σε εμφάνιση σε μορφή πίνακα, εμφανίζεται επιπλέον και ο τύπος των αρχείων. Ακόμη, για κάθε τύπο αρχείου υπάρχει και διαφορετική εικόνα που να το περιγράφει, η οποία εμφανίζεται πάνω από το όνομά του αρχείου. Τέλος, υπάρχει η δυνατότητα επιλογής του στυλ εμφάνισης των εικόνων που περιγράφουν τα αρχεία και τους καταλόγους. Προσφέρονται πέντε δυνατές επιλογές: Windows, AquaFusion, Crystal, Conqueror, Simple. Ανάλογα με την επιλογή του χρήστη, εμφανίζεται και διαφορετική ομάδα εικόνων για τα αρχεία και τους καταλόγους.

Ακολουθεί ένα παράδειγμα εμφάνισης των περιεχομένων ενός καταλόγου με χρήση του PHP Directory Indexer.



Εικόνα 14: Παράδειγμα χρήσης του PHP Directory Indexer

Σύγκριση:

Παρατηρούμε πως ο PHP Directory Indexer έχει αρκετά κοινά σημεία με τον δικό μας file manager, κυρίως ως προς τον τρόπο εμφάνισης των αρχείων. Τα κοινά σημεία των δύο συστημάτων απαριθμούνται παρακάτω:

- 1) Δυνατότητα μεταπήδησης από μορφή πίνακα σε μορφή λίστας
- 2) Δυνατότητα επιλογής ταξινόμησης κατά όνομα, μέγεθος και ημερομηνία.
- 3) Ύπαρξη διαφορετικής εικόνας ανάλογα με τον τύπο του αρχείου.
- 4) Εμφάνιση ονόματος, μεγέθους και ημερομηνίας τελευταίας τροποποίησης των αρχείων

Υπάρχουν όμως και πολλά πράγματα που ενώ τα παρέχει ο file manager μας, δεν τα παρέχει ο PHP Directory Indexer. Οι διαφορές τους περιγράφονται παρακάτω:

1. Η βασική διαφορά των δύο file managers όμως είναι πως με τον PHP Directory Indexer το μόνο που επιτρέπεται στον χρήστη είναι να βλέπει τα αρχεία και να ρυθμίζει τον τρόπο με τον οποίο αυτά θα εμφανίζονται. Δεν του δίνει τη δυνατότητα να εκτελέσει λειτουργίες πάνω σε αυτά.

2. Ο PHP Directory Indexer δεν παρέχει τη δυνατότητα επιλογής του αριθμού των στηλών στις οποίες θα εμφανίζονται τα αρχεία. Στον file manager μας, ο χρήστης μπορεί να επιλέξει έναν αριθμό από 1-10 και έτσι όταν έχουμε εμφάνιση σε μορφή πίνακα, σε κάθε γραμμή του πίνακα θα εμφανίζονται τόσα αρχεία όσες είναι και οι στήλες που επέλεξε ο χρήστης.
3. Ο file manager μας δεν παρέχει τη δυνατότητα επιλογής του στυλ εμφάνισης των εικονιδίων των αρχείων.
4. Ο PHP Directory Indexer δεν επιτρέπει την ταξινόμηση κατά τύπο αρχείου.
5. Ο PHP Directory Indexer δεν επιτρέπει την εμφάνιση /απόκρυψη κάποιων στοιχείων των αρχείων. Όλα τα στοιχεία τους εμφανίζονται συνεχώς, χωρίς να υπάρχει επιλογή απόκρυψής τους.
6. Ο PHP Directory Indexer δεν επιτρέπει την εμφάνιση των δικαιωμάτων των περιεχομένων του τρέχοντος καταλόγου.

Συμπερασματικά, θα μπορούσαμε να πούμε πως ως προς τον τρόπο εμφάνισης των περιεχομένων ενός καταλόγου, που είναι και το κύριο αντικείμενο του PHP Directory Indexer, ο file manager μας ανταποκρίνεται εξίσου καλά όσο και ο PHP Directory Indexer ενώ παρέχει και επιπλέον δυνατότητες όπως τις λειτουργίες Delete-Copy/Move-New file-New folder και άλλες.

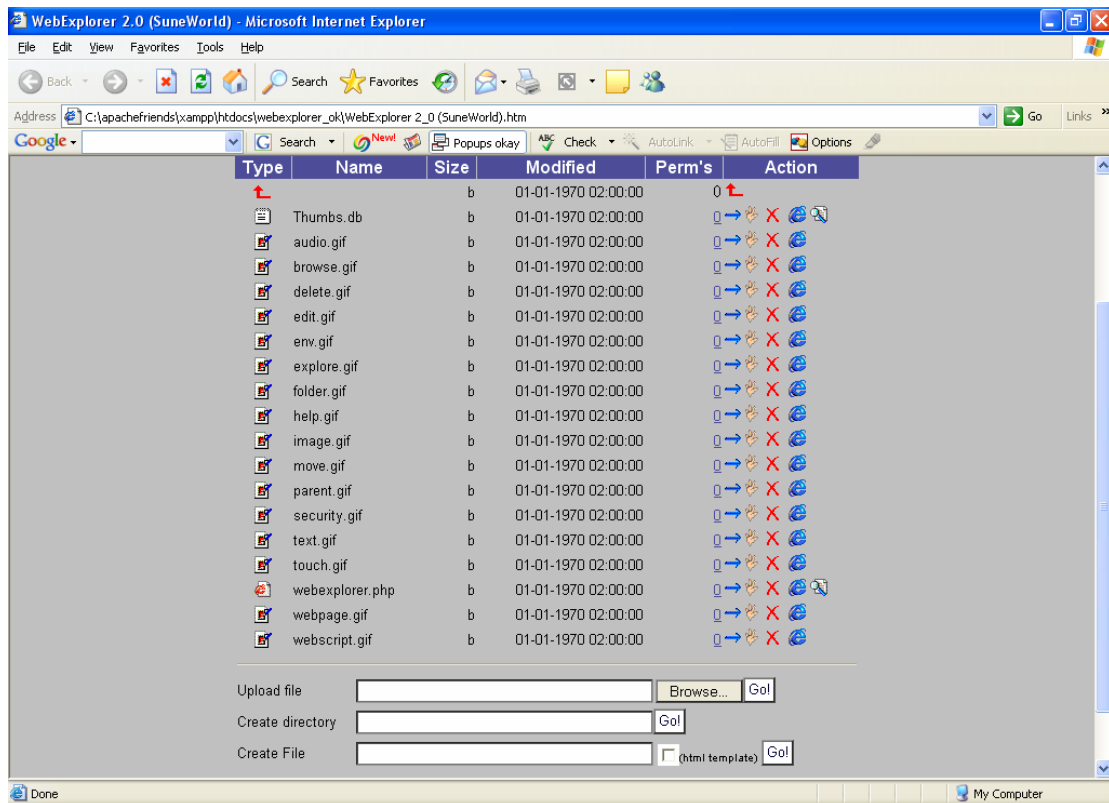
- **WebExplorer [21,24]**

Ο WebExplorer είναι ένας Php web file manager που παρέχει τις εξής δυνατότητες:

- 1) Δημιουργία νέου αρχείου και καταλόγου
- 2) Προσθήκη ενός νέου αρχείου μέσω της διαδικασίας upload.
- 3) Θέαση των περιεχομένων ενός αρχείου
- 4) Τροποποίηση
- 5) Μετονομασία
- 6) Αντιγραφή
- 7) Μετακίνηση

Πρόκειται για έναν αρκετά καλό διαχειριστή αρχείων που επιτρέπει στον χρήστη να διαχειρίζεται τα αρχεία με έναν αρκετά αποδοτικό τρόπο και να εκτελεί σε αυτά τις βασικές λειτουργίες.

Ακολουθεί ένα παράδειγμα εφαρμογής του WebExplorer σε μια ομάδα αρχείων, όπου και παρατηρούμε τις δυνατές επιλογές πάνω στα αρχεία και τους καταλόγους.



Εικόνα 15: Παράδειγμα χρήσης του Web Explorer

Σύγκριση:

Συγκρίνοντας τον WebExplorer με τον file manager μας, θα μπορούσαμε να πούμε πως στις βασικές λειτουργίες τους τα δύο συστήματα μοιάζουν αρκετά. Υπάρχουν όμως και αρκετές διαφορές που περιγράφονται παρακάτω:

1. Ταξινόμηση : Δεν είναι δυνατή η επιλογή της ταξινόμησης των αρχείων. Τα αρχεία είναι ταξινομημένα κατά αύξουσα αλφαβητική σειρά. Ο χρήστης δεν μπορεί να καθορίσει αυτός τον τρόπο που θα ταξινομούνται σε αντίθεση με το σύστημα μας όπου υπάρχουν 4 επιλογές ταξινόμησης.
2. Εμφάνιση των αρχείων σε στήλες : Τα περιεχόμενα ενός καταλόγου εμφανίζονται μόνο σε μορφή λίστας, το ένα κάτω από το άλλο και όχι σε μορφή πίνακα όπως στο δικό μας σύστημα.
3. Εμφάνιση /Απόκρυψη των λεπτομερειών των αρχείων: Για κάποια στοιχεία των αρχείων, όπως το μέγεθος ή την ημερομηνία δημιουργίας, δεν υπάρχει η δυνατότητα απόκρυψής τους ούτως ώστε να εμφανίζονται μόνο αν το επιθυμεί ο χρήστης.
4. Αντιγραφή/ Μετακίνηση καταλόγων: Δεν υπάρχει καν μια τέτοια επιλογή διαθέσιμη. Ενώ μπορούμε να μετακινήσουμε και να αντιγράψουμε αρχεία, δεν υπάρχει η ίδια δυνατότητα για τους καταλόγους. Έναν κατάλογο

μπορούμε μόνο να τον μετονομάσουμε ή να τον διαγράψουμε, σε αντίθεση με τον file manager μας όπου προσφέρεται και η δυνατότητα αντιγραφής ή μετακίνησης του.

5. Ομαδική αντιγραφή/μετακίνηση αρχείων και καταλόγων : Ενώ μπορούμε να αντιγράψουμε ή να μετακινήσουμε ένα αρχείο, δεν μπορούμε να επιλέξουμε μια ομάδα αρχείων και να εκτελέσουμε ταυτόχρονα σε αυτά την ‘Copy’ ή τη ‘Move’ λειτουργία.
6. Διαγραφή καταλόγου: Παρατηρούμε πως ενώ μπορούμε να διαγράψουμε αρχεία, δεν μπορούμε να διαγράψουμε έναν κατάλογο, εκτός και αν αυτός είναι άδειος. Κατάλογοι με ένα ή παραπάνω αρχεία δεν μπορούν να διαγραφούν. Θα πρέπει πρώτα να διαγραφούν όλα τα αρχεία του καταλόγου και μετά ο ίδιος ο κατάλογος. Κάτι το οποίο δεν συμβαίνει στο δικό μας file manager καθώς παρέχει τη δυνατότητα διαγραφής ενός καταλόγου με ταυτόχρονη διαγραφή όλων των δεδομένων του αναδρομικά.
7. Ο WebExplorer προσφέρει και την ‘touch’ επιλογή με την οποία επιτρέπει στον χρήστη να δημιουργήσει ένα timestamp για κάθε αρχείο, κάτι που δεν συμβαίνει στο δικό μας σύστημα.

Συνοψίζοντας, συμπεραίνουμε ότι ως προς το θέμα της εμφάνισης των αρχείων αλλά και των λειτουργιών επί αυτών, ο file manager μας υπερτερεί. Και αυτό γιατί παρέχει περισσότερες λειτουργίες επί των αρχείων και ταυτόχρονη δυνατότητα μεταπήδησης από λίστα σε πίνακα.

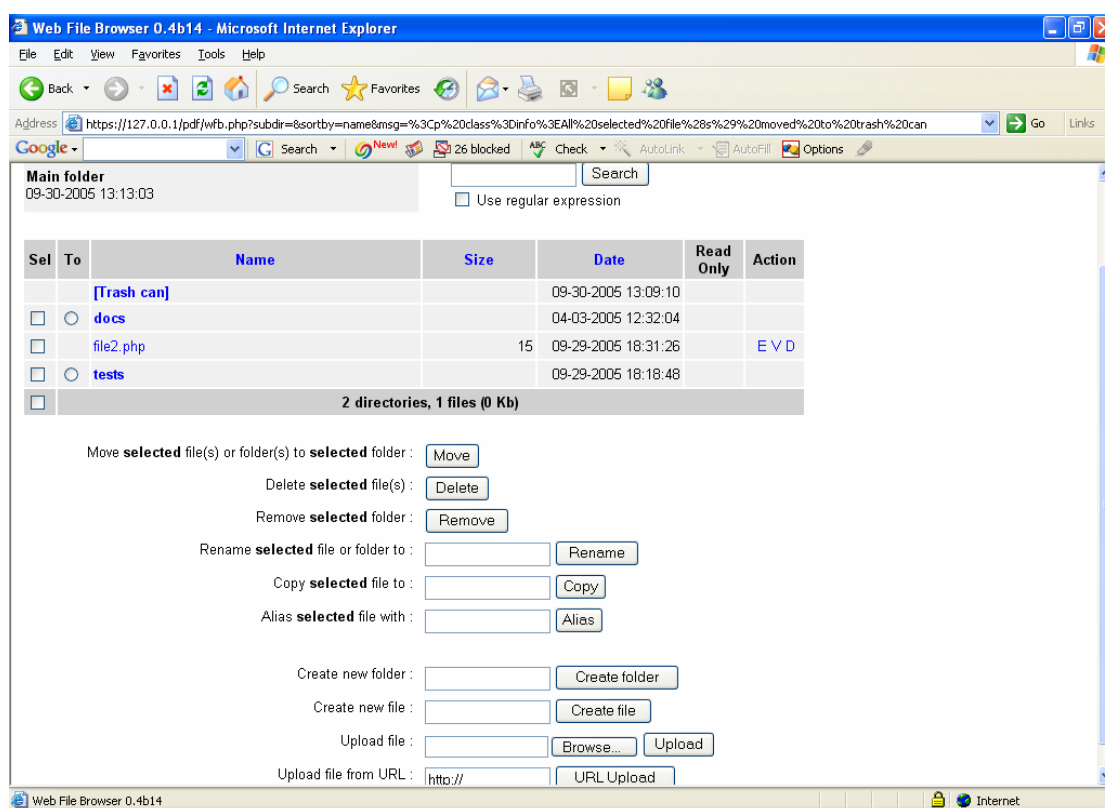
■ **WebFileBrowser** [25]

Ο WebFileBrowser είναι ένας web file manager γραμμένος στην γλώσσα PHP. Πρόκειται ίσως για τον πιο ολοκληρωμένο διαχειριστή αρχείων από όσους περιγράφηκαν. Παρέχει τις βασικές λειτουργίες και κάποιες επιπλέον δυνατότητες που δεν είναι δυνατές στο δικό μας σύστημα, όπως :

- 1) Αναζήτηση ενός αρχείου μέσα στους καταλόγους: Ο χρήστης μπορεί να δώσει το όνομα ενός αρχείου και το σύστημα θα ψάξει μέσα σε όλους τους υποκαταλόγους του τρέχοντος καταλόγου για να βρει το αρχείο. Σε περίπτωση επιτυχημένης αναζήτησης, το αρχείο εμφανίζεται στον χρήστη και ταυτόχρονα παρέχεται στον χρήστη η δυνατότητα να μεταφερθεί σε αυτόν τον κατάλογο και να δει τα περιεχόμενά του.
- 2) Χρήση κάδου ανακύκλωσης: Όταν ένα αρχείο ή κατάλογος διαγράφεται, μεταφέρεται στο trash can. Ο χρήστης μπορεί να μεταβεί στον κάδο ανακύκλωσης και να επαναφέρει ένα αρχείο αν το επιθυμεί ή να αδειάσει ολόκληρο τον κάδο ανακύκλωσης.

- 3) Φόρτωση αρχείου από ιστοσελίδα: Ο χρήστης μπορεί να δώσει το URL μιας ιστοσελίδας όπου υπάρχει το αρχείο που θέλουμε να φορτώσουμε και αυτό θα προστεθεί στα αρχεία του τρέχοντος καταλόγου.
- 4) Δημιουργία ενός αρχείου τύπου WFBALIAS: Ο χρήστης μπορεί να επιλέξει ένα αρχείο και να δημιουργήσει ένα alias αυτού του αρχείου. Έτσι στη λίστα των αρχείων, δε θα εμφανίζεται το αρχείο αλλά το alias του. Ανοίγοντας το θα εμφανίζονται τα περιεχόμενα του αρχικού αρχείου. Πρόκειται για ένα συμβολικό σύνδεσμο που θα δείχνει στο αρχικό αρχείο.

Ακολουθεί ένα παράδειγμα εμφάνισης των περιεχομένων ενός καταλόγου με τη χρήση του Web File Browser.



Εικόνα 16: Παράδειγμα χρήσης του Web File Browser

Σύγκριση:

Συγκρίνοντας τον WebFileBrowser με τον file manager μας, θα μπορούσαμε να πούμε πως ανακαλύψαμε πολλές ομοιότητες στις βασικές λειτουργίες τους. Και τα δυο συστήματα παρέχουν τις βασικές λειτουργίες όπως : New File, New Folder, Copy, Move, Delete, Rename, Upload. Οι επιπλέον δυνατότητες του WebFileBrowser περιγράφηκαν προηγουμένως. Υπάρχουν όμως και αρκετές διαφορές ανάμεσα στα δυο συστήματα όπως:

1. Ταξινόμηση : Ο WebFileBrowser ενώ χρησιμοποιεί αλφαβητική ταξινόμηση δεν ξεχωρίζει τους καταλόγους από τα αρχεία. Τα ταξινομεί όλα μαζί σε αντίθεση με το σύστημα μας όπου πρώτα εμφανίζονται όλοι οι κατάλογοι ταξινομημένοι και μετά τα αρχεία ταξινομημένα και αυτά. Ο τρόπος ταξινόμησης είναι μόνο ένας, ενώ στο δικό μας προσφέρονται τέσσερις επιλογές ταξινόμησης.
2. Διαγραφή καταλόγου: Ενώ είναι διαθέσιμη η πολλαπλή διαγραφή των αρχείων, δεν είναι δυνατή η διαγραφή ενός μη άδειου καταλόγου. Θα πρέπει πρώτα να διαγραφούν όλα τα περιεχόμενά του και μετά ο ίδιος ο κατάλογος.
3. Αντιγραφή καταλόγου: Δεν υπάρχει επιλογή αντιγραφής ενός καταλόγου αλλά μόνο ενός αρχείου, σε αντίθεση με το σύστημα μας όπου είναι δυνατές και οι δύο επιλογές.
4. Εμφάνιση των αρχείων σε στήλες : Τα περιεχόμενα ενός καταλόγου εμφανίζονται μόνο σε μορφή λίστας, το ένα κάτω από το άλλο και όχι σε μορφή πίνακα όπως στο δικό μας σύστημα.
5. Εμφάνιση /Απόκρυψη των λεπτομερειών των αρχείων: Για κάποια στοιχεία των αρχείων, όπως το μέγεθος ή την ημερομηνία δημιουργίας, δεν υπάρχει η δυνατότητα απόκρυψής τους ούτως ώστε να εμφανίζονται μόνο αν το επιθυμεί ο χρήστης.
6. Εμφάνιση δικαιωμάτων: Ο WebFileBrowser μας ενημερώνει αν ένα αρχείο ή ένας κατάλογος έχει μόνο το δικαίωμα ανάγνωσης, δηλαδή είναι read only. Δεν εμφανίζει αναλυτικά τα δικαιώματα των αρχείων, πράγμα που κάνει ο file manager μας.

Σε γενικές γραμμές, ο WebFileBrowser θεωρείται αρκετά καλός διαχειριστής αρχείων που προσφέρει και κάποιες δυνατότητες που δεν συναντούνται τόσο συχνά σε διαχειριστές αρχείων όπως η ύπαρξη του κάδου ανακύκλωσης ή το ψάξιμο ενός αρχείου ή η δημιουργία ενός alias κάποιου αρχείου. Παρ' όλα αυτά για τις ανάγκες που δημιουργήθηκε ο δικός μας διαχειριστής, οι επιλογές αυτές δεν ήταν απαραίτητες γι' αυτό και δεν τις αναπτύξαμε.

4.6.2 Συγκριτικός πίνακας

Ακολουθεί ένας πίνακας που συνοψίζει τα όσα αναφέρθηκαν στην προηγούμενη παράγραφο. Παρουσιάζει το σύνολο των λειτουργιών που παρέχονται από τα συστήματα που περιγράφηκαν. Για κάθε διαχειριστή αρχείων επισημαίνει ποιες από όλες τις λειτουργίες μπορεί να παρέχει στους χρήστες του. Έτσι, γίνεται μια άμεση σύγκριση των τεσσάρων παραπάνω διαχειριστών με το δικό μας σύστημα.

Δυνατότητες		Σύστημα		File Thingie	Php Directory Indexer	Web Explorer	Web File Browser	PPG File Manager
		File Thingie	Php Directory Indexer	Web Explorer	Web File Browser	PPG File Manager		
Ατομική διαγραφή	Αρχείου	✓		✓	✓	✓		
	Μη άδειου καταλόγου	✓				✓		
Ομαδική διαγραφή	Αρχείων	✓			✓	✓		
	Μη άδειων καταλόγων					✓		
Ατομική αντιγραφή/ μετακίνηση	Αρχείου			✓	✓	✓		
	Καταλόγου					✓		
Ομαδική αντιγραφή/ μετακίνηση	Αρχείων				✓	✓		
	Καταλόγων					✓		
Ταξινόμηση κατά:	Όνομα	✓	✓	✓	✓	✓		
	Μέγεθος	✓	✓			✓		
	Ημερομηνία	✓	✓			✓		
	Τύπο					✓		
Μετονομασία		✓		✓	✓	✓		
Αλλαγή του τύπου ενός αρχείου				✓	✓	✓		
Τροποποίηση		✓		✓	✓	✓		
Εμφάνιση σε κατακόρυφη λίστα		✓	✓	✓	✓	✓		
Εμφάνιση σε στήλες			✓			✓		
Εμφάνιση/απόκρυψη λεπτομερειών						✓		
Επιλογή του στυλ εμφάνισης των εικονιδίων			✓			✓		
Δημιουργία αρχείου		✓		✓	✓	✓		
Δημιουργία καταλόγου		✓		✓	✓	✓		
Μεταφόρτωση(upload) αρχείου		✓		✓	✓	✓		
Μεταφόρτωση(upload) αρχείου από URL					✓			
Αναζήτηση αρχείου					✓			
Επιλογή touch				✓				

Χρήση κάδου ανακύκλωσης				✓	
Δημιουργία αρχείου alias				✓	
Εμφάνιση δικαιωμάτων			✓		✓

Εικόνα 17: Σύγκριση των διαχειριστών αρχείων που περιγράφηκαν

4.6.3 Συνοψίζοντας

Παρουσιάσαμε κάποιους ενδεικτικούς διαχειριστές αρχείων που έχουν αναπτυχθεί προς χρήση μέσω ιστοσελίδων. Σκοπός της παρουσίασης αυτής ήταν να διαπιστώσουμε αν ο δικός μας διαχειριστής αρχείων, συγκρινόμενος με τους υπάρχοντες, παρέχει τουλάχιστον τις βασικές λειτουργίες. Διαπιστώσαμε πως κάτι τέτοιο ισχύει. Και μάλιστα πολλές φορές παρέχει και κάποιες επιπλέον δυνατότητες και επιλογές στους χρήστες από αρκετούς υπάρχοντες διαχειριστές αρχείων.

Το σημαντικότερο πλεονέκτημα του διαχειριστή μας θεωρήθηκε η δυνατότητα μεταπήδησης από τη μορφή πίνακα στη μορφή κατακόρυφης λίστας. Και αυτό γιατί στους περισσότερους διαχειριστές αρχείων τα αρχεία εμφανίζονται το ένα κάτω από το άλλο σε μορφή λίστας. Στο δικό μας σύστημα, εισήχθη η δυνατότητα επιλογής του τρόπου εμφάνισης από τον ίδιο τον χρήστη, ανάλογα με τις ανάγκες του. Αν έχει πολλά αρχεία και δεν τον ενδιαφέρει να βλέπει τις λεπτομέρειες των αρχείων αλλά μόνο το όνομά τους, επιλέγει εμφάνιση σε μεγάλο αριθμό στηλών. Έτσι θα έχει μπροστά του έναν περιεκτικό πίνακα με όλα τα αρχεία και θα μπορεί να τα βλέπει όλα ούτως ώστε να επιλέξει σε ποια από αυτά θα εκτελέσει τις λειτουργίες που επιθυμεί. Μάλιστα, του δίνεται η δυνατότητα να επιλέξει μια ομάδα από αυτά και να εκτελέσει την ίδια λειτουργία (copy - move - delete) σε όλα τα αρχεία της ομάδας. Έτσι, με την εμφάνιση σε μορφή πίνακα θα αποφύγει την μακροσκελή λίστα στην οποία εμφανίζονται συνήθως τα αρχεία στους περισσότερους file managers, όπου και έχουμε ένα αρχείο σε κάθε γραμμή της λίστας. Εδώ έχουμε περισσότερα από ένα αρχεία σε κάθε γραμμή, εξοικονομούμε χώρο και μπορούμε πιο εύκολα να εντοπίσουμε ένα αρχείο που μπορεί ψάχνουμε. Αν πάλι ο χρήστης θέλει να βλέπει τις λεπτομέρειες των αρχείων, μπορεί να αλλάξει τις τρέχουσες ρυθμίσεις και να επιλέξει την εμφάνιση των λεπτομερειών αυτών (Επιλογή 'File details').

Αν ο χρήστης θέλει να αφιερώσει περισσότερο χώρο σε κάθε αρχείο στον πίνακα ούτως ώστε να φαίνονται πιο καθαρά οι λεπτομέρειές του, μπορεί να επιλέξει εμφάνιση σε μικρότερο αριθμό στηλών. Αν πάλι προτιμά την εμφάνιση

σε λίστα, μπορεί να επιλέξει εμφάνιση σε μία στήλη οπότε και σε κάθε γραμμή θα εμφανίζεται ένα μόνο αρχείο. Στην περίπτωση αυτή, οι δυνατές λειτουργίες για κάθε αρχείο θα εμφανίζονται δίπλα από το όνομα του αρχείου καλύπτοντας ολόκληρη τη γραμμή που προσφέρεται για αυτό το αρχείο. Έτσι, εξασφαλίζουμε την εξοικονόμηση χώρου και τη μείωση του μεγέθους της λίστας.

Ένα ακόμη πλεονέκτημα του διαχειριστή μας είναι και η δυνατότητα που παρέχει στον χρήστη να επιλέξει ο ίδιος τον τρόπο ταξινόμησης. Είναι σημαντική η δυνατότητα ταξινόμησης κατά τύπο, η οποία δεν υπάρχει σε πολλούς php file managers. Έτσι, μπορεί ο χρήστης να βλέπει τα αρχεία ταξινομημένα σε ομάδες με βάση τον τύπο τους και να γνωρίζει τους διαθέσιμους τύπους και πόσα αρχεία υπάρχουν από κάθε τύπο.

Επίσης, αρκετά σημαντικό πλεονέκτημα θεωρείται η ομαδική διαγραφή, αντιγραφή ή μετακίνηση αρχείων και καταλόγων. Σε πολλά συστήματα, υπάρχει μια επιλογή διαγραφής δίπλα σε κάθε αρχείο. Στο δικό μας σύστημα, εκτός από αυτή την επιλογή προσφέρεται στο χρήστη και η δυνατότητα επιλογής πολλών αρχείων και καταλόγων και η ταυτόχρονη διαγραφή τους. Το ίδιο ισχύει και για τη μετακίνηση ή την αντιγραφή. Ακόμη, σε πολλούς file managers ένας κατάλογος διαγράφεται μόνο όταν είναι κενός. Πρέπει πρώτα να διαγραφούν όλα τα περιεχόμενά του και μετά ο ίδιος κάτι που δεν ισχύει στο δικό μας file manager. Ακόμη, αντιγράφοντας έναν κατάλογο, αυτός αντιγράφεται ολόκληρος στον κατάλογο προορισμού με όλα τα αρχεία και τους υποκαταλόγους του. Το ίδιο ισχύει και για τη μετακίνηση.

Τέλος, θα πρέπει να τονισθεί ότι οι περισσότεροι από τους παραπάνω διαχειριστές αρχείων αποτελούν ένα πακέτο που δεν περιέχει ένα αρχείο μόνο, αλλά πολλά περισσότερα. Το δικό μας πακέτο αποτελείται από ένα αρχείο, έναν κατάλογο με τα εικονίδια και το css αρχείο. Ο κώδικας για όλες τις λειτουργίες που παρέχει το σύστημα βρίσκεται μέσα σε αυτό το αρχείο. Έτσι, η εγκατάστασή του είναι ιδιαίτερα εύκολη και απαιτεί λιγότερο ελεύθερο χώρο στον server που θα εγκατασταθεί.

Βέβαια, θα πρέπει να σημειωθεί πως υπάρχουν πολλοί περισσότεροι διαχειριστές αρχείων από αυτούς που περιγράφηκαν. Οι περισσότεροι όμως έχουν ένα βασικό μειονέκτημα: δεν είναι αυτόνομοι αλλά είναι ενσωματωμένοι σε ένα ήδη υπάρχον σύστημα. Έτσι, η εφαρμογή τους σε οποιοδήποτε σύστημα είναι ιδιαίτερα δύσκολη καθώς απαιτείται η ενσωμάτωση ολόκληρου του πακέτου των αρχείων που τους υλοποιεί. Το πρόβλημα αυτό δεν υπήρχε στους διαχειριστές που περιγράφηκαν αλλά ούτε και στον δικό μας, καθώς όλοι είναι αυτόνομοι.

Κεφάλαιο 5

Σύνοψη και μελλοντικές επεκτάσεις

5.1 Σύνοψη

Ο στόχος της εργασίας ήταν η δημιουργία ενός συστήματος για την ικανοποίηση των τυπικών αναγκών μιας ακαδημαϊκής ερευνητικής ομάδας. Οι ανάγκες αυτές περιελάμβαναν:

- Τη δημιουργία ενός συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων:

Σκοπός της υλοποίησης του συστήματος αυτού ήταν η διαχείριση των εργασιών της ερευνητικής ομάδας μέσω του Web. Ως πρακτική εφαρμογή, το σύστημα που δημιουργήθηκε ενσωματώθηκε στις ιστοσελίδες του PPG, αλλά με ελάχιστες αλλαγές των παραμέτρων του μπορεί να ενσωματωθεί σε οποιοδήποτε ιστότοπο, όπου υπάρχει η ανάγκη διαχείρισης βιβλιογραφίας. Βασική προϋπόθεση είναι η ύπαρξη των εργαλείων PHP, Apache και MySQL. Οι υπηρεσίες που παρέχει το σύστημα είναι οι εξής:

- Προβολή της λίστας των εργασιών (βιβλίων, άρθρων, πτυχιακών εργασιών, τεχνικών αναφορών, δημοσιεύσεων) της ερευνητικής ομάδας.
- Αναζήτηση ανάμεσα στις εργασίες με βάση διάφορα κριτήρια
- Εισαγωγή στην Ιδιωτική Περιοχή του συστήματος, με χρήση του κατάλληλου μυστικού κωδικού και:
 - Εισαγωγή νέων εργασιών.
 - Προσθήκη των εργασιών και σε ηλεκτρονική μορφή (διαδικασία upload).
 - Τροποποίηση των υπαρχόντων εργασιών.
 - Διαγραφή των υπαρχόντων εργασιών.

- Τη δημιουργία ενός διαχειριστή αρχείων:

Σκοπός της υλοποίησης του συστήματος αυτού ήταν η παροχή απομακρυσμένης πρόσβασης στα ηλεκτρονικά αρχεία της ομάδας που θα ανήκουν στον κοινόχρηστο χώρο του συστήματος. Υπάρχουν διάφοροι τύποι

αρχείων που μπορούν να αποτελέσουν την ομάδα των ηλεκτρονικών αρχείων του PPG, όπως για παράδειγμα pdf, ps, txt, zip, php, html. Κατά τη διάρκεια υλοποίησης του συστήματος όμως, δεν τέθηκε κανένας περιορισμός στον τύπο των αρχείων της ομάδας. Κάθε ηλεκτρονική μορφή είναι αποδεκτή, ακόμα και αρχεία ήχου ή εικόνας. Οι δυνατότητες που παρέχει ο διαχειριστής είναι οι εξής:

- Δημιουργία νέου αρχείου και καταλόγου
- Προσθήκη ενός αρχείου οποιουδήποτε τύπου μέσω της διαδικασίας upload
- Μετονομασία αρχείου και καταλόγου
- Τροποποίηση αρχείου
- Ατομική/Ομαδική διαγραφή αρχείων και καταλόγων
- Ατομική/Ομαδική αντιγραφή αρχείων και καταλόγων
- Ατομική/Ομαδική μετακίνηση αρχείων και καταλόγων
- Ταξινόμηση καταλόγων και αρχείων
- Εμφάνιση/απόκρυψη των λεπτομερειών των αρχείων
- Εμφάνιση των περιεχομένων του καταλόγου σε στήλες που επιλέγονται από τον χρήστη

Και τα δύο συστήματα που δημιουργήθηκαν, εφαρμόστηκαν στις ιστοσελίδες του PPG αλλά με την αλλαγή ελάχιστων παραμέτρων μπορούμε να επιτύχουμε την ασφαλή μεταφορά τους σε οποιοδήποτε σύστημα ομάδας με παρόμοιες ανάγκες. Στόχος μας ήταν να εξασφαλίσουμε πως τα συστήματα που υλοποιήθηκαν θα είναι αρκετά ικανά για τον σκοπό που δημιουργήθηκαν γι αυτό και κατά τη διάρκεια υλοποίησης τους προσπαθήσαμε να προσαρμοστούμε στις ανάγκες της συγκεκριμένης εργασίας που μας είχε ανατεθεί.

5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Μια σημαντική βελτίωση που θα μπορούσε να γίνει στο σύστημα διαχείρισης βιβλιογραφίας και δημοσιεύσεων είναι η αυτόματη εξαγωγή των στοιχείων της βάσης στη μορφή που απαιτεί το BIBTEX [4]. Το BIBTEX είναι ένα πρόγραμμα που ορίζει μια συγκεκριμένη μορφή την οποία πρέπει να έχουν τα πεδία μιας ερευνητικής ομάδας, όπως για παράδειγμα τον τρόπο σύνταξης του συγγραφέα, του τίτλου της εργασίας και άλλα. Είναι απαραίτητο κατά τη χρήση του LATEX[5], που αποτελεί έναν από τους πιο δημοφιλείς επεξεργαστές κειμένου σε πολλά επιστημονικά πεδία.

Κατά τη χρήση του LATEX, τα βιβλιογραφικά στοιχεία που προκύπτουν (π.χ αναφορές σε ερευνητικές εργασίες) διεκπεραιώνονται μέσω του προγράμματος BIBTEX.

Σημαντικές βελτιώσεις θα μπορούσαν να γίνουν και στο διαχειριστή αρχείων που δημιουργήθηκε. Και αυτό γιατί παρόλο που παρέχει αρκετές δυνατότητες στον χρήστη, υπάρχουν ακόμη κάποιοι τομείς στους οποίους θα μπορούσε να βελτιωθεί.

Όπως αναφέρθηκε και στο κεφάλαιο 'Ιδιαιτερότητες υλοποίησης', υπάρχουν αρκετές μεταβλητές του προγράμματος (που αποτελούν και τις 'USER CONFIGURATIONS' του συστήματος) που μπορούν να αλλάξουν τιμές μελλοντικά με σκοπό την επέκταση των ρυθμίσεων του διαχειριστή αρχείων αλλά και την γενικότερη ανάπτυξη του. Για παράδειγμα, μπορεί να τροποποιηθεί ο αριθμός των στηλών εμφάνισης των περιεχομένων ενός καταλόγου, ανάλογα με τις ανάγκες της ιστοσελίδας στην οποία ενσωματώθηκε ο συγκεκριμένος διαχειριστής αρχείων. Ακόμη, μπορεί να αλλάξουν οι επιλογές ταξινόμησης προσθέτοντας νέους τρόπους ταξινόμησης αλλά και η μορφή εμφάνισης των αρχείων αλλάζοντας το .css αρχείο που έχει δημιουργηθεί για τον διαχειριστή αρχείων. Έτσι, ο διαχειριστής μπορεί να τροποποιηθεί και να αλλάξει η μορφή του για να προσαρμοστεί στο γενικότερο στυλ του συστήματος στο οποίο τοποθετείται.

Παρόλα αυτά, υπάρχουν και κάποιες λειτουργίες που μπορούν να προστεθούν στον διαχειριστή αρχείων που δημιουργήθηκε, όπως για παράδειγμα η εμφάνιση των δικαιωμάτων των αρχείων και η δυνατότητα τροποποίησης τους (chmod λειτουργία). Ακόμη, θα μπορούσαμε να προσθέσουμε την δυνατότητα αναζήτησης ενός αρχείου μέσα στους υποκαταλόγους του τρέχοντος καταλόγου, τη δυνατότητα ύπαρξης κάρδου ανακύκλωσης, τη δυνατότητα αποκοπής ενός αρχείου και επικόλλησης του σε έναν άλλο κατάλογο (cut and paste), τη δυνατότητα δημιουργίας συμβολικού συνδέσμου ενός αρχείου ή καταλόγου και τη δυνατότητα προσθήκης ενός αρχείου μέσα από ένα URL που θα δίνει ο χρήστης.

Συμπερασματικά, σε κάθε περίπτωση, μπορούν να γίνουν όσες αλλαγές είναι απαραίτητες προκειμένου να επιτύχουμε την ικανοποίηση των αναγκών του χρήστη. Ανάλογα με τον λόγο για τον οποίο θέλουμε έναν τέτοιο διαχειριστή αρχείων, μπορούμε να αναπτύξουμε και τα διάφορα τμήματα του, προσθέτοντας ή αφαιρώντας συνεχώς επιλογές.

Βιβλιογραφία

- [1] Chris Lea, Wankyu Choi, Allian Kent, Ganesh Prasad and Chris Ullman, *Beginning PHP4 (Programmer to Programmer)*, 2003.
- [2] Chris Lea, Mike Buzzard, Dilip Thomas, Jessey-White and Cinis NA, *PHP MySQL Website Programming: Problem - Design – Solution*, 2003.
- [3] Michael K. Glass, Yann Le Scouarnec, Elizabeth Naramore, Gary Mailer, Jeremy Stolz and Jason Gerner, *Beginning PHP, Apache, MySQL Web Development*, 2002.
- [4] Oren Patashnik, *BibTeXing*, 1988.
- [5] Leslie Lamport, *LaTeX: A Document Preparation System*, 1985
- [6] The PHP homepage
<http://www.php.net>
- [7] The APACHE homepage
<http://www.apache.org/>
- [8] MYSQL :: The World's most popular Open Source Database
<http://www.mysql.com/>
- [9] The Netcraft Web survey, September 2005
http://news.netcraft.com/archives/web_server_survey.html
- [10] The PHP usage statistics
<http://www.php.net/usage>
- [11] PHP History
<http://gr.php.net/history>
- [12] PEAR :: The PHP Extension And Application Repository
<http://pear.php.net/>
- [13] The PHP-GTK project
<http://gtk.php.net/>

- [14] The Forrester Research: Open Source Databases Come Of Age, December 2004
<http://www.mysql.com/why-mysql/analyst-reports/forrester-open-source-databases.pdf>

- [15] PHP mailing lists
http://www.scriptsbank.com/PHP_Scripts/Mailing_List_Managers/

- [16] The Netcraft survey for Apache
<http://news.netcraft.com/>

- [17] The Apache authentication
<http://httpd.apache.org/docs/1.3/howto/auth.html>

- [18] The new world PHP - Scripts and Programs - File Managers
<http://www.the-new-world.com/list/PHP/Scripts+and+Programs/File+Managers/>

- [19] File Managers
<http://www.alwaysfreeware.co.uk/fmanagers.html>

- [20] Hotscripts.com ::PHP :: Scripts and Programs :: File Manipulation :: File Management
http://www.hotscripts.com/PHP/Scripts_and_Programs/File_Manipulation/File_Management/index.html

- [21] Zend, the PHP company - File Management
<http://www.zend.com/codex.php?CID=283>

- [22] File thingie : A small php file manager for download
<http://www.solitude.dk/filethingie/>

- [23] The PHP Directory Indexer
<http://linux.ues.edu.sv/servidor/phpwebthings/indexer/indexer-0.2/>

- [24] The WebExplorer
<http://www.suneworld.com/index.php?pagename=webexplorer.php>

- [25] The WebFileBrowser
<http://sourceforge.net/projects/webfilebrowser/>

Παράρτημα Α

Οδηγίες εγκατάστασης του συστήματος

Η χρήση και των δύο τμημάτων της πτυχιακής εργασίας, μπορεί να γενικευτεί. Και τα δύο τμήματα μαζί μπορούν να αποτελέσουν ένα **πακέτο PHP** που μπορεί να χρησιμοποιηθεί από οποιονδήποτε επιθυμεί να δημιουργήσει ένα σύστημα διαχείρισης βιβλιογραφικών δεδομένων και κοινόχρηστων αρχείων.

A.1 Σύστημα διαχείρισης βιβλιογραφίας

Το σύστημα διαχείρισης βιβλιογραφίας που υλοποιήθηκε είναι αυτόνομο και ικανό να ενσωματωθεί σε οποιονδήποτε ιστότοπο που βασίζεται στο συνδυασμό Apache, PHP και MySQL. Οι μόνες αλλαγές που απαιτούνται είναι οι εξής:

- Αλλαγή του css αρχείου. Όλες οι οπτικές ρυθμίσεις βρίσκονται σε αυτό το αρχείο. Προκειμένου να προσαρμοστεί το σύστημα μας στις ανάγκες του εκάστοτε χρήστη, αρκεί να τροποποιηθεί αυτό το αρχείο.
- Αλλαγή του αρχείου με τον κωδικό πρόσβασης στη βάση: Όλες οι ρυθμίσεις που αφορούν την πρόσβαση στον server και κατόπιν στη βάση του συστήματος, όπως το username και το password, βρίσκονται σε ένα μόνο αρχείο. Η αλλαγή των μεταβλητών αυτού του αρχείου είναι αρκετή για να αποκτήσει το σύστημα πρόσβαση στον server του νέου ιστότοπου.
- Όταν κάποιος θελήσει να ενσωματώσει το σύστημα μας στον ιστότοπο του, θα πρέπει να δημιουργήσει μια βάση δεδομένων και έναν πίνακα με πεδία ίδια με τον δικό μας. Έπειτα, θα πρέπει να τροποποιήσει τις μεταβλητές που χρησιμοποιούνται για το όνομα της βάσης και του πίνακα στα php αρχεία που υλοποιούν το σύστημα, ώστε αυτές να ανταποκρίνονται στα ονόματα της βάσης και του πίνακα του νέου συστήματος. Οι μεταβλητές αυτές βρίσκονται στο αρχείο που περιγράφηκε προηγουμένως, όπου υπάρχει και ο κωδικός πρόσβασης στη βάση.
- Στο ίδιο αρχείο βρίσκεται και η μεταβλητή που καθορίζει τον κατάλογο στον οποίο αποθηκεύονται οι ηλεκτρονικές μορφές των αρχείων κατά την φόρτωση μιας νέας εργασίας στο σύστημα από τους χρήστες. Η τιμή της μεταβλητής μπορεί να αλλάξει ανάλογα με τον νέο κατάλογο που επιθυμούμε να μεταφέρονται οι εργασίες.

A.2 Σύστημα διαχείρισης αρχείων

Ο διαχειριστής αρχείων που δημιουργήθηκε για την επεξεργασία των αρχείων ενός κοινόχρηστου χώρου, μπορεί να τοποθετηθεί σε οποιοδήποτε κατάλογο τα περιεχόμενα του οποίου θέλουμε να επεξεργαζόμαστε μέσω του Web. Το μεγάλο πλεονέκτημα του είναι ότι αποτελεί ένα πακέτο με έναν κατάλογο και δύο αρχεία μόνο. Ο κατάλογος περιέχει τα εικονίδια του συστήματος: αυτά που είναι διαθέσιμα για τους διάφορους τύπους αρχείων και κάποια άλλα που περιγράφουν τις προσφερόμενες λειτουργίες πάνω στα αρχεία και τους καταλόγους, όπως για παράδειγμα το εικονίδιο διαγραφής. Το πρώτο αρχείο είναι γραμμένο στην γλώσσα PHP και αποτελεί ουσιαστικά εκείνο στο οποίο υλοποιείται ο διαχειριστής αρχείων και το δεύτερο είναι το .css αρχείο που καθορίζει τη μορφή του διαχειριστή. Έτσι, παίρνοντας κάποιος το πακέτο του file manager θα χρειαστεί να τροποποιήσει ελάχιστα πράγματα προκειμένου να τον χρησιμοποιήσει για τους δικούς του σκοπούς.

Η εγκατάστασή του διαχειριστή είναι πολύ εύκολη και γρήγορη και περιγράφεται στη συνέχεια:

- Μεταφορά ολόκληρου του πακέτου (των δύο αρχείων και του καταλόγου των εικονιδίων) στον κατάλογο τα περιεχόμενα του οποίου επιθυμούμε να διαχειριστούμε μέσω του file manager.
- Αλλαγή τριών μεταβλητών του αρχείου που υλοποιεί τον file manager:
 - Αλλαγή της μεταβλητής \$rootname που αποτελεί το όνομα με το οποίο εμφανίζεται ο αρχικός κατάλογος στην κορυφή του file manager. Η τρέχουσα τιμή της είναι demo, οπότε όταν θα βρισκόμαστε στον αρχικό κατάλογο θα εμφανίζεται στην κορυφή του file manager το εξής μονοπάτι: **Directory: demo/**. Η τιμή της μεταβλητής αυτής μπορεί να αλλάξει ανάλογα με τις ανάγκες του νέου χρήστη.
 - Αλλαγή της μεταβλητής \$fmicondir που αποτελεί και το όνομα του καταλόγου με τα εικονίδια του συστήματος. Ο νέος χρήστης θα πρέπει να θέσει σαν τιμή της μεταβλητής το όνομα του καταλόγου όπου είναι αποθηκευμένα τα δικά του εικονίδια για να μπορέσει να τα αναγνωρίσει ο νέος διαχειριστής αρχείων.
 - Αλλαγή της μεταβλητής \$fmcss που αποτελεί και το όνομα του css αρχείου του συστήματος προκειμένου να εφαρμοσθεί στον file manager το στυλ εμφάνισης που επιθυμεί ο νέος χρήστης.

Έτσι, τοποθετώντας το PHP αρχείο στον κατάλογο που επιθυμούμε και ονομάζοντας το index.php, έχουμε επιτύχει το εξής: Όταν κάποιος χρήστης του συστήματος προσπαθήσει να εισέλθει στον κατάλογο αυτό, το σύστημα θα

καλέσει το αρχείο `index.php` οπότε και θα ενεργοποιηθεί ο `file manager` με αποτέλεσμα την εμφάνιση των αρχείων και των υποκαταλόγων του αρχικού μας καταλόγου και τη δυνατότητα περαιτέρω επεξεργασίας τους μέσω του `file manager`.

Ακολουθεί ένα παράδειγμα εγκατάστασης του `file manager` σε έναν κατάλογο ενός συστήματος που επεξηγεί τα όσα αναφέρθηκαν.

- Αντιγραφή του αρχείου που υλοποιεί τον `file manager` στον επιθυμητό κατάλογο.
- Μετονομασία του αρχείου, αν αυτό είναι επιθυμητό. Μπορούμε να του δώσουμε οποιοδήποτε όνομα επιθυμούμε. Ανεξάρτητα από το όνομα που θα δοθεί στο αρχείο, αυτό δεν θα εμφανίζεται στην λίστα του `file manager` για να μην είναι ορατό από τον κάθε χρήστη του συστήματος. Έστω ότι του δίνεται το όνομα `sfm.php`. Ας σημειωθεί ότι αν δεν μετονομάσουμε το αρχείο, αυτό θα αντιγραφεί στον κατάλογο μας με το τρέχον όνομα του δηλαδή ως `index.php`.
- Αλλαγή των μεταβλητών `$rootname`, `$fmicondir` και `$fmcss` για να προσαρμοστούν στα νέα δεδομένα του συστήματος.

Αν στη συνέχεια ανοίξουμε τον `Web Browser` μας και πάμε στην διεύθυνση <http://ourserver.ourdomain/ourdirectory/sfm.php> ή στην <http://ourserver.ourdomain/ourdirectory/index.php>, θα αντικρίσουμε τους υποκαταλόγους και τα αρχεία του αρχικού μας καταλόγου (`ourdirectory`) μαζί με όλες τις δυνατές λειτουργίες του `filemanager`.

Παράρτημα Β

Πηγαίος κώδικας

B.1 Τα αρχεία του συστήματος διαχείρισης βιβλιογραφίας και δημοσιεύσεων

Ακολουθεί η παρουσίαση κάποιων ενδεικτικών php αρχείων που δημιουργήθηκαν προκειμένου να υλοποιηθεί το σύστημα διαχείρισης βιβλιογραφίας και δημοσιεύσεων. Στα αρχεία αυτά υλοποιούνται οι βασικές λειτουργίες που μπορεί να εκτελέσει ο χρήστης πάνω στις υπάρχουσες εργασίες του συστήματος. Έτσι, μέσα στον κώδικα τους διαφαίνεται η χρήση των κλήσεων της MySQL για την επεξεργασία των δεδομένων της βάσης του συστήματος. Θα παρουσιασθούν συνολικά 7 αρχεία που περιλαμβάνουν τις εξής λειτουργίες: προβολή λίστας εργασιών, αναζήτηση εργασίας μέσα στη βάση του συστήματος, εισαγωγή εργασίας, φόρτωση της ηλεκτρονικής της μορφής, εμφάνιση της λίστας των εργασιών με τις δυνατότητες τροποποίησης/διαγραφής και τροποποίηση/διαγραφή εργασίας.

Λίστα αρχείων:

- showpub.inc.php
- searchpubs.php
- dbinsert.php
- mod_del_list.php
- modify.php
- dbmodify.php
- delete.php

B.1.1 showpub.inc.php

```
<?php
/* SHOWPUB.INC.PHP
 *   Produces the HTML code for displaying a publication.
 *   The top section is the style used for each field, and
 *   should be customized according to taste.
 *   Here we follow roughly the IEEE style.
 *
 */
```



```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*   STYLES FOR THE PUBLICATION FIELDS
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

$style["Title"]["before"] = "&#8220;"; // Smart quotes
$style["Title"]["after"] = "&#8221;";

$style["Authors"]["before"] = "";
$style["Authors"]["after"] = "";

$style["Journal"]["before"] = "<i>";
$style["Journal"]["after"] = "</i>";

$style["Volume"]["before"] = "Vol. ";
$style["Volume"]["after"] = "";

$style["Number"]["before"] = "No. ";
$style["Number"]["after"] = "";

$style["Conference"]["before"] = "in <i>Proc. ";
$style["Conference"]["after"] = "</i>";

$style["Address"]["before"] = "";
$style["Address"]["after"] = "";

$style["Booktitle"]["before"] = "<i>";
$style["Booktitle"]["after"] = "</i>";

$style["Editors"]["before"] = "";
$style["Editors"]["after"] = " eds.";

$style["Publisher"]["before"] = "";
$style["Publisher"]["after"] = "";

$style["Month"]["before"] = "";
$style["Month"]["after"] = "";

$style["Year"]["before"] = "";
$style["Year"]["after"] = "";

$style["Info"]["before"] = "";
$style["Info"]["after"] = "";

$style["Pages"]["before"] = "pp. ";
$style["Pages"]["after"] = "";

/* We assume the stored month is an int from 1 to 12
*/
$monthname = array(
    "what month is this?",
    "Jan.", "Feb.", "Mar.", "Apr.", "May", "June",
    "July", "Aug.", "Sept.", "Oct.", "Nov.", "Dec."
);

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*   ACTUAL CODE
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

```

```

/* Generates php code to be eval()'ed later, and
 * produce HTML for a publication field.
 */
function code($field)
{
    global $style;

    $string =
        "if ($" . $field . ") {" .
        "    if (\$comma) echo \", \";" .
        "    echo \"" . $style["$field"] ["before"] . "$" . $field .
        $style["$field"] ["after"] . "\";" .
        "    \$comma = 1; " .
        "} ";

    return ($string);
}

/*
 * Display the idx-th of the DB records
 */
function show_publication($record, $pubsurl)
{
    global $monthname;

    $papertype = $record["Type"];
    $Title      = $record["Title"];
    $Authors    = $record["Authors"];
    $Journal    = $record["Journal"];
    $Volume     = $record["Volume"];
    $Number     = $record["Number"];
    $Conference = $record["Conference"];
    $Address    = $record["Address"];
    $Booktitle  = $record["Booktitle"];
    $Editors    = $record["Editors"];
    $Publisher  = $record["Publisher"];
    $Month      = $monthname[ $record["Month"] ];
    $Year       = $record["Year"];
    $Info       = $record["Info"];
    $Pages      = $record["Pages"];

    $fname1     = $record["File1"];    // The 5 uploaded files
    $fname2     = $record["File2"];
    $fname3     = $record["File3"];
    $fname4     = $record["File4"];
    $fname5     = $record["File5"];
    $ftype1     = $record["Type1"];    // Their types
    $ftype2     = $record["Type2"];
    $ftype3     = $record["Type3"];
    $ftype4     = $record["Type4"];
    $ftype5     = $record["Type5"];

    /* Generate PHP code for each field */
    $Titlecode  = code("Title"      );
    $Authorscode = code("Authors"   );
    $Journalcode = code("Journal"   );
    $Volumecode  = code("Volume"    );
    $Numbercode  = code("Number"    );

```

```

$Conferencecode = code("Conference");
$Addresscode    = code("Address"  );
$Booktitlecode  = code("Booktitle");
$Editorscode    = code("Editors"  );
$Publishercode  = code("Publisher" );
$Monthcode     = code("Month"     );
$Yearcode      = code("Year"      );
$Infocode      = code("Info"      );
$Pagescode     = code("Pages"     );

$comma = 0;      /* Flag gia swsth efanish kommatwn */

switch ($papertype)
{
  case "Journal article":
    eval($Authorscode);
    eval($Titlecode);
    eval($Journalcode);
    eval($Volumecode);
    eval($Numbercode);
    eval($Monthcode);
    if ($Month) { $commabak = $comma; $comma = 0; echo " "; }
    eval($Yearcode);
    if ($Month) { $comma = $commabak || $comma; }
    eval($Pagescode);
    break;

  case "Conference article":
    eval($Authorscode);
    eval($Titlecode);
    eval($Conferencecode);
    eval($Addresscode);
    eval($Monthcode);
    if ($Month) { $commabak = $comma; $comma = 0; echo " "; }
    eval($Yearcode);
    if ($Month) { $comma = $commabak || $comma; }
    eval($Pagescode);
    break;

  case "Technical report":
    eval($Authorscode);
    eval($Titlecode);
    eval($Addresscode);
    eval($Numbercode);
    eval($Monthcode);
    if ($Month) { $commabak = $comma; $comma = 0; echo " "; }
    eval($Yearcode);
    if ($Month) { $comma = $commabak || $comma; }
    break;

  case "Book chapter":
    eval($Authorscode);
    eval($Titlecode);
    eval($Booktitlecode);
    eval($Editorscode);
    eval($Publishercode);
    eval($Yearcode);
    eval($Pagescode);
    break;

  case "Book":

```

```

    eval($Authorscode);
    eval($Booktitlecode);
    eval($Publishercode);
    eval($Yearcode);
    break;

case "Thesis":
    eval($Authorscode);
    eval($Booktitlecode);
    eval($Infocode);
    eval($Numbercode);
    eval($Addresscode);
    eval($Monthcode);
    if ($Month) { $commabak = $comma; $comma = 0; echo " "; }
    eval($Yearcode);
    if ($Month) { $comma = $commabak || $comma; }
    break;

case "Other":
default:
    eval($Authorscode);
    eval($Titlecode);
    eval($Infocode);
    break;
}

if ($comma)
    echo "<br>";

/*
 * Output the list of offered files
 */

$comma = 0;      /* Flag gia swsth efanish kommatwn */

for ($i = 1; $i <= 5; $i++)
{
    $name = "fname$i";
    $url = "furl$i";
    $type = "ftype$i";

    if ($$name) {
        if ($comma) echo ", ";
        echo "<a href=\"";
        if ($i < 3) echo "$pubsurl/";      /* Last 2 are not files */
        echo $$name . "\">" . $$type . "</a>";
        $comma = 1;
    }
}

if ($comma)
    echo "<br>";
}

?>

```

B.1.2 searchpubs.php

```
<?php

/*
 * Search driver for our publications
 * Accepts the following GET stuff:
 *   query (what to search for - if null, it means "")
 *   where (= all/authors/titles/years/keywords - if null means all)
 *   sortby (= ddate/adate - if null, decreasing date by default)
 *   papers/trs/theses/others (if present, include the corresponding
pub type)
 *   alltypes (if present, list all pubs - so as to avoid the above 4
flags)
 */

/* This one prepends backslashes (\) before
 * any dangerous user input, like quotes.
 */
function fix($string)
{
    if (!get_magic_quotes_gpc()) /* Off */
        return ( addslashes($string) );
    else
        return ($string);
}

include("dbinfo.inc.php");
include("showpub.inc.php");

$query    = fix( rawurldecode($_GET["query"]) );
$where    = rawurldecode($_GET["where"]); /*
all,authors,titles,years,keywords */
$sortby   = rawurldecode($_GET["sortby"]); /* ddate, adate */
$papers   = $_GET["papers"];
$trs      = $_GET["trs"];
$theses   = $_GET["theses"];
$others   = $_GET["others"];
$allflag  = $_GET["alltypes"];

/* No need to continue */
if (!$papers && !$trs && !$theses && !$others && !$allflag)
    exit;

/* Connect to DB */
$mid = @mysql_connect("agatha.cs.uoi.gr", $username, $password)
        or die("Unable to connect to database server!");
@mysql_select_db($database,$mid)
        or die( "Unable to select required database!");

/* The query string
 */

$sortstr = "Year desc, Month desc, id desc";
if ($sortby == "adate")
    $sortstr = "Year asc, Month asc, id asc";

$what = "";
if ($papers)
    $what .= " Type='Journal article' OR Type='Conference Article' ";
```

```

if ($trs)
  $what .= (($what != "") ? " OR " : "" ) . " Type='Technical report'
";
if ($theses)
  $what .= (($what != "") ? " OR " : "" ) . " Type='Thesis' ";
if ($others)
  $what .= (($what != "") ? " OR " : "" ) .
    " Type='Book' OR Type='Book chapter' OR Type='Other' ";

$ask = "";
if ($query != "")
{
  $ask = ($what == "") ? "(" : "AND (";

  if ($where == "authors")
    $ask .= "Authors LIKE '%$query%'";
  elseif ($where == "titles")
    $ask .= "Title LIKE '%$query%'";
  elseif ($where == "years")
    $ask .= "Year LIKE '%$query%'";
  elseif ($where == "keywords")
    $ask .= "Keywords LIKE '%$query%'";
  else
    $ask .= "(Authors LIKE '%$query%') OR (Title LIKE '%$query%') OR
    .
      (Year LIKE '%$query%') OR (Keywords LIKE '%$query%')";

  $ask .= ")";
}

$lookfor = "";
if ($what != "" || $ask != "")
{
  if ($what == "")
    $lookfor = "WHERE $ask";
  else
    $lookfor = "WHERE ($what) $ask";
}

$qstr = "SELECT * FROM $tablename $lookfor ORDER BY $sortstr";
$result = mysql_query($qstr);

/* Elegxos gia 0 papers sth bash */
if(mysql_numrows($result) == 0)
{
  mysql_close();
  echo "\nNo publications found.\n\n";
  die();
}

while ($row = mysql_fetch_array($result))
{
  echo "<li>";
  show_publication($row, $pubsurl);
  echo "<br></li>\n";
}

mysql_close();

?>

```

B.1.3 dbinsert.php

```
<?php

/* DBINSERT.PHP
 *   Inserts a new publication in the DB
 *
 */

include("html.inc.php");
include("../dbinfo.inc.php");

$message = "";

/* Upload 1 file
 */
function upload_file($fid, &$type, &$newname)
{
    global $pubspath; // Needed
    global $message;

    $file = $_FILES["$fid"]["name"];
    $size = $_FILES["$fid"]["size"];
    if (!$type) // Get the one from the system
        $type = $_FILES["$fid"]["type"];

    if ($file == "")
    {
        $newname = "";
        return (1); // All ok */
    }

    /* periptwsh pou einai eite adeio to arxeio pou bazw h den yparxei
     * to arxeio ayto! (oxi otan den kanw upload).
     * Den ginetai h eisagwgh alla epistrefw pisw gia na diorthwsw to
     * onoma tou arxeiou
     */
    if ($size <= 0)
    {
        $message .= "<b>Error!!</b><br><br>\nFile \"" . basename($file) .
            "\" does not exist or is empty.<br>\n" .
            "Go back & check the paths of the files you want to
upload.\n";
        return (0);
    }

    //anoigw to arxeio gia na parw to metrthth
    if (!$fp = fopen("counter.txt", "r"))
    {
        $message .= "<b>Error!!</b><br><br>\nCannot read the counter
file.\n";
        return (0);
    }
    $i = fread($fp, 20);
    fclose($fp);

    $name = basename($file);
    $newname = $i . $name; // Make unique name
```

```

$uploadfile = $pubspath . $newname; // Final destination

    if(!move_uploaded_file($_FILES["$fid"]["tmp_name"],$uploadfile)
)

        $message .= "File \"$name\" could not be
uploaded.\n<pre>";

    else{

        $message .= "File \"$name\" was successfully
uploaded.\n<pre>" .
            "    New name: $newname\n" .
            "    File size: " . number_format($size) . "\n" .
            "    File type: $type\n</pre><br>\n";
    }
    $i++;
    if (!($fp = fopen("counter.txt","w+")))
    {
        $message .= "<b>Error!!</b><br><br>\nCannot write the counter
file.\n";
        return (0);
    }
    fwrite($fp,$i);
    fclose($fp);

    return (1);
}

/* This one prepends backslashes (\) before
* any dangerous user input, like quotes.
*/
function fix($string)
{
    if (!get_magic_quotes_gpc()) /* Off */
        return ( addslashes($string) );
    else
        return ($string);
}

/*
* Main code
*/

$BibtexKey = fix($_POST["Bibkey"]);
$title     = fix($_POST["Title"]);
$Authors   = fix($_POST["Authors"]);
$Journal   = fix($_POST["Journal"]);
$Volume    = fix($_POST["Volume"]);
$Number    = fix($_POST["Number"]);
$Conference = fix($_POST["Conference"]);
$Address    = fix($_POST["Address"]);
$Booktitle = fix($_POST["Booktitle"]);
$Editors    = fix($_POST["Editors"]);
$Publisher  = fix($_POST["Publisher"]);
$Month      = fix($_POST["Month"]);
$Year       = fix($_POST["Year"]);

```



```

$Info      = fix($_POST["Info"]);
$Pages     = fix($_POST["Pages"]);
$Keywords  = fix($_POST["Keywords"]);

$ftype1    = fix($_POST["Type1"]);
$ftype2    = fix($_POST["Type2"]);
$ftype3    = fix($_POST["Type3"]);
$ftype4    = fix($_POST["Type4"]);
$ftype5    = fix($_POST["Type5"]);

$fname4    = fix($_POST["File4"]);
$fname5    = fix($_POST["File5"]);

if (!$Authors)
    die("At least the author field should be filled!\n");

/* Upload the files first */
$fname1 = ""; $fname2 = ""; $fname3 = "";
if (!(upload_file("File1", $ftype1, $fname1) &&
    upload_file("File2", $ftype2, $fname2) &&
    upload_file("File3", $ftype3, $fname3)))
    die("$message\n");

/* Connect to DB */
$mid = mysql_connect("agatha.cs.uoi.gr", $username, $password)
    or die("Unable to connect to database server!");
@mysql_select_db($database,$mid)
    or die( "Unable to select required database!");

/* Start the page */
html_start_page("PPG - Added publication", "internal.css", 5,
"insert.php");
html_start_content("Info");
echo "$message\n";      // From the file uploads

/* Form the query
*/
$pubtype = $_POST["pubtype"];
$query = "INSERT into $tablename VALUES('','$BibtexKey','$pubtype','";

switch ($pubtype)
{
    case "Journal article":
        $query .= ( "'$Title','$Authors','$Journal','$Volume','$Number',"
        ."'','','','','$Month','$Year','$Pages','',''$Keywords'," );
        break;
    case "Conference article":
        $query .= (
        "'$Title','$Authors','','','$Conference','$Address',"
        ."'','','','$Month','$Year','$Pages','',''$Keywords',"
        );
        break;
    case "Technical report":
        $query .= ( "'$Title','$Authors','','','$Number','',''$Address',"
        ."'','','','$Month','$Year','','','$Keywords'," );
        break;
    case "Book chapter":

```

```

    $query .= (
    "$Title','$Authors','','','','','$Booktitle','$Editors',"
        ."'$Publisher','','$Year','$Pages','','$Keywords'," );
    break;
    case "Book":
    $query .= ( "'','$Authors','','','','','$Booktitle','',"
        ."'$Publisher','','$Year','','$Keywords'," );
    break;
    case "Thesis":
    $query .= (
    "'','$Authors','','','$Number','','','$Address','$Booktitle',"
        ."'','$Month','$Year','','$Info','$Keywords'," );
    break;
    case "Other":
    default:
    $query .= (
    "$Title','$Authors','','','','','','','','','','',"
        ."'$Info','$Keywords'," );
    break;
    }

    $query .= ( "'$fname1','$fname2','$fname3','$fname4','$fname5',"
        ."'$ftype1','$ftype2','$ftype3','$ftype4','$ftype5'" );

    if (!mysql_query($query))
    {
        echo( "<b>Error inserting the publication in the
        database:</b>\n<br>\n"
            . mysql_error() . "\n");
        mysql_close();
        html_end_content();
        html_end_page();
        die();
    }

    ?>

    <h4>Publication added successfully</h4>
    <br><br>

    You will be redirected to <a href="insert.php"><b>add another
    publication</b></a> in 5 seconds.

    <br><br>

    You may also go <a href="index.php"><b>back to main administration
    page.</b></a>

    <?php

    mysql_close();

    html_end_content();
    html_end_page();

    ?>

```



```

        or die( "Unable to select required database!");

    /* Start the page */
    html_start_page("PPG - Modify/Delete Publications", "internal.css",
0, "");
    html_start_content("List of publications to modify/delete");

    /* Pare ta papers se fthinousa xronologikh seira */
    $query = "SELECT * FROM $tablename order by Year desc, Month desc,
id ";
    $result = mysql_query($query);

    /* Elegxos gia 0 papers sth bash */
    if(mysql_numrows($result) == 0)
    {
        mysql_close();
        echo "\nNo publications found.\n\n";
        html_end_content();
        html_end_page();
        die();
    }

    echo "<ul>\n";

    while ($row = mysql_fetch_array($result))
    {
        echo "<li>";
        show_publication($row, $pubsurl);
        show_buttons($row["id"], $row["Type"]);
        echo "<br></li>\n";
    }

    echo "</ul>\n";
    mysql_close();

    echo "<br><form action=\"index.php\" method=\"post\">\n";
    echo "  <input type=\"submit\" name=\"back\" ";
    echo "value=\" Back to main administration page \"/>\n";
    echo "</form>\n";

    html_end_content();
    html_end_page();
} /* else */

?>

```

B.1.5 modify.php

```

<?php

/* MODIFY.PHP
 *   Code to modify a selected publication
 *
 */

include("../dbinfo.inc.php");
include("html.inc.php");

```

```

include("forms.php");

$id    = rawurldecode( $_GET['id'] );
$type  = rawurldecode( $_GET['type'] );

/* Connect to DB */
$mid = mysql_connect("agatha.cs.uoi.gr", $username, $password)
      or die("Unable to connect to database server!");
@mysql_select_db($database, $mid)
      or die("Unable to select required database!");

/* Anaktisi toy paper */
$query = "SELECT * FROM $tablename WHERE id='$id'";
$result = mysql_query($query);
if (!$result)
    die("Cannot retrieve the publication to be modified.\n");
$row = mysql_fetch_array($result);

mysql_close();

/* Start the page */
html_start_page("PPG - Modifying publication","internal.css",0,"");
html_start_content("Modifying the selected publication");

echo "<h4>Publication type: $type</h4><br>\n";
show_form($type, "dbmodify.php", $row);

echo "<br><br>";
echo "You may <a href=\"mod_del_list.php\"><b><u>&nbsp;   CANCEL  
&nbsp;   </u></b></a>";
echo " or &nbsp;   <input type=\"Submit\" value=\" Modify publication  
\n/>\n";
echo "</form>\n";

html_end_content();
html_end_page();

?>

```

B.1.6 dbmodify.php

```

<?php

/* DBMODIFY.PHP
 *   Modify a publication in the DB.
 *
 */

include("html.inc.php");
include("../dbinfo.inc.php");

$message = "";

/* Upload 1 file
 */
function upload_file($fid, &$stype, &$newname)

```

```

{
    global $pubspath; // Needed
    global $message;

    $file = $_FILES["$fid"]["name"];
    $size = $_FILES["$fid"]["size"];
    if (!$type) // Get the one from the system
        $type = $_FILES["$fid"]["type"];

    if ($file == "")
    {
        $newname = "";
        return (1); // All ok */
    }

    /* periptwsh pou einai eite adeio to arxeio pou bazw h den yparxei
    * to arxeio ayto! (oxi otan den kanw upload).
    * Den ginetai h eisagwgh alla epistrefw pisw gia na diorthwsw to
    * onoma tou arxeiou
    */
    if ($size <= 0)
    {
        $message .= "<b>Error!!</b><br><br>\nFile \"\" . basename($file) .
            \"\" does not exist or is empty.<br>\n\" .
            "Go back & check the paths of the files you want to
upload.\n";
        return (0);
    }

    //anoigw to arxeio gia na parw to metrhth
    if (!$fp = fopen("counter.txt","r"))
    {
        $message .= "<b>Error!!</b><br><br>\nCannot read the counter
file.\n";
        return (0);
    }
    $i = fread($fp, 20);
    fclose($fp);

    $name = basename($file);
    $newname = $i . $name; // Make unique name
    $uploadfile = $pubspath . $newname; // Final destination

    if (!move_uploaded_file($_FILES["$fid"]["tmp_name"], $uploadfile))
    {
        $message .= "<b>Error!!</b><br><br>\nFile \"$name\" couldn't be
moved.\n";
        return (0);
    }
    else
        $message .= "File \"$name\" was successfully uploaded.\n<pre>\" .
            " New name: $newname\n\" .
            " File size: \" . number_format($size) . "\n\" .
            " File type: $type\n</pre><br>\n";

    $i++;
    if (!$fp = fopen("counter.txt","w+"))
    {
        $message .= "<b>Error!!</b><br><br>\nCannot write the counter
file.\n";
        return (0);
    }
}

```

```

    }
    fwrite($fp,$i);
    fclose($fp);

    return (1);
}

/* This one prepends backslashes (\) before
 * any dangerous user input, like quotes.
 */
function fix($string)
{
    if (!get_magic_quotes_gpc()) /* Off */
        return ( addslashes($string) );
    else
        return ($string);
}

/*
 * Main code
 */

$BibtexKey = fix($_POST["Bibkey"]);
$title     = fix($_POST["Title"]);
$Authors   = fix($_POST["Authors"]);
$Journal   = fix($_POST["Journal"]);
$Volume    = fix($_POST["Volume"]);
$Number    = fix($_POST["Number"]);
$Conference = fix($_POST["Conference"]);
$Address    = fix($_POST["Address"]);
$Booktitle = fix($_POST["Booktitle"]);
$Editors   = fix($_POST["Editors"]);
$Publisher  = fix($_POST["Publisher"]);
$Month     = fix($_POST["Month"]);
$Year      = fix($_POST["Year"]);
$Info      = fix($_POST["Info"]);
$Pages     = fix($_POST["Pages"]);
$Keywords  = fix($_POST["Keywords"]);

$fstype1   = fix($_POST["Type1"]);
$fstype2   = fix($_POST["Type2"]);
$fstype3   = fix($_POST["Type3"]);
$fstype4   = fix($_POST["Type4"]);
$fstype5   = fix($_POST["Type5"]);

$fname4    = fix($_POST["File4"]);
$fname5    = fix($_POST["File5"]);

if (!$Authors)
    die("At least the author field should be filled!\n");

/* Upload the files first */
$fname1 = ""; $fname2 = ""; $fname3 = "";
if (!(upload_file("File1", $fstype1, $fname1) &&
    upload_file("File2", $fstype2, $fname2) &&
    upload_file("File3", $fstype3, $fname3)))
    die("$message\n");

```

```

/* Connect to DB */
$mid = mysql_connect("agatha.cs.uoi.gr", $username, $password)
      or die("Unable to connect to database server!");
@mysql_select_db($database,$mid)
      or die( "Unable to select required database!");

/* Start the page */
html_start_page("PPG - Modified publication", "internal.css",
               8, "mod_del_list.php");
html_start_content("Info");
echo "$message\n";      // From the file uploads

/* Read in the old row - I don't think this is needed!!!
*/
$pubtype = $_POST["pubtype"];
$pubid   = $_POST["pubid"];

$result = mysql_query( "SELECT * FROM $tablename where id=$pubid" );
$row     = mysql_fetch_array($result);

/* Form the query
*/
$query = "UPDATE $tablename SET Bibkey='$Bibtex_key'";

switch ($pubtype)
{
  case "Journal article":
    $query .= (
      "Title='$Title',Authors='$Authors',Journal='$Journal',"
      ."Volume='$Volume',Number='$Number',Month='$Month',"
      ."Year='$Year',Pages='$Pages',Keywords='$Keywords'" );
    break;
  case "Conference article":
    $query .= (
      "Title='$Title',Authors='$Authors',Conference='$Conference',"
      ."Address='$Address',Month='$Month',Year='$Year',"
      ."Pages='$Pages',Keywords='$Keywords'" );
    break;
  case "Technical report":
    $query .= ( "Title='$Title',Authors='$Authors',Number='$Number',"
      ."Address='$Address',Month='$Month',Year='$Year',"
      ."Keywords='$Keywords'" );
    break;
  case "Book chapter":
    $query .= (
      "Title='$Title',Authors='$Authors',Booktitle='$Booktitle',"
      ."Editors='$Editors',Publisher='$Publisher',Year='$Year',"
      ."Pages='$Pages',Keywords='$Keywords'" );
    break;
  case "Book":
    $query .= ( "Authors='$Authors',Booktitle='$Booktitle',"
      ."Publisher='$Publisher',Year='$Year',Keywords='$Keywords'" );
    break;
  case "Thesis":
    $query .= (
      "Authors='$Authors',Number='$Number',Booktitle='$Booktitle',"
      ."Address='$Address',Month='$Month',Year='$Year',"
      ."Info='$Info',Keywords='$Keywords'" );
    break;
}

```



```

    case "Other":
    default:
        $query .= ( "Title='$Title',Authors='$Authors',"
                    ."Info='$Info',Keywords='$Keywords'" );
        break;
    }

/* Don't touch the downloads unless something new was downloaded
*/
if ($fname1) $query .= ",File1='$fname1'";
if ($fname2) $query .= ",File2='$fname2'";
if ($fname3) $query .= ",File3='$fname3'";

$query .= ",File4='$fname4'";
$query .= ",File5='$fname5'";

$query .= ( ",Type1='$ftype1',Type2='$ftype2',Type3='$ftype3'"
            ."Type4='$ftype4',Type5='$ftype5'" );

$query .= " WHERE id='$pubid'";

if (!mysql_query($query))
{
    echo( "<b>Error modifying the publication in the
database:</b>\n<br>\n"
        . mysql_error() . "\n");
    mysql_close();
    html_end_content();
    html_end_page();
    die();
}

?>

<h4>Publication modified successfully.</h4>
<br><br>

You will be redirected to <a
href="mod_del_list.php"><b>modify/delete
another publication</b></a>
in 8 seconds.

<br><br>

You may also go <a href="index.php"><b>back to main administration
page.</b></a>

<?php
mysql_close();

html_end_content();
html_end_page();

?>

```



```

{
include("../dbinfo.inc.php");
include("html.inc.php");
include("../showpub.inc.php");

/* Connect to DB */
$mid = mysql_connect("agatha.cs.uoi.gr", $username, $password)
    or die("Unable to connect to database server!");
@mysql_select_db($database, $mid)
    or die( "Unable to select required database!");

$id = rawurldecode( $_GET['id'] );
$type = rawurldecode( $_GET['type'] );

/* Emfanizw ta stoixeia tou pros diagrafh paper */
$query = "SELECT * FROM $tablename WHERE id='$id'";
$result = mysql_query($query);

if (!$result)
    die("Cannot SELECT the publication to be deleted.\n");

/* Start the page */
html_start_page("PPG - Deleting publication","internal.css", 0,
"");
html_start_content("Confirming publication removal");

echo "<h4>Are you sure you want to delete the following publication
?</h4><br>\n";

show_publication( mysql_fetch_array($result), $pubsurl );
echo "<br><br>\n";

echo "<form action=\"delete.php\" method=\"post\">\n";
echo "<input type=hidden name=id value=\"\$id\"/>\n";
echo "<input type=\"submit\" name=\"DOIT\" value=\" Yes \"/>\n";
echo "&nbsp;&nbsp;&nbsp;&nbsp;<input type=\"submit\" name=\"DONTDOIT\"
value=\" No \"/>\n";
echo "</form>\n";

mysql_close();

html_end_content();
html_end_page();
}

?>

```

B.2 Ο κώδικας του διαχειριστή αρχείων

Ακολουθεί η παρουσίαση του αρχείου όπου υλοποιείται ο file manager του συστήματος.

```
<?php

/** PPG FileManager
 * A simple file manager for the Parallel Processing Group's
 * site.
 * By Lambrini Konsta and V.V. Dimakopoulos.
 */

/* * * * * *
 *
 * CONFIGURATION
 *
 * * * * * */

$UploadLimit = "524288"; // max upload file size in bytes
(0.5MBytes)
$DirsizeLimit = "134217728"; // max total for directory (128MBytes)
$MkdirAllowed = 1; // allow creation of directories
$MkfileAllowed = 1; // allow creation of files (including
Uploads)
$rootname = "demo"; // The 'nickname' for the root dir

$fmcss = "fmstyle.css"; // The stylesheet
$fmicondir = "fmicons"; // Where the icons are kept

$defaultSortby = "Name"; // "Name", "Size", "Date", "Type"
$defaultColumns = 3; // # columns for file list
$defaultDetails = FALSE; // Don't show file details

/* * * * * *
 *
 * DISCOVER PARAMETERS
 *
 * * * * * */

/* If we are on www.site.com/xx/yy/filemanager.php, then
MyRelativeURL
 * is "/xx/yy/filemanager.php", while MyRelativePath is "/xx/yy/".
 */
$MyRelativeURL = $_SERVER['PHP_SELF'];
// Installed dir only - relative path.
$MyRelativePath = dirname($MyRelativeURL) . '/';

/* Now get real path info. ScriptName is the name of this file.
 * RootPath is the actual path in the server's file system where this
file is.
 */
$ScriptName = basename(__FILE__);
```

```

$RootPath = str_replace($ScriptName,"",__FILE__);

/* This is the current (working) directory. It is relative to the
root
* directory (i.e. relative to where this script is located). If we
are
* at the root, wdir = "", i.e. it does not contain any slashes,
otherwise
* it always ends in a slash.
*/
$wdir = array_key_exists("wdir", $_REQUEST) ? $_REQUEST["wdir"] : "";
if (ereg("\\.\\.\/", $wdir)) // Don't allow attempts to go above the
root!
{
    $wdir="";
    errorPage("You cannot rename files or directories above the root
directory!");
}

/* Display options
*/
$sort    = array_key_exists("sort", $_REQUEST) ? $_REQUEST["sort"]
          : $defaultSortby;
$cols    = array_key_exists("cols", $_REQUEST) ? $_REQUEST["cols"]
          : $defaultColumns;
$details = array_key_exists("details", $_REQUEST) ?
$_REQUEST["details"]
          : $defaultDetails;

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*             MAIN CODE
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

// Add characters to strip out of filenames
$snr = array("%","'","+","\\","/","#","..","!",'"',' ','?','*','~');

/* A status message to be displayed if needed
*/
$msg = "";

/*
* Basic control: what we are supposed to do
*/
$action = array_key_exists("action", $_REQUEST) ? $_REQUEST["action"]
: "";
$which  = array_key_exists("which", $_REQUEST) ? $_REQUEST["which"] :
"";
$which  = trim($which);

if ($action)
    switch ($action)
    {
        case "back":
            $wdir = substr($wdir, 0, -1);
            $slashpos = strrpos($wdir, "/");

```

```

        $wdir = ($slashpos == 0) ? "" : substr($wdir, 0,
($slashpos+1));
        break;

        case "upload":
            $f = array_key_exists("upldfile", $_FILES) ?
$_FILES['upldfile'] : "";
            if (!$f)
                $msg = "No file selected to upload!";
            else
                $msg = uploadFile($f);
            break;

        /* Create new file/folder */
        case "Folder":
            if (!$which)
                $msg = "No name given for the new folder!";
            else
                $msg = createFolder($which);
            break;
        case "Text file":
            if (!$which)
                $msg = "No name given for the new file!";
            else
                newfilePage($which);
            break;
        case "Save":
            $msg = saveNewFile();
            break;

        /* Rename file/folder */
        case "rename":
            if (ereg("\.\./", $which)) // Don't allow attempts to go
above the root!
                errorPage("You cannot rename files or directories above the
root directory!");
            renamePage($which);
            break;
        case "Rename it":
            $from = array_key_exists("from", $_POST) ? $_POST["from"] :
"";
            $to = array_key_exists("to", $_POST) ? $_POST["to"] : "";
            $msg = renameFile($from, $to);
            break;

        /* Edit file */
        case "edit":
            if (ereg("\.\./", $which)) // Don't allow attempts to go
above the root!
                errorPage("You cannot edit files that belong to directories
above the root directory!");
            editPage($which);
            break;
        case "Save changes":
            $msg = saveEditedFile();
            break;

        /* Delete files/folders */
        case "delone":
            if (ereg("\.\./", $which)) // Don't allow attempts to go
above the root!

```

```

        errorPage("You cannot delete files or directories above the
root directory!");
        deletePage( array($which) );
        break;
    case "Delete checked files":
        $cf = getCheckedFiles();
        if (count($cf) == 0)
            $msg = "No files checked!";
        else
            deletePage($cf);
        break;
    case "Delete":
        $msg = deleteAllFiles();
        break;

    /* Copy/move files/folders */
    case "cmone":
        if (ereg("\.\./", $which)) // Don't allow attempts to go
above the root!
            errorPage("You cannot copy or move files or directories above
the root directory!");
            copymovePage( array($which) );
            break;
    case "Copy/Move checked files":
        $cf = getCheckedFiles();
        if (count($cf) == 0)
            $msg = "No files checked!";
        else
            copymovePage($cf);
        break;
    case "Do it!": /* Comes from the copymovePage() */
        $msg = copymoveAllFiles();
        break;

};

/*
 * If we arrive here, we must display the main window, i.e. the file
list.
 */

$FullPath = substr($RootPath.$wdir, 0, -1); // remove the slash from
the path
if (readDirectory($FullPath, $folder_table, $file_table))
    errorPage("Cannot display files in the requested directory.");

htmlDrawPage();
flush();
exit;

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *
 *          FUNCTIONS
 *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

/* Check if Dir has MakDir permissions */
function CreateDirOK($chkpath)
{

```

```

global $fmicondir;

$NoCreateDirs = array("docs/", "$fmicondir/");
foreach($NoCreateDirs as $name)
    if($chkpath == $name)
        return (FALSE);
return (TRUE);
}

/* Check if Dir has Upload permissions */
function UpPathOK($chkpath)
{
    global $fmicondir;

    $NoUploadDirs = array("docs/", "$fmicondir/");
    foreach($NoUploadDirs as $name)
        if($chkpath == $name)
            return (FALSE);
    return (TRUE);
}

function IsHidden($file)
{
    global $ScriptName, $fmcss, $fmicondir;

    $HiddenFiles = array("index.php", ".htaccess", "$fmcss",
        "$fmicondir", $ScriptName);
    foreach($HiddenFiles as $name)
        if ($file == $name)
            return (TRUE);
    return (FALSE);
}

function Ok2Edit($fileext)
{
    $fileext = strtolower($fileext);
    $EditExtensions = array("htm","html","txt","php","css","c","h");
    foreach ($EditExtensions as $name)
        if ($fileext == $name)
            return (TRUE);
    return (FALSE);
}

function TargetOK($schck)
{
    global $ScriptName, $fmcss, $fmicondir;

    $ModifyBlock = array("index.php", ".htaccess", "$fmcss",
        "$fmicondir", $ScriptName);
    foreach($ModifyBlock as $name)
        if ($schck == $name)
            return (FALSE);
    return (TRUE);
}

```



```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*          UTILITIES
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

/* calculate the size of files in $dir */
function dirsiz($dir)
{
    $size = -1;
    if ($dh = @opendir($dir))
    {
        while (($file = readdir($dh)) != false)
        {
            if ($file != "." and $file != "..")
            {
                $path = $dir."/".$file;
                if (is_dir($path))
                    $size += dirsiz("$path/");
                elseif (is_file($path))
                    $size += filesize($path);
            }
        }
        closedir($dh);
    }

    return $size;
}

function sizestring($size)
{
    $kb = 1024;           // Kilobyte
    $mb = 1024 * $kb;    // Megabyte
    $gb = 1024 * $mb;    // Gigabyte
    $tb = 1024 * $gb;    // Terabyte
    if ($size < $kb)
        return $size." B";
    elseif ($size < $mb)
        return round($size/$kb,2)." KB";
    elseif ($size < $gb)
        return round($size/$mb,2)." MB";
    elseif ($size < $tb)
        return round($size/$gb,2)." GB";
    else
        return round($size/$tb,2)." TB";
}

/* Recursively find all (sub)directories under $dir
*/
function alldirs($dir)
{
    global $wdir;

    if ( ($hndl = opendir($dir)) <= 0 )
        return ( array() );

    for ($list = array(0 => "$dir"); ($file = readdir($hndl)); )
    {

```

```

        if ($file=='.' || $file=='..' || is_link($dir.$file) ||
            IsHidden($file) || !UpPathOK($file.'/'))
            continue;
        if (is_dir($dir.$file))
            $list = array_merge($list, alldirs($dir.$file."/"));
    }

    closedir($hndl);
    return ($list);
}

/* Find all the files that have been checked; store them in
CheckedFiles[].
*/
function getCheckedFiles()
{
    global $RootPath, $wdir;

    $me = basename(__FILE__); // get the file name

    $FullPath = substr($RootPath.$wdir, 0, -1); // remove the slash
    if ( !($dir = @opendir($FullPath)) )
        return ( array() );

    $fc = 0;
    while ($chk_file = readdir($dir))
    {
        /* Count all files except hidden, .., .. and ourselves. */
        $match = 1;
        if ($chk_file != "." && $chk_file != ".." && $chk_file != "$me")
            $match = 0;
        if (IsHidden($chk_file))
            $match = 1;

        if (!$match)
            $fc++;
    }

    if ($fc == 0)
        return ( array() );

    for($i = $j = 0; $i < $fc; $i++)
        if ( array_key_exists("id_$i", $_POST) )
        {
            $CheckedFiles[$j] = $_POST["id_$i"];
            $j++;
        };
    return ($CheckedFiles);
}

function uploadFile($file)
{
    global $snr, $UploadLimit, $DirsizeLimit, $RootPath, $wdir,
    $MyRelativePath;

    $upfile = strip_tags ($file['name']);
    $upfile = str_replace($snr, "", $upfile); // remove any % signs
    from the file name
    $upfile = trim($upfile);

```

```

if ($upfile == "")
    return ( "No file selected to upload!" );

/* if the file size is within allowed limits */
if ($file['size'] > 0 && $file['size'] < $UploadLimit)
{
    /* if adding the file will not exceed the maximum allowed total
*/
    if ((dirsize($RootPath)+ $file['size']) < $DirsizeLimit)
    {
        if (TargetOK($upfile)==1)
        {
            /* put the file in the directory */
            if (move_uploaded_file($file['tmp_name'],
$RootPath.$wdir.$upfile))
                $msg = "File '$upfile' was successfully uploaded.";
            else
                $msg = "Could not upload file '$upfile' to
$MyRelativePath.$wdir.";
        }
        else
            $msg = "The filename '$upfile' is BLOCKED from being uploaded
here.";
        }
        else
            $msg = "There is not enough free space and the file
could<br>not be uploaded.";
        }
        else
            $msg = "The file was greater than the maximum allowed<br>
file size of " . sizestring($UploadLimit) .
" and could not be uploaded.";
    return ($msg);
}

```

```

function errorPage($message)
{
    global $wdir,$MyRelativeURL;

    htmlStart();
    $opts = options2form();

    echo <<<content

    <center>
    <br><br>
    <form method="post" action="$MyRelativeURL">
    <input type="hidden" name=wdir value="$wdir">
    $opts

    <table cellpadding=5 cellspacing=0 border=0
        style="border: solid 1px #000000;">
        <tr bgcolor="#BBBBBB">
            <td align="center"
                class="dialoghead" style="border-bottom: solid 1px
#000000;">
                <b>ERROR</b></td></tr>
            <tr bgcolor="#DDDDDD">
                <td align="center">
                    <table cellspacing=5 cellpadding=5 border=0>

```

```

        <tr>
            <td class=dialog>$message</td></tr>
        </table></td></tr>
        <tr bgcolor="#DDDDDD">
            <td align="center">
                <input class=button type="submit" name="cancel" value="Return
back">
            </td></tr>
        </table>
    </form>
</center>
</body>
</html>
content;

flush();
exit;
}

```

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *
 *          CREATE/EDIT A FILE
 *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

```

```

function newfilePage($file)
{
    global $RootPath,$wdir,$rootname,$MyRelativeURL,$snr;

    $file = str_replace($snr, "", $file);
    if (TargetOK($file) && UpPathOK($wdir))
    {
        if ( file_exists($RootPath.$wdir.$file) )
            errorPage("This file already exists!");
    }
    else
        errorPage("Cannot create this file here.");

    /* Standard html stuff */
    htmlStart();
    $opts = options2form();

    echo <<<content
    <center>
    <br><br>
    <form method="post" action="$MyRelativeURL">
    <input type="hidden" name="fname" value="$file">
    <input type="hidden" name=wdir value="$wdir">
    $opts

    <table cellpadding=5 cellspacing=0 border=0
        style="border: solid 1px #000000;">
        <tr bgcolor="#BBBBBB">
            <td align="center"
                class="dialoghead" style="border-bottom: solid 1px
#000000;">
                <b>NEW TEXT FILE</b></td></tr>
        <tr bgcolor="#DDDDDD">
            <td align="center" class="dialoghead">

```



```

        if ( file_exists($RootPath.$wdir.$fname) )
            return ("File '$fname' already exists -- cannot re-create
it!");
    }
    else
        return ("Cannot create '$fname' here.");

$content = stripslashes( $_REQUEST["newcontent"] );
$fp = @fopen($RootPath.$wdir.$fname, "w");
if ($fp)
{
    if ($content)
        fwrite($fp, $content);
    fclose($fp);
    $msg = "File created ok.";
}
else
    $msg = "Could not create file.";
return ($msg);
}

```

```

function editPage($file)
{
    global $RootPath,$wdir,$rootname,$MyRelativeURL;

    /* Standard html stuff */
    htmlStart();
    $opts = options2form();

    echo <<<content

    <center>
    <br><br>
    <form method="post" action="$MyRelativeURL">
    <input type="hidden" name=wdir value="$wdir">
    $opts

    <table cellpadding=5 cellspacing=0 border=0
        style="border: solid 1px #000000;">
        <tr bgcolor="#BBBBBB">
            <td align="center"
                class="dialoghead" style="border-bottom: solid 1px
#000000;">
                <b>EDITING</b></td></tr>
content;

    /* Open file */
    $fp = @fopen($RootPath.$wdir.$file, "r");
    if (!$fp)
    {
        echo <<<failed
        <tr bgcolor="#DDDDDD">
            <td align="center">
                <table cellspacing=10 cellpadding=10 border=0>
                    <tr><td align="center">
                        Could not open file '$file' for editing.</td></tr>
                    <tr><td align="center">

```



```
global $RootPath, $wdir;

if ( !array_key_exists("newcontent", $_REQUEST) ||
    !array_key_exists("fname", $_REQUEST) )
    errorPage("Something is wrong;<br>Invalid file name or non-existent content.");

$content = stripslashes( $_REQUEST["newcontent"] );
$fp = @fopen($RootPath.$wdir.$_REQUEST["fname"], "w");
if ($fp)
{
    fwrite($fp, $content);
    fclose($fp);
    $msg = "File updated.";
}
else
    $msg = "Could not save edited file.";
return ($msg);
}

function createFolder($name)
{
    global $snr, $RootPath, $wdir, $MkdirAllowed;

    if (!$MkdirAllowed)
        return ("Not allowed to create a folder here!");

    $dirname = strip_tags ($name);
    $dirname = str_replace($snr, "", $dirname);
    $dirname = trim($dirname);
    if (TargetOK($dirname))
    {
        if ( mkdir($RootPath.$wdir.$dirname) ) // 0777 by default
            $msg = "Created new folder '$dirname'";
        else
            $msg = "Could not create folder '$dirname'";
    }
    else
        $msg = "Cannot create a folder with this name ('$dirname')";
    return ($msg);
}

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                                                                                            *
*                                                                                                                                            *
*                                     RENAME A FILE                                                                                              *
*                                                                                                                                            *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

function options2url()
{
    global $sort, $cols, $details;

    return ( "&sort=$sort&cols=$cols&details=$details" );
}

function options2form()
{
```



```

    </html>
content;

    flush();
    exit;
}

function renameFile($old, $new)
{
    global $snr, $RootPath, $wdir;

    if (!($old && $new))
        errorPage("Something is wrong.<br>Invalid name used for
renaming.");

    $new = trim( str_replace( $snr, "", strip_tags($new) ) );
    if ($old == $new)
        return ("Cannot rename a file to itself!");

    if (TargetOK($old))    // Modify this file okay?
        $go1=1;
    if (TargetOK($new))    // Is New name Okay?
        $go2=1;
    if ($go1+$go2==2)
    {
        if ( @rename($RootPath.$wdir.$old, $RootPath.$wdir.$new) < 1 )
            $msg = "Could not rename file '$old' to '$new'.";
        else
            $msg = "File '$old' renamed to '$new'.";
    }
    else
    {
        if ($go1==0)
            $msg = "File '$old' cannot be renamed.";
        if ($go2==0)
            $msg = "Cannot use the name '$new'.";
    }
    return ($msg);
}

/* * * * * *
*
*          DELETE FILES
*
* * * * * */

function deletePage($chkfiles)
{
    global $RootPath, $wdir, $MyRelativeURL, $msg;

    if (count($chkfiles) == 0)
        errorPage("<p>There are no files to delete!");

    htmlStart();
    $opts = options2form();
    $c = count($chkfiles);

    echo <<<delp1

```



```

if (is_file( $fullpath ) || is_link($fullpath))
    $msg = unlink($fullpath) ?
        "File <b>$MyRelativePath$pathex$file </b> deleted." :
        "Unable to delete file <b>$MyRelativePath$pathex$file
</b>";
elseif (is_dir($fullpath))
{
    $handle = opendir($fullpath);    /* Recurse */
    $file2 = $file;
    while ($aux = readdir($handle))
    {
        if ($aux != "." && $aux != "..")
        {
            $file = $file2."/".$aux;
            deleteFile($fullpath."/".$aux,$pathex,$file);
        }
    }
    closedir($handle);
    $msg .= rmdir($fullpath) ?
        "Directory <b>$MyRelativePath$pathex$file2 </b>
deleted." :
        "Unable to delete directory
<b>$MyRelativePath$pathex$file2 </b>";
}
}

function deleteAllFiles()
{
    global $RootPath, $wdir;

    $msg = "";

    $FullPath = substr($RootPath.$wdir, 0, -1); // remove the slash
    if ($dir = @opendir($FullPath)) // Make sure we can do this
    there
    {
        $count = $_POST['fnum'];
        $files = $_POST['box'];
        if (!$files || !$count || $count <= 0)
            return ("Nothing to delete.");

        for($j = 0; $j < $count; $j++)
        {
            if (TargetOK($files[$j]))
                deleteFile($RootPath.$wdir.$files[$j], $wdir, $files[$j]);
            else
                $msg .= "File '$wdir.$files[$j]'" cannot be deleted.<br>";
        }
        if ($msg == "")
            $msg = "All files deleted ok.";

        closedir($dir);
    }
    else
        $msg = "Cannot delete files in this directory.";
    return ($msg);
}

```



```

{
    global $RootPath,$wdir,$MyRelativePath;

    /* Check for overwrites! */
    $new = $orig;
    while ( file_exists($RootPath.$dest.$new) )
        $new = "copy_" . $new;

    if ( is_file($RootPath.$wdir.$orig) ||
is_link($RootPath.$wdir.$orig) )
    {
        /* Simple file
        */
        if ( !@copy($RootPath.$wdir.$orig, $RootPath.$dest.$new) )
            return ("Cannot perform the copy/move.<br>");

        $chmodperms=fileperms($RootPath.$wdir.$orig);
        if ( !chmod($RootPath.$dest.$new,$chmodperms) ) //copy with the
existing file permissions
            return ("Cannot change the file's permissions.<br>");

        if ($action == 'move')
            if ( !@unlink($RootPath.$wdir.$orig) )
            {
                if ( !@unlink($RootPath.$dest.$new) ) /* Try to remove the
new file */
                    return ("Could not move file '$orig'.<br>&nbsp;&nbsp; " .
                        "I made a copy to the destination folder,
though!");
                else
                    return ("Could not move file '$orig'.<br>&nbsp;&nbsp; " .
                        "Try copying it -- folder permissions seem to allow
it.");
            }

            if ($new != $orig)
                return ( (($action == "move") ? "Moved ":"Copied ") .
                    "$orig as $MyRelativePath$dest<u>$new</u>\n" .
                    "(so as to avoid overwriting an existing file).<br>"
                );
            return ( "$orig" . (($action == "move") ? " moved " : " copied ")
                .
                    "to $MyRelativePath$dest<br>" );
        }
    elseif ( is_dir( $RootPath.$wdir.$orig ) )
    {
        /* Directory
        */

        //Avoid the recursive copy/move
        if ( strrpos($dest, $wdir.$orig) == TRUE)
            return ("Cannot do this recursive copy/move!<br>\n");

        if (!@mkdir($RootPath.$dest.$new, 0777))
            return ("Cannot create directory '$RootPath$dest$new'.<br>\n");

        if ( !($handle = @opendir($RootPath.$wdir.$orig) ) )
            return ("Cannot read directory '$RootPath$wdir$orig'.<br>\n");

        while ($aux = @readdir($handle))
        {

```

```

    if ($aux != "." && $aux != "..")
        copymoveFile($orig."/".$aux, $dest, $action);
}

closedir($handle);
if ($action == 'move')
    @rmdir($RootPath.$wdir.$orig);

//Change the new directory's permissions
$chmodperm=fileperms($RootPath.$wdir.$orig);
if ( !chmod($RootPath.$dest.$new,$chmodperm) //copy with the
existing directory permissions
    return ("Cannot copy with the existing directory
permissions.<br>");

if ($new != $orig)
    return ( "Directory $MyRelativePath$wdir$orig " .
        (($action == "move") ? " moved":" copied")." to
$MyRelativePath$dest<u>$new</u>\n" .
        "(so as to avoid overwriting an existing
directory).<br>" );

    return ( "Directory $MyRelativePath$wdir$orig " .
        (($action == "move") ? "moved":"copied")." to
$MyRelativePath$dest<br>" );
        echo "<br>";

}

}

/* * * * * *
*
*          GET FILE PERMISSIONS
*
* * * * * */

```

```

function getFilePerms( $in_Perms ) {

    $perms= '';

    if($in_Perms & 0x1000)    // FIFO pipe
        $perms = 'p';
    elseif($in_Perms & 0x2000) // Character special
        $perms = 'c';
    elseif($in_Perms & 0x4000) // Directory
        $perms = 'd';
    elseif($in_Perms & 0x6000) // Block special
        $perms = 'b';
    elseif($in_Perms & 0x8000) // Regular
        $perms = '&minus;';
    elseif($in_Perms & 0xA000) // Symbolic Link
        $perms = 'l';
    elseif($in_Perms & 0xC000) // Socket
        $perms = 's';
    else
        $perms = 'u';
}

```



```

// owner
$perms .= (($in_Perms & 0x0100) ? 'r' : '&minus;') .
          (($in_Perms & 0x0080) ? 'w' : '&minus;') .
          (($in_Perms & 0x0040) ? (($in_Perms & 0x0800) ? 's' : 'x'
) :
          (($in_Perms & 0x0800) ? 'S' :
'&minus;'));

// group
$perms .= (($in_Perms & 0x0020) ? 'r' : '&minus;') .
          (($in_Perms & 0x0010) ? 'w' : '&minus;') .
          (($in_Perms & 0x0008) ? (($in_Perms & 0x0400) ? 's' : 'x'
) :
          (($in_Perms & 0x0400) ? 'S' :
'&minus;'));

// others
$perms .= (($in_Perms & 0x0004) ? 'r' : '&minus;') .
          (($in_Perms & 0x0002) ? 'w' : '&minus;') .
          (($in_Perms & 0x0001) ? (($in_Perms & 0x0200) ? 't' : 'x'
) :
          (($in_Perms & 0x0200) ? 'T' :
'&minus;'));
return $perms;
}

```

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*          FILE/FOLDER LIST
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

```

```

/* Determine file types from their extension
*/

```

```

function getfiletype($file)
{
    $file = strtolower($file);
    $dotpos = strrpos($file, ".");
    if ($dotpos < 1)
        return "Unknown";
    else
        $ext = substr($file, $dotpos+1);

    if ($ext=="html" || $ext == "htm")
        return "HTML";
    elseif ($ext=="css")
        return "CSS";
    elseif ($ext=="asp")
        return "ASP";
    elseif ($ext=="php")
        return "PHP";
    elseif ($ext=="jpg" || $ext == "jpeg")
        return "JPG";
    elseif ($ext=="bmp")
        return "BMP";
    elseif ($ext=="gif")
        return "GIF";
}

```

```

elseif ($ext=="png")
    return "PNG";
elseif ($ext=="txt")
    return "TXT";
elseif ($ext=="pdf")
    return "PDF";
elseif ($ext=="ps")
    return "PS";
elseif ($ext=="eps")
    return "EPS";
elseif ($ext=="zip")
    return "ZIP";
elseif ($ext=="rar")
    return "RAR";
elseif ($ext=="tar")
    return "TAR";
elseif ($ext=="gz")
    return "GZ";
else
    return "Other";
}

/* From the file type, determine which icon should be used
*/
function fileIcon($ftype)
{
    if ($ftype=="Folder")
        $img = "folder.gif";
    elseif ($ftype=="HTML" || $ftype=="PHP" || $ftype=="CSS" ||
$ftype=="ASP")
        $img = "file_web.gif";
    elseif ($ftype=="BMP" || $ftype=="JPG" || $ftype=="GIF" ||
$ftype=="PNG")
        $img = "file_img.gif";
    elseif ($ftype=="TXT")
        $img = "file_txt.gif";
    elseif ($ftype=="PDF")
        $img = "file_pdf.gif";
    elseif ($ftype=="PS" || $ftype=="EPS")
        $img = "file_ps.gif";
    elseif ($ftype=="ZIP" || $ftype=="RAR" || $ftype=="GZ" ||
$ftype=="TAR")
        $img = "file_zip.gif";
    else
        $img = "file_other.gif";
    return $img;
}

/* Forms a link to a file
*/
function fileLink($file,$ftype,$fsize,$fdate)
{
    global $wdir,$urlpath,$MyRelativeURL,$details;

    if ($ftype == "Folder")
        return ( "<a href=\"\$MyRelativeURL?\" . options2url() .
                \"&wdir=$wdir$file/\\" class=\"file\">$file</a>" );

    $fl = "<a href=\"\$urlpath$wdir$file\" class=\"file\">$file</a>";

```

```

    if ($details)
        $fl .= "<br><font class=\"details\">($fdate -- $fsize)</font>";
    return ($fl);
}

/* Sorting routine
*/
function myCompare($arrayA, $arrayB)
{
    global $sort;

    if ($sort=="Name")
        $sortKey=0;
    elseif ($sort=="Size")
        $sortKey=1;
    elseif ($sort=="Date")
        $sortKey=2;
    else if ($sort=="Type")
        $sortKey=3;

    $a = strtolower($arrayA[$sortKey]);
    $b = strtolower($arrayB[$sortKey]);

    if ($a == $b)
    {
        if ($sortKey != 0) /* Force a second key: the name */
        {
            $a = strtolower($arrayA[0]);
            $b = strtolower($arrayB[0]);
            if ($a != $b)
                return ( $a < $b ? -1 : 1 );
        }
        return 0;
    }
    return ( $a < $b ? -1 : 1 );
}

/* Gets a list of folders and files in the given directory.
* The lists are sorted.
*/
function readDirectory($path, &$folder_table, &$file_table)
{
    global $ScriptName;

    if ( !($dir = @opendir($path)) )
        return (1);

    $foc = 0; /* Counters for folders and files */
    $fic = 0;
    while ($file = @readdir($dir))
    {
        $match=1;
        if ($file != "." && $file != ".." && $file != "$ScriptName")
            $match = 0;
        if (IsHidden($file))
            $match=1;

        if (!$match)
        {

```

```

        $filedata = stat($path.'/'.$file); // get some info about the
file
        if (is_dir($path.'/'.$file))      // it's a folder
        {
            $folder_table[$foc][0] = $file;
            $folder_table[$foc][1] = $filedata[7]; // size in bytes
            $folder_table[$foc][2] = $filedata[9]; // time of last
modification
            $folder_table[$foc][3] = "Folder";
            $folder_table[$foc][5] = date("m/d/Y", $filedata[9]);
            $foc++;
        }
        else
        {
            $file_table[$fic][0] = $file;
            $file_table[$fic][1] = $filedata[7]; // size in bytes
            $file_table[$fic][2] = $filedata[9]; // time of last
modification
            $file_table[$fic][3] = getfiletype($file);
            $file_table[$fic][5] = date("m/d/Y", $filedata[9]);
            $fic++;
        }
    }
}

if ($foc > 0)
    usort($folder_table, "myCompare" );
if ($fic > 0)
    usort($file_table, "myCompare" );

return (0);
}

function displayFiles($folder_table, $file_table)
{
    global $cols, $MyRelativeURL, $MyRelativePath, $RootPath, $wdir,
    $details, $fmicondir;

    echo "<table width=100% cellpadding=0 cellspacing=2 " .
        "border=0 bgcolor=#DDDDDD">";

    $RenameCode = "<img src=\"$fmicondir/rename.gif\" alt=\"Rename\"
valign=middle border=0>";
    $EditCode = "<img src=\"$fmicondir/edit.gif\" alt=\"Edit\"
valign=middle border=0>";
    $DeleteCode = "<img src=\"$fmicondir/delete.gif\" alt=\"Delete\"
valign=middle border=0>";
    $CopymoveCode = "<img src=\"$fmicondir/copy.gif\" alt=\"Copy/Move\"
valign=middle border=0>";

    $foc = count($folder_table);
    $fic = count($file_table);
    if ($foc + $fic == 0)
    {
        echo "<tr><td align=center><br><span class=\"file\">\" .
            \"No files found.</span><br><br></td></tr></table>";
        return;
    }
}

```

```

/* Special care here */
if ($cols == 1)
{
    echo <<<df1
    <tr class="file">
        <td colspan=2 align=center style="border-bottom: solid 1px
#CCCCCC;"><b>File</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Permissions</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Rename</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Edit</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Delete</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Copy/move</td>
        <td align=center style="border-bottom: solid 1px
#CCCCCC;"><b>Check</td>
    </tr>

df1;
}

$cells=0;
if ($cols=="") /* Default to 3 columns */
    $cols=3;
$width = round(100/$cols);

/* Folders first
*/
for ($i = 0; $i < $foc; $i++)
{
    $fname=$folder_table[$i][0];
    $ftype=$folder_table[$i][3];
    $fdate=$folder_table[$i][5];

    $path=$RootPath.$wdir.$fname;
    $chmodlink = getFilePerms(fileperms($path));
    $chmodlink="<font face='Verdana, Arial, Helvetica'
size='1'>$chmodlink</font>";
    $renamelink = (TargetOK($fname)) ?
        "<a
href=\"\$MyRelativeURL?action=rename&which=$fname\" .
        \"&wdir=$wdir\" . options2url() .
        \">$RenameCode</a>\"
        : \"\";
    $deletelink = (TargetOK($fname)) ?
        "<a
href=\"\$MyRelativeURL?action=delone&which=$fname\" .
        \"&wdir=$wdir\" . options2url() .
        \">$DeleteCode</a>\"
        : \"\";
    $cpmvlink = "<a href=\"\$MyRelativeURL?action=cmone&which=$fname\"
.
        \"&wdir=$wdir\" . options2url() . \">$CopymoveCode</a>\";

    $fsize=sizestring($folder_table[$i][1]);

```

```

$fileicon = fileIcon($ftype);
$filelink = fileLink($fname,$ftype,$fsize,$fdate);

if ($cells == 0)
    echo "<tr>\n";
if ($cols != 1)
    echo "<td width=$width%>\n";

showFile($fileicon,$filelink,$i,$fname,$chmodlink,$renamelink,"", $del
etelink,$cpmvlink);
if ($cols != 1)
    echo "</td>\n";
$cells++;
if ($cells % $cols == 0 && $cols != 1)
{
    $cells=0;
    echo "</tr>\n";
}
}

/* Files next
*/
for($i = 0; $i < $fic; $i++)
{
    $fname=$file_table[$i][0];
    $ftype=$file_table[$i][3];
    $fdate=$file_table[$i][5];

    $path=$RootPath.$wdir.$fname;
    $chmodlink = getFilePerms(fileperms($path));
    $chmodlink="<font face='Verdana, Arial, Helvetica'
size='1'>$chmodlink</font>";
    $renamelink = (TargetOK($fname)) ?
        "<a
href=\"\$MyRelativeURL?action=rename&which=$fname\" .
        \"&wdir=$wdir\" . options2url() .
        \">$RenameCode</a>\"
        : \"\";
    $deletelink = (TargetOK($fname)) ?
        "<a
href=\"\$MyRelativeURL?action=delone&which=$fname\" .
        \"&wdir=$wdir\" . options2url() . \">$DeleteCode</a>\"
        : \"\";
    $cpmvlink = "<a href=\"\$MyRelativeURL?action=cmone&which=$fname\"
        .
        \"&wdir=$wdir\" . options2url() . \">$CopymoveCode</a>\";

    // find out if the file is one that can be edited
    $editlink = \"\";
    if ($ftype != "Folder")
        if (Ok2Edit($ftype))
            if (TargetOK($fname))
                $editlink = "<a
href=\"\$MyRelativeURL?action=edit&which=$fname\" .
                \"&wdir=$wdir\" . options2url() . \">\" .
                \"<font face='Verdana, Arial, Helvetica' size='1' \" . \"color='#666666'
                title=\"Edit $fname\"><b>$EditCode\" . \"</b></font></a>\";

```

```

    $fsize=string($file_table[$i][1]);
    $fileicon = fileIcon($ftype);
    $filelink = fileLink($fname,$ftype,$fsize,$fdate);

    if ($cells == 0)
        echo "<tr>\n";
    if ($cols != 1)
        echo "<td width=$width%>\n";
    showFile($fileicon,$filelink,$i +
    $foc,$fname,$chmodlink,$renamelink,$editlink,
        $deletelink,$cpmvlink);
    if ($cols != 1)
        echo "</td>\n";
    $cells++;
    if ($cells % $cols == 0 && $cols != 1)
    {
        $cells=0;
        echo "</tr>\n";
    }
}

if ($cells != 0 && $cols != 1) /* In case last row was incomplete
*/
    echo "</tr>\n";
    echo "</table>";
}

/* This function shows one file.
* There are two cases: (a) single-column and (b) multiple columns.
*/
function
showFile($icon,$link,$id,$checkboxvalue,$chmdlink,$renamelink,$editli
nk,
        $dellink,$cmlink )
{
    global $fmicondir, $cols;

    /* Each file is shown in a very small table */
    if ($cols == 1)
    {
        echo <<<sfhtml1
        <tr>
        <td width=32 valign="middle">
            </td>
        <td>$link</td>
        <td align=center>$chmdlink</td>
        <td align=center>$renamelink</td>
        <td align=center>$editlink</td>
        <td align=center>$dellink</td>
        <td align=center>$cmlink</td>
        <td align=center><input type='checkbox' name='id_$id'
value='$checkboxvalue'></td>
        </tr>

sfhtml1;
        return;
    }

    echo <<<sfhtml

```

```

<table cellpadding=0 cellspacing=2 border=0>
<tr><td width=32 rowspan="3" valign="middle">
    </td>
    <td>$link</td></tr>
<tr><td><img src=$fmicondir/sepline.gif border=0></td></tr>
<tr><td class="sidebar"><input type='checkbox' name='id_$id'
                                value='$checkboxvalue'>&nbsp;
sfhtml;

if ($renamelink)
    echo "$renamelink&nbsp;";
if ($editlink)
    echo "$editlink&nbsp;";
if ($dellink)
    echo "$dellink&nbsp;";
if ($cmlink)
    echo "$cmlink";

echo "</td></tr></table>\n";
}

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*           MAIN PAGE HTML CODE
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */

function displayActions()
{
    global $MyRelativeURL, $wdir, $MkdirAllowed, $MkfileAllowed;

    $createbuttons = ($MkdirAllowed && CreateDirOK($wdir)) ?
        "<input class=sidebarbutton type=\"submit\" name=\"action\"
value=\"Folder\">"
        : "";

    if(!$MkdirAllowed && CreateDirOK($wdir))
        $createbuttons.=" [Directory creation not permitted here
]<br>";

    if ($MkfileAllowed && UpPathOK($wdir))
    {
        if ($MkdirAllowed && CreateDirOK($wdir))
            $createbuttons .= "&nbsp;&nbsp;&nbsp;";
        $createbuttons .=
            "<input class=sidebarbutton type=\"submit\" name=\"action\"
value=\"Text file\">";
    }

    else
        $createbuttons .=" [New File Creation not permitted here ]<br>"
;

    $opts = options2form();

```



```

$docode = "";
if ($MkdirAllowed || $MkfileAllowed)
    $docode .= <<<dohtml1
    <form method="post" action="$MyRelativeURL">
        $opts
        <input type="hidden" name="wdir" value="$wdir">
        <table width=100% cellpadding=2 cellspacing=0 border=0
            style="border: solid 1px #000000;">
        <tr><td bgcolor="#777777" align="center"
            style="border-bottom: solid 1px #000000;">
            <span class="sidebarhead">Create</span></td></tr>
        <tr><td align="center" class="sidebar" bgcolor="#DDDDDD">
            <table width=100% cellpadding=1 cellspacing=0 border=0
class="sidebar">
                <tr><td align="left">
                    <br>
                    &nbsp;   Name: </td></tr>
                <tr><td align="center">
                    <input class=sidebarbutton size="15" type="text"
name="which"><br>
                    </td></tr>
                <tr><td align="left">
                    &nbsp;   Type: </td></tr>
                <tr><td align="center">
                    $createbuttons</td></tr>
                <tr><td align="left" class=sidebar>
                    &nbsp;   </td></tr>
            </table>
        </td></tr>
        </table>
    </form>
    <br>
dohtml1;

    if ($MkfileAllowed && UpPathOK($wdir))
        $docode .= <<<dohtml2
        <form method="post" action="$MyRelativeURL"
enctype="multipart/form-data">
            <input type="hidden" name="action" value="upload">
            <input type="hidden" name="wdir" value="$wdir">
            $opts
            <table width=100% cellpadding=2 cellspacing=0 border=0
                style="border: solid 1px #000000;">
            <tr><td bgcolor="#777777" align="center"
                style="border-bottom: solid 1px #000000">
                <span class="sidebarhead">Upload</span></td></tr>
            <tr><td align="center" class="sidebar" bgcolor="#DDDDDD">
                <table width=100% cellpadding=2 cellspacing=0 border=0
class="sidebar">
                    <tr><td align="left">
                        <br>&nbsp;   File: </td></tr>
                    <tr><td align="center">
                        <input class=sidebarbutton type="file"
name="upldfile" size="10">
                        </td></tr>
                    <tr><td align="center">
                        <input class=sidebarbutton type="submit"
value="Upload it">
                        </td></tr>
                </table>
            </td></tr>
        </table>
        </form>
    </if>
    </pre>

```

```

                </table>
            </td></tr>
        </table>
    </form>
    <br>

dohtml2;

    return ($docode);
}

function displayOptions()
{
    global $MyRelativeURL, $details, $wdir, $cols, $sort;

    $arr = array(1,2,3,4,5,6,7,8,9,10);
    foreach ($arr as $tmp)
        if ($tmp == $cols)
            $column_opt .= "<option selected class=sidebar>$tmp</option>";
        else
            $column_opt .= "<option class=sidebar>$tmp</option>";

    $arr = array("Name", "Size", "Date", "Type");
    foreach ($arr as $tmp)
        if ($tmp == $sort)
            $sort_opt .= "<option selected class=sidebar>$tmp</option>";
        else
            $sort_opt .= "<option class=sidebar>$tmp</option>";

    $show_file_details = "<input type=\"checkbox\" name=\"details\"
valign=\"middle\"";
    if ($details)
        $show_file_details .= " checked";
    $show_file_details .= " class=button>";

    $docode = <<<dohtml
    <table width=100% cellpadding=2 cellspacing=0 border=0
        style="border: solid 1px #000000;" class="sidebar">
    <tr><td bgcolor="#777777" align="center"
        style="border-bottom: solid 1px #000000">
        <span class="sidebarhead">Display Options</span></td></tr>
    <tr><td align="center" class="sidebar" bgcolor="#DDDDDD">
        <form method="post" action="$MyRelativeURL">
            <input type="hidden" name="wdir" value="$wdir">
            <br>Sort by:
            <select name="sort" class=sidebar>$sort_opt</select>
            <p>Columns:
            <select name="cols" class=sidebar>$column_opt</select>
            <p>File details: $show_file_details
            <br><br><br>
            <input type=submit name=Submit value=" Set options "
class=sidebarbutton>
            </form>
        </td></tr></table>

dohtml;

    return ($docode);
}

```

```

function htmlStart()
{
    global $rootname, $fmicondir, $fmcscs;

    $x = $rootname;
    echo <<<html1
    <html>
    <head>
    <link rel="stylesheet" type="text/css" href="$fmcscs" />
    <title>PPG file manager: $x</title>
    </head>
    <body bgcolor="#AAADDD">
    <img src=$fmicondir/fmlogo.png>

html1;
}

function htmlDrawPage()
{
    global $rootname,$wdir,$MyRelativeURL,$sort,$cols,$fmicondir,
        $details,$folder_table,$file_table,$RootPath,$msg;

    htmlStart();

    $opturl = options2url();
    $opts = options2form();
    $me = "$MyRelativeURL?wdir=$wdir$file$opturl";

    echo
    "<center>
    <table cellpadding=2 cellspacing=10 border=0 width=100%>
    <tr>";

    /*** Main part ***/
    echo
    "    <td width=75% valign=\"top\">
    <table width=100% cellpadding=0 cellspacing=0 border=0
    style=\"border: solid 1px #000000;\">\n";

    /*** First row (current dir) ***/
    echo <<<hdp1
    <tr><td>
    <table width=100% cellpadding=0 cellspacing=5
    bgcolor="#003366"
    border=0 style="border-bottom: solid 1px #000000;";>
    <tr>
    <td class="header" align="left">
    <b>&nbsp;Directory:</b>
    $rootname/$wdir</font></td>
    <td align="right">
    <a href="$ScriptName?.$opturl"
class="header">&raquo;Home</a>&nbsp;
    <a href="$me" class="header">&raquo;
    Refresh </a>&nbsp;</td>
    </tr></table>

    </td></tr>

```

```

hdp1;

/** Second row (delete/copy) */
echo <<<hdp2

    <form method="post" action="$MyRelativeURL">
        <input type="hidden" name=wdir value="$wdir">
        $opts

    <tr><td>
        <table cellpadding=10 cellspacing=0 width=100%
bgcolor=#BBBBBB
        border=0 style="border-top: solid 1px #999999;">
        <tr>
        <td align=center width=50%>
            <input class=button type="submit" name="action"
            value="Delete checked files">
        </td>
        <td align=center>
            <input class=button type="submit" name="action"
            value="Copy/Move checked files">
        </td></tr></table>

    </td></tr>

hdp2;

/* If the current directory is a subdirectory
 * show a back link to get back to the previous directory
 */
if($wdir)
{
    echo <<<hdp4
        <tr><td style="border-top: solid 1px #999999;">
            <table cellpadding=0 cellspacing=5 border=0 width=100%
bgcolor='#DDDDDD'>
            <tr>
            <td align="center">
                <a href="$me&action=back" class=back>
                
                &nbsp;&laquo; BACK</a>&nbsp;&nbsp;
            </td></tr></table>
        </td></tr>

hdp4;
}

/** The list of files */
echo "          <tr><td style=\"border-top: solid 1px
#999999;\">\n";
displayFiles($folder_table, $file_table);
$foc = count($folder_table);
$fic = count($file_table);

echo <<<hdp5
        </td></tr>
        </form>

hdp5;

```

```

    /*** The last row ***/
    $bytes = ($details) ? ", " . sizestring( dirsized($RootPath.$wdir) )
    . " total"
        : "";

    echo <<<hdp6
        <tr><td bgcolor="#BBBBBB" style="border-top: solid 1px
#999999;">
            <table cellpadding=0 cellspacing=3 width=98%>
                <tr><td class=infoline>
                    <b>Total: </b> $foc folders, $fic files$bytes
                </td></tr></table>

            </td></tr>
        </table>

hdp6;

    if ($msg)
    {
        echo <<<hdp7
            <br>
            <hr noshade size=1 width=30% color="#999999" align=left>
            <table cellpadding=0 cellspacing=0 border=0
class=statusmsg>
                <tr><td valign=top>&raquo;&nbsp;</td><td
valign=top>$msg</td></tr>
            </table>

hdp7;
    }

    /*** Left column ***/
    echo "</td>\n<td width=25% valign=\"top\">\n";
    echo displayActions() . displayOptions();
    echo "</td></tr></table>\n</center>\n\n";

    echo "</body></html>\n";
}
?>

```