

Περιεχόμενα

Εισαγωγή.....	1
Διαμόρφωση της MySQL.....	3
Διαμόρφωση του Apache.....	8
Εγκατάσταση της PHP.....	15
Κέντρα προστασίας, πρώτες βοήθειες, Links.....	18
Εμφάνιση απειλούμενων πουλιών και ανακοινώσεων.....	21
Δημιουργία ενός ημερολογίου.....	28
Δημιουργία Φόρουμ.....	31
Χειρισμός του περιβάλλοντος διαχείρισης	61

Εισαγωγή

Πριν μερικά χρόνια όταν κάποιος ήθελε να βρει κάποιες πληροφορίες για ένα προϊόν, για ένα αεροπλάνο ,για μία υπηρεσία και γενικά κάποιες πληροφορίες για κάτι που τον ενδιέφερε έπρεπε να αρκестεί στις πληροφορίες που του παρεχόταν από το περιβάλλον όπου βρισκόταν (πόλη ή χωριό).Με την ανάπτυξη του Internet, ο κάθε άνθρωπος μπορεί να βρει πληροφορίες που τον ενδιαφέρουν από την άλλη άκρη του κόσμου και μάλιστα σε πολύ μικρό χρονικό διάστημα.

Στην αρχή αυτό γινόταν μόνο με τη μορφή κειμένου. Έπειτα εμφανίστηκαν οι πρώτες εφαρμογές browser με γραφικό περιβάλλον, η οποία άνοιξε το δρόμο για την εμφάνιση έγχρωμου κειμένου και γραφικών στις ιστοσελίδες. Αυτό ήταν που βοήθησε πολύ στην εξάπλωση του Internet.

Το σημαντικό με το Internet είναι ότι είναι ανεξάρτητο από υπολογιστές. Έτσι μπορεί να αναζητήσει πληροφορίες κάποιος με παλιών προδιαγραφών PC, κάποιος με ένα τεχνολογικά προηγμένο σταθμό εργασίας ή με ένα mainframe πολλών εκατομμυρίων δολαρίων.

Θέλοντας κι εμείς να διαθέσουμε κάποιες πληροφορίες στο Internet, τις οποίες θα μπορούσαν να τις δουν αρκετοί άνθρωποι, αποφασίσαμε να δημιουργήσουμε μία ιστοσελίδα με σκοπό την επισήμανση των απειλούμενων πουλιών στο νομό των Ιωαννίνων.

Ιδιαίτερη βαρύτητα δόθηκε στη δυναμική της διάσταση όπου επιδιώχθηκε οι εξουσιοδοτημένοι χρήστες να έχουν τη δυνατότητα ανανέωσης του υλικού. Για να γίνει αυτό πρέπει να βρεθεί κάποιος τρόπος να δημιουργήσουμε την σελίδα μας (π.χ. ποια γλώσσα προγραμματισμού για το Internet θα χρησιμοποιήσουμε, ποια βάση δεδομένων και ποιο server θα χρησιμοποιήσουμε). Η διαδικασία περιγράφεται σε αυτή την πτυχιακή εργασία.

Χωρίζετε σε δύο τμήματα. Το πρώτο ασχολείται με την εγκατάσταση και διαμόρφωση της MySQL, του Apache και της PHP και το δεύτερο με αυτό που θα είναι εμφανές από τον κόσμο του Internet.

Κεφάλαιο 1

Εγκατάσταση και διαμόρφωση της MySQL

Στο πρώτο κεφάλαιο θα ασχοληθούμε με την εγκατάσταση και την διαμόρφωση της MySQL. Αυτό γίνεται γιατί η εγκατάσταση της PHP είναι πιο εύκολη όταν είναι εγκατεστημένη η MySQL

Εγκαθιστώντας την MySQL

Ας αρχίσουμε διαμορφώνοντας την MySQL. Κάνουμε εγκατάσταση το πρόγραμμα της MySQL. Ο προκαθορισμένος κατάλογος όπου θα εγκατασταθεί η ίδια η MySQL είναι ο κατάλογος C:\mysql. Μπορείτε να την πάτε σε ένα διαφορετικό κατάλογο αν χρειάζεται, αφού εγκατασταθεί πλήρως

Εάν μετακινήσετε τελικά την MySQL, πρέπει να πείτε στο mysqld που βρίσκονται τα παρόντα, παρέχοντας επιλογές στο mysqld. Χρησιμοποιήστε το C:\mysql\bin\mysqld --help το για να εμφανίσετε όλες τις επιλογές. Για παράδειγμα, αν έχετε μετακινήσει την MySQL στο ' D:\programs\mysql ',

πρέπει να ξεκινήσετε το `mysqld` με το `' D:\programs\mysql\bin -basedir D:\programs\mysql'`.

Στις νεότερες εκδόσεις στη MySQL, μπορείτε να δημιουργήσετε ένα αρχείο `' C:\my.cnf '` που περιέχει την προκαθορισμένη επιλογή για τον MySQL διακομιστή. Αντιγράψτε το αρχείο `' C:\mysql\ my -- xxxxx.cnf '` και τροποποιήστε το ανάλογα με την προκαθορισμένη διαμόρφωση.

Windows 95/98

Η έκδοση της MySQL για τα Windows 95/98 έρχεται με δύο διαφορετικούς MySQL διακομιστές:

- `Mysqld`: Μεταγλωτισμένο με πλήρη εκσφαλμάτωση και αυτόματη δέσμευση μνήμης για έλεγχο
- `Mysqld-opt`: Βελτιστοποιημένο για ένα επεξεργαστή Pentium

Και τα δύο θα δουλέψουν σε σύγχρονους επεξεργαστές Intel X386 ή νεότερους.

Μπορείτε να ξεκινήσετε τον διακομιστή `mysqld` από την προτροπή των Windows πληκτρολογώντας τα παρακάτω:

```
C:\mysql\bin\mysqld-opt
```

Αυτό ξεκινά τον MySQL διακομιστή στο υπόβαθρο. Αν ο διακομιστής δεν ξεκινά, ελέγξτε αν το αρχείο `' \ mysql \ mysql.err'` περιέχει λάθη που δείχνουν ότι κάτι είναι λάθος. Μπορείτε να τερματίσετε τον MySQL διακομιστή εκτελώντας το `C:\ mysql \ bin \ mysqladmin -u root shutdown`

Windows NT/Win2000

Υπάρχουν μερικές διαφορές, ανάλογα αν τρέχετε την MySQL στα NT ή στα Windows 2000. Στην διαμόρφωση των NT/Win2000, το όνομα του διακομιστή είναι `mysqld-nt` και κανονικά θα εγκατασταθεί στην ίδια υπηρεσία. Μπορείτε να εγκαταστήσετε τον διακομιστή ως υπηρεσία, ως εξής:

```
C:\ mysql \ bin \ mysql-nt -install
```

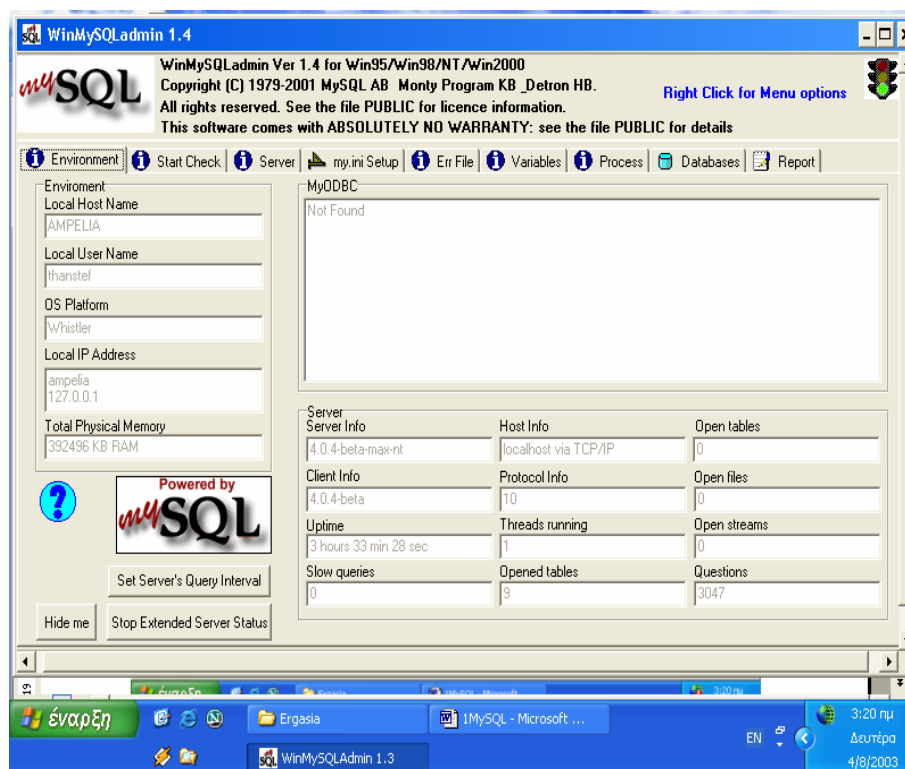
Τώρα μπορείτε να ξεκινήσετε και να σταματήσετε τον MySQL διακομιστή ως υπηρεσία, με το

```
NET START mysql
```

```
NET STOP mysql
```

Όταν τελειώσει η διαδικασία εγκατάστασης, ο χρήστης μπορεί να πάει στο φάκελο όπου είναι εγκατεστημένο το πρόγραμμα C:\mysql\bin και να τρέξει το βοηθητικό πρόγραμμα winmysqladmin.exe. Αν κάνετε διπλό κλικ πάνω στο όνομα αυτού του αρχείου θα εκκινήσει ο MySQL server και θα εμφανιστεί ένα εικονίδιο φωτεινού σηματοδότη στη γραμμή εργασιών της επιφάνειας εργασίας.

Όταν εκκινήσετε το WinMySQLAdmin για πρώτη φορά, θα σας ζητηθεί ένα όνομα χρήστη και ένας κωδικός πρόσβασης. Αφού ολοκληρωθεί η διαδικασία του λογαριασμού μπορείτε να κάνετε δεξί κλικ στο εικονίδιο του φωτεινού σηματοδότη και show me και εμφανίζεται το παράθυρο του WinMySQLAdmin. Το παράθυρο αυτό παρουσιάζεται στην εικόνα 1.1



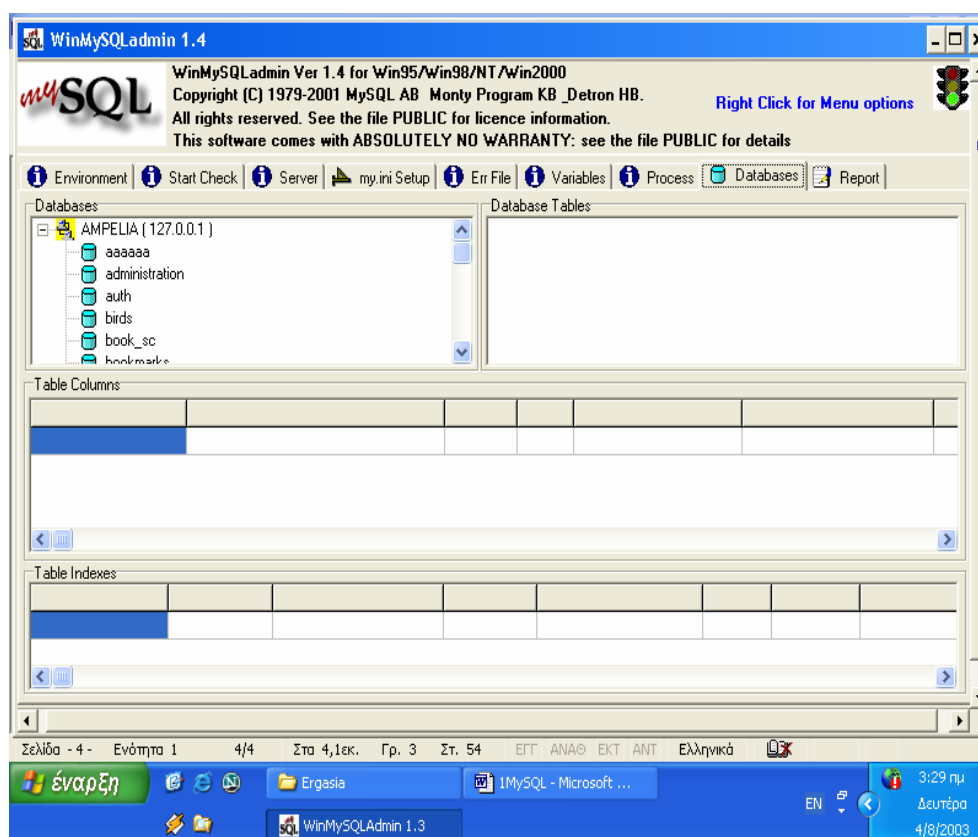
Εικόνα 1.1

Εμφάνιση του παραθύρου WinMySQLAdmin.

Το WinMySQLAdmin ανιχνεύει αυτόματα τις πληροφορίες για το περιβάλλον στο οποίο τρέχει, όπως η διεύθυνση IP και το όνομα του υπολογιστή. Οι καρτέλες που εμφανίζονται στο πάνω μέρος του παραθύρου

σας δίνουν τη δυνατότητα να εξετάσετε διάφορες πληροφορίες για το σύστημα και να προσαρμόσετε τις επιλογές διαμόρφωσης της MySQL.

Για παράδειγμα αν επιλέξετε την καρτέλα databases εμφανίζονται πληροφορίες για της βάσεις που έχουν δημιουργηθεί. Ένα παράδειγμα είναι αυτό της εικόνας 1.2



Εικόνα 1.2

Εμφάνιση της καρτέλας databases από το WinMySQLAdmin

Αφού εγκατασταθεί ο διακομιστής πρέπει να ξεκινήσει χρησιμοποιώντας την εντολή NET START MySQL. Αν θέλετε κάποιες επιλογές, πρέπει να καθοριστούν ως παράμετροι εκκίνησης του προγράμματος SCM πριν ξεκινήσετε την MySQL υπηρεσία. Όταν τρέχει το mysql-nt, μπορεί να σταματήσει χρησιμοποιώντας το mysqladmin ή χρησιμοποιώντας την εντολή NET STOP MySQL.

Για να ελέγξετε αν η MySQL δουλεύει, μπορείτε να εκτελέσετε τις παρακάτω εντολές:

```
C:\mysql\bin\mysqlshow
```



```
C:\mysql\bin\mysqlshow -u root mysql
```

```
C:\mysql\bin\mysqladmin version status proc
```

```
C:\mysql\bin\mysqladmin -u root shutdown
```

Αυτές οι εντολές δουλεύουν το ίδιο στα διάφορα λειτουργικά συστήματα των Windows.

Η MySQL θα δημιουργήσει δύο βάσεις δεδομένων, τις βάσεις δεδομένων mysql και test. Η βάση δεδομένων mysql θα χρησιμοποιηθεί για αποθήκευση των δικαιωμάτων και πρόσβαση στο διακομιστή. Η βάση δεδομένων test δεν απαιτείται, αλλά σας δίνει ένα ασφαλές μέρος να εκτελείται εντολές, να δείτε αν όλα διαμορφώθηκαν σωστά.

Αν χρειάζεστε περισσότερες πληροφορίες, δείτε την Web τοποθεσία της MySQL, στο <http://www.mysql.com>.

Κεφάλαιο 2

Εγκατάσταση και διαμόρφωση του Apache

Εγκατάσταση του Apache

Ο Apache 1.3 και νεώτερος, έχει σχεδιαστεί να τρέχει στα Windows NT 4.0 και Windows 2000. Η εγκατάσταση θα δουλέψει μόνο με την οικογένεια επεξεργαστών x86, όπως της Intel. Ωστόσο, ο Apache μπορεί επίσης να τρέξει στα Windows 95 και 98, αλλά δεν έχει ελεγχθεί. Σε όλες τις περιπτώσεις πρέπει να εγκατασταθεί δικτύωση TCP/IP. Βεβαιωθείτε ότι χρησιμοποιείτε την βιβλιοθήκη Winsock 2, αν αποφασίσετε να τον εγκαταστήσετε στα Windows 95 ή 98.

Σας προτείνεται να μεταφέρετε την έκδοση του Apache για Windows με την επέκταση .exe, αν είστε αρχάριοι και δεν θέλετε να μεταγλωττίσετε των

κώδικα προέλευσης. Αυτό το αρχείο περιέχει τον Apache διακομιστή, έτοιμο για εγκατάσταση.

Η διαδικασία εγκατάστασης θα πρέπει να σας φαίνεται γνωστή. Το πρόγραμμα εγκατάστασης θα σας ζητήσει τα παρακάτω:

- Τον κατάλογο που θα εγκαταστήσετε τον Apache. (Το προκαθορισμένο είναι το C:\ Program Files \ Apache Group \ Apache)
- Το όνομα του μενού έναρξης. (Το προκαθορισμένο είναι το Apache Web Server)
- Τον τύπο εγκατάστασης. Η επιλογή Typical εγκαθιστά τα πάντα εκτός από τον κώδικα προέλευσης. Η επιλογή minimum δεν εγκαθιστά τα εγχειρίδια ή τον κώδικα προέλευσης. Επιλέξτε Custom αν θέλετε τον κώδικα προέλευσης.

Μετά την εγκατάσταση του Apache, θα πρέπει να τροποποιήσετε τα αρχεία διαμόρφωσης που υπάρχουν στον κατάλογο conf. Θα δούμε την επεξεργασία του αρχείου διαμόρφωσης httpd.conf, αφού εγκαταστήσετε την PHP.

Εκτελώντας τον Apache στα Windows

Υπάρχουν δύο τρόποι που μπορείτε να τρέξετε τον Apache:

- Από ένα παράθυρο κονσόλας
- Ως Windows υπηρεσία

Η επιλογή της υπηρεσίας, χρησιμοποιείται γενικά στα Windows NT και 2000. Χρησιμοποιείτε αυτή την επιλογή, αν θέλετε ο Apache να αρχίζει αυτόματα όταν ξεκινά ο υπολογιστής σας και να συνεχίζει να τρέχει μετά την αποσύνδεση. Η επιλογή της κονσόλας στοχεύει βασικά στους χρήστες των Windows 95 και 98. Ωστόσο, εμφανίστηκε ένας διακομιστής μετά την έκδοση 1.3.13, που επιτρέπει στους χρήστες των Windows 95 και 98 να τρέχουν τον Apache σαν υπηρεσία. Αυτός ο διακομιστής θεωρείται λίγο πειραματικός από τους προγραμματιστές του.

Θα εγκαταστήσουμε τον Apache σαν υπηρεσία, μόνο αφού τον ελέγξουμε με επιτυχία από το παράθυρο της κονσόλας. Συνεπώς θα καλύψουμε πρώτα τον τρόπο να ξεκινάμε τον Apache από την κονσόλα και μετά θα καλύψουμε την μέθοδο της υπηρεσίας.

Τρέχοντας τον Apache σε παράθυρο κονσόλας

Για να τρέξετε τον Apache με ένα παράθυρο κονσόλας, επιλέξτε την επιλογή Start Apache as console App, από το μενού Start. Αυτό θα ανοίξει ένα παράθυρο κονσόλας και θα ξεκινήσει τον Apache. Αυτό το παράθυρο θα παραμένει ενεργό μέχρι να σταματήσετε τον διακομιστή Apache.

Για να τον σταματήσετε, θα πρέπει είτε να τρέξετε την επιλογή Shutdown Apache as Console App από το μενού Start ή να ανοίξετε ένα άλλο παράθυρο εντολών και να πληκτρολογήσετε το παρακάτω (για την έκδοση 1.3.3 και νεότερη):

```
C:\Program Files\Apache Group\Apache> apache -k shutdown
```

Αντίθετα με την MySQL, ο Apache δεν ξεκινά ως διαδικασία στο υπόβαθρο, έτσι μπορείτε να τον σταματήσετε πατώντας Ctrl-C ή Ctrl-Break στο παράθυρο κονσόλας του Apache ή κλείνοντας το παράθυρο της κονσόλας (Και πάλι αυτό θα δουλέψει μόνο για εκδόσεις παλιότερες από την 1.3.3)

Εκτελώντας τον Apache ως υπηρεσία

Πριν ξεκινήσετε τον Apache ως υπηρεσία, πρέπει να τον εγκαταστήσετε ως υπηρεσία. Να θυμάστε ότι μπορούν να εγκατασταθούν πολλές υπηρεσίες Apache σε ένα υπολογιστή, με διαφορετικά ονόματα και διαμορφώσεις.

Για να εγκαταστήσετε την προκαθορισμένη υπηρεσία Apache (που ονομάζεται Apache), τρέξτε την επιλογή Install Apache as Service (NT only) από το μενού Start. Ανοίξτε το παράθυρο Services στο Control Panel, επιλέξτε Apache και κάντε κλικ στο Start. Ο Apache θα τρέχει τώρα, κρυμμένος στο υπόβαθρο. Μπορείτε αργότερα να σταματήσετε τον Apache, κάνοντας κλικ στο Stop. Ως εναλλακτική λύση στο παράθυρο Services, μπορείτε να αρχίσετε και να σταματήσετε την υπηρεσία Apache από τη γραμμή εντολών με τα

```
NET START apache
```

```
NET STOP apache
```

Σημειώστε ότι αυτό είναι παρόμοιο με τον διακομιστή MySQL για τα NT και Windows 2000.

Ο Apache αντίθετα με άλλες εφαρμογές των Windows NT και 2000, καταγράφει τα λάθη στο δικό του αρχείο error.log, που βρίσκεται μέσα στο

φάκελο ρίζα του διακομιστή Apache. Δεν παρέχει πληροφορίες μέσου του τυπικού Event log.

Όπως αναφέρθηκε προηγουμένως, μπορούν να εγκατασταθούν πολλά στιγμιότυπα του Apache και να τρέχουν ως υπηρεσίες. Για να πείτε σε μία εγκατεστημένη υπηρεσία Apache να ξεκινήσει, να επανεκκινήσει ή να τερματίσει, θα πρέπει να παρέχετε το όνομα της υπηρεσίας ως εξής:

```
Apache -n "service name" -k start
```

```
Apache -n "service name" -k restart
```

```
Apache -n "service name" -k shutdown
```

Για την λεπτομερή υπηρεσία Apache, η επιλογή -n Apache εξακολουθεί να απαιτείται, επειδή οι εντολές -k χωρίς την επιλογή -n κατευθύνουν τον Apache να εκτελεσθεί σε ένα παράθυρο κονσόλας. Τα εισαγωγικά απαιτούνται μόνο αν το όνομα της υπηρεσίας περιέχει κενά.

Ο Apache θα συνδεθεί με τη θύρα 80 (εκτός και αν αλλάξατε τις οδηγίες Port, Listen, ή BindAddress στα αρχεία διαμόρφωσης), αφού ξεκινήσει. Για να συνδεθείτε στο διακομιστή και να αποκτήσετε πρόσβαση στην προκαθορισμένη σελίδα, ξεκινήστε ένα browser και δώστε αυτό το URL:

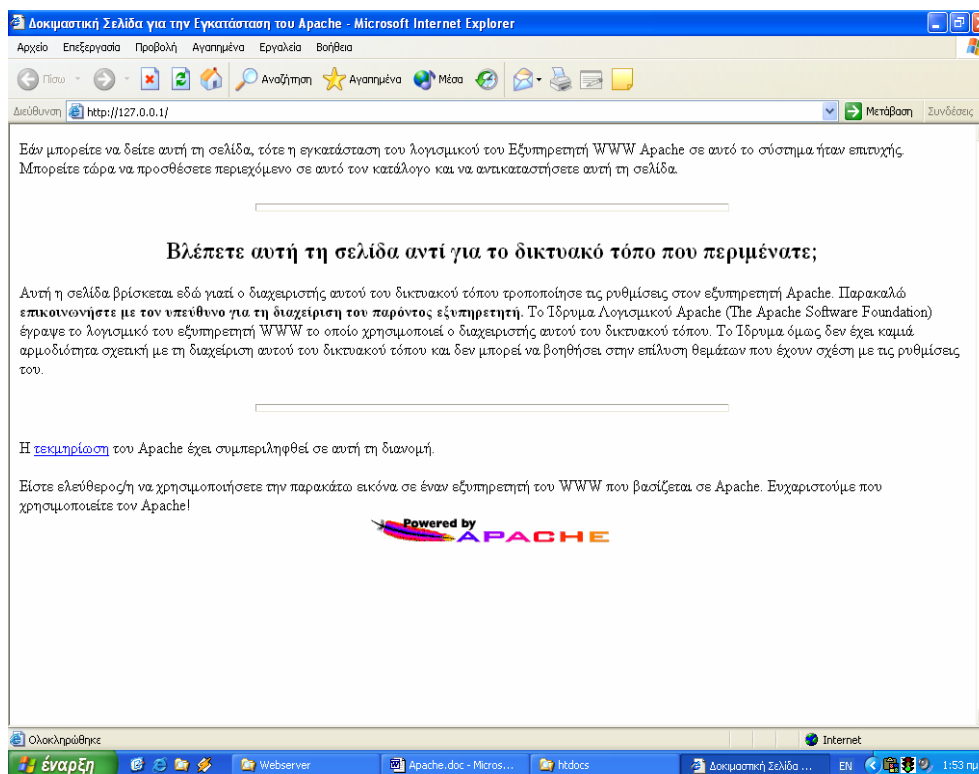
<http://localhost/>

Αυτό θα πρέπει να απαντήσει με μία σελίδα καλωσορίσματος, παρόμοια με αυτή που φαίνεται στην εικόνα 2.1 και μια σύνδεση στο εγχειρίδιο του Apache. Αν δεν συμβεί τίποτα ή πάρετε λάθη, κοιτάξτε το αρχείο error.log στον κατάλογο logs. Αν ο κύριος υπολογιστής σας δεν είναι συνδεδεμένος στο Internet, πρέπει να το χρησιμοποιήσετε αυτό με το URL:

<http://127.0.0.1/>

Αυτή είναι η IP διεύθυνση που σημαίνει localhost.

Αν έχετε αλλάξει τον αριθμό της θύρας από 80, θα πρέπει να προσαρτήσετε το :port_number στο τέλος του URL.



Εικόνα 2.1
Η σελίδα καλωσορίσματος του Apache

Η δομή του αρχείου διαμόρφωσης του Apache

Ο Apache διατηρεί όλες τις παραμέτρους διαμόρφωσης του σε αρχεία απλού κειμένου. Το κύριο αρχείο παραμέτρων διαμόρφωσης ονομάζεται `httpd.conf`. Το αρχείο αυτό περιέχει ντιρεκτίβες και περιέκτες, οι οποίοι σας δίνουν τη δυνατότητα να εγκαταστήσετε τον Apache ανάλογα με τις ανάγκες σας. Οι ντιρεκτίβες διαμορφώνουν συγκεκριμένες ρυθμίσεις του Apache, όπως οι παράμετροι για τον έλεγχο πρόσβασης, την απόδοση και τη λειτουργία στο δίκτυο. Οι περιέκτες καθορίζουν το πλαίσιο στο οποίο αναφέρονται αυτές οι ρυθμίσεις. Για παράδειγμα, οι παράμετροι εξουσιοδότησης (authorization) μπορούν να αναφέρονται στον server σαν σύνολο, σε ένα κατάλογο, ή σε ένα μεμονωμένο αρχείο.

Τα αρχεία καταγραφής του Apache

Εξ ορισμού, ο Apache περιλαμβάνει δύο αρχεία καταγραφής (log files). Το αρχείο `access_log` χρησιμοποιείται για την παρακολούθηση των αιτήσεων που στέλνουν τα client συστήματα στον server. Το αρχείο `error_log` χρησιμοποιείται για την καταγραφή σημαντικών συμβάντων, όπως π.χ. τα σφάλματα που συμβαίνουν ή την επανεκκίνηση του server. Τα αρχεία αυτά

δεν υπάρχουν μέχρι να τρέξετε τον server για πρώτη φορά. Σε συστήματα με τα Windows τα ονόματα αυτών των αρχείων είναι access.log και error.log.

Το αρχείο access_log

Όταν ένα client ζητά ένα αρχείο από τον server, ο Apache καταγράφει αρκετά στοιχεία σχετιζόμενα με την συγκεκριμένη αίτηση, συμπεριλαμβανόμενης της διεύθυνσης IP του client συστήματος, του έγγραφου που ζητήθηκε, του κωδικού κατάστασης HTTP και της τρέχουσας ώρας. Στη Λίστα 2.1 φαίνεται ένα παράδειγμα καταχωρίσεων αυτού του αρχείου καταγραφής.

Λίστα 2.1 Παράδειγμα καταχωρήσεων στο αρχείο access_log

```
127.0.0.1 - - [25/Mar/2003:02:46:07 +0200] "GET / HTTP/1.1" 200 1655
127.0.0.1 - - [25/Mar/2003:02:46:07 +0200] "GET /apache_pb.gif HTTP/1.1" 200
2326
127.0.0.1 - - [25/Mar/2003:02:46:11 +0200] "GET / HTTP/1.1" 200 1655
```

Το αρχείο error_log

Το αρχείο περιλαμβάνει τα μηνύματα σφάλματος, τα μηνύματα που εμφανίζονται κατά την εκκίνηση του server, και οποιαδήποτε άλλα σημαντικά συμβάντα λαμβάνουν χώρα κατά τη διάρκεια ζωής του server. Το αρχείο είναι το πρώτο σημείο στο οποίο θα πρέπει να ανατρέχετε όταν αντιμετωπίζετε κάποιο πρόβλημα με τον Apache. Η Λίστα 2.2 παρουσιάζει παραδείγματα καταχωρίσεων από το αρχείο error_log.

Λίστα 2.2 Παράδειγμα καταχωρήσεων στο αρχείο error_log

```
[Sat Mar 29 11:22:58 2003] [notice] Parent: Received restart signal --
Restarting the server.
[Sat Mar 29 11:22:58 2003] [notice] Child 1468: Exit event signaled. Child
process is ending.
[Sat Mar 29 11:22:58 2003] [notice] Parent: Created child process 276
[Sat Mar 29 11:22:58 2003] [notice] Child 276: Child process is running
[Sat Mar 29 11:22:59 2003] [notice] Child 1468: Released the start mutex
```

Τέλος πηγαίνουμε στο αρχείο httpd.conf και αλλάζουμε τις εξής ρυθμίσεις:

- Ορίζουμε σαν ServerRoot εκεί που βρίσκεται εγκατεστημένος ο Apache δηλ. **ServerRoot "C:/Program Files/Apache Group/Apache2"**
- Τον κατάλογο κάτω από τον οποίο θα μοιράζονται τα αρχεία δηλ. **DocumentRoot "C:/webserver"**
- Εκεί που γράφει : **<Directory "C:/webserver">** το ορίζουμε σε ότι έχουμε βάλει σαν DocumentRoot

Τώρα μπορούμε να βάλουμε κάποια αρχεία στον κατάλογο που ορίσαμε και να τα δούμε από τον browser μας.

Κεφάλαιο 3

Εγκατάσταση και διαμόρφωση της PHP

Εγκατάσταση της PHP για Windows

Είμαστε έτοιμοι τώρα να εγκαταστήσουμε την PHP για Windows. Βεβαιωθείτε ότι σταματήσατε τον Apache πριν ξεκινήσετε την διαδικασία εγκατάστασης της PHP. Η διαδικασία είναι πολύ απλή, αν καταλάβετε ότι, αντίθετα με την PHP 3, η PHP 4 έχει διαιρεθεί σε διάφορα συστατικά, που απαιτούν διάφορα DLL. Δηλαδή, δεν μπορείτε να τρέξετε την PHP σε κατάσταση CGI, ως αυτόνομο εκτελέσιμο πρόγραμμα. Πρέπει να βεβαιωθείτε ότι τα DLL υπάρχουν σε ένα κατάλογο που είναι στη διαδρομή των Windows. Ο ευκολότερος τρόπος για να το κάνετε αυτό είναι να αντιγράψετε αυτό το DLL στο SYSTEM (Windows 9x) ή στο SYSTEM32 (Windows NT), που είναι

κάτω από τον κατάλογο των Windows. Τα DLL που πρέπει να αντιγραφούν, είναι τα MSVCRT.DLL (μπορεί να είναι ήδη εκεί) και PHP4TS.DLL

Παρέχουμε τα βήματα για να εγκαταστήσετε και να διαμορφώσετε την PHP σε ένα υπολογιστή Windows.

1. Αρχίστε αντιγράφοντας το php.ini-dist στον κατάλογο '%WINDOWS%' και μετονομάστε το σε 'php.ini'. Η μεταβλητή '%WINDOWS%' συνήθως δείχνει στο C:\WINDOWS για τα Windows 9x και στο C:\WINNT για διακομιστές NT.
2. Τροποποιήστε το αρχείο php.ini και αλλάξτε την ρύθμιση extension_dir για να δείχνει στον κατάλογο που περιέχει τις DLL λειτουργικές μονάδες για τις επεκτάσεις. Ορίστε το doc_root ώστε να δείχνει στο Web διακομιστή ρίζα, δηλαδή στον ορατό κατάλογο ρίζα του διακομιστή.
3. Αφαιρέστε τα σχόλια από το αρχείο php.ini για τις λειτουργικές μονάδες που θέλετε να φορτώνονται όταν ξεκινάτε την PHP. Αφαιρέστε τα σχόλια από τις γραμμές extension=php_*.dll, για να φορτώσετε τις λειτουργικές μονάδες. Σημειώστε ότι κάποιες λειτουργικές μονάδες απαιτούν να εγκατασταθούν επιπλέον βιβλιοθήκες στο σύστημα, για να δουλέψει σωστά η λειτουργική μονάδα. Σημειώστε επίσης, ότι η υποστήριξη για MySQL είναι τώρα ενσωματωμένη στην PHP 4. Δεν απαιτείται φόρτωση μέσω αυτής της μεθόδου.

Τώρα, το μόνο που χρειάζεστε είναι να τροποποιήσετε το αρχείο httpd.conf στον κατάλογο conf του Apache, για να διαμορφώσετε τον Apache να δουλεύει με τη βιβλιοθήκη PHP CGI. Προσθέστε τις παρακάτω οδηγίες στο αρχείο config.

- ScriptAlias /php/ "c:/path-to-your-php-dir/"
- Addtype application/x-httpd-php .php
- Addtype application/x-httpd-php .html
- Action application/x-httpd-php "/php/php.exe "

Σημειώστε ότι με την οδηγία Addtype, μπορείτε να καθορίσετε πως θα πρέπει να χειρίζεται ο Apache τις διάφορες επεκτάσεις αρχείων. Στην περίπτωση που αναφέρθηκε προηγουμένως, καθορίζουμε ότι ο Apache

χειρίζεται οποιοδήποτε αρχείο .php και .html ως αρχείο php. Θα μπορούσατε, για παράδειγμα, να χειριστείτε τα αρχεία .htm και .html ως PHP script, προσθέτοντας την παρακάτω οδηγία:

- Addtype application/x-httpd-php .html
- Addtype application/x-httpd-php .htm

Ξεκινήστε τον Apache και ελέγξτε τον για να βεβαιωθείτε ότι δουλεύει η PHP. Δημιουργήστε ένα αρχείο test.php και προσθέστε τις παρακάτω γραμμές σε αυτή:

```
<? phpinfo() ?>
```

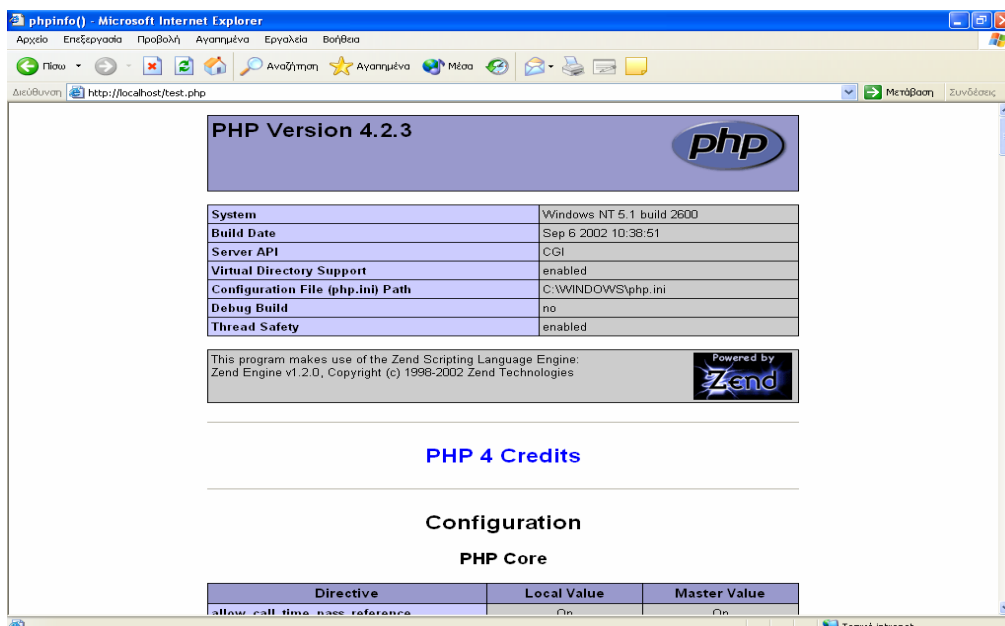
Βεβαιωθείτε ότι το αρχείο είναι στον κατάλογο ρίζα του Apache και μετά εμφανίστε τα στο browser, ως εξής:

<http://localhost/test.php>

ή

<http://your-ip-number-here/test.php>

Αν δείτε μια σελίδα παρόμοια με την παρακάτω, θα ξέρετε ότι ο Apache και η PHP δουλεύουν μαζί. Θυμηθείτε να ελέγξετε ότι δουλεύει και η MySQL μαζί τους. Κάντε γράφοντας ένα απλό PHP script για να συνδεθείτε στον διακομιστή και εισάγετε / εξάγετε κάποια δεδομένα από τη βάση δεδομένων



Εικόνα 3.1

Η σελίδα που επιβεβαιώνει ότι Apache και PHP δουλεύουν

Κεφάλαιο 4

Στη εργασία μας δώσαμε μεγάλη βαρύτητα στην επαναχρησιμοποίηση του κώδικα που δημιουργούμε και στο να είναι αυτός ο κώδικας συντηρήσιμος. Με την έννοια συντηρήσιμος, εννοούμε να μπορούμε όταν θέλουμε μετά από καιρό να διορθώσουμε κάτι στις γραμμές κώδικα που έχουμε χρησιμοποιήσει, να μπορούμε να το κάνουμε σχετικά εύκολα.

Χρησιμοποιώντας την εντολή `include()` ή `require()` μπορείτε να φορτώσετε ένα αρχείο στο PHP script σας. Το αρχείο μπορεί να περιέχει οτιδήποτε θα πληκτρολογήσατε κανονικά σε ένα script, συμπεριλαμβανομένων PHP εντολών, κειμένου, HTML ετικετών, PHP συναρτήσεων ή κλάσεων.

Στη σελίδα μας όλα τα αρχεία, μέσω της εντολής `include()`, χρησιμοποιούν το `header.php` και το `footer.php`. Αυτά τα δύο αρχεία περιέχουν εκείνες τις γραμμές κώδικα, που έχουν σχέση με την εμφάνιση ολόκληρη την ιστοσελίδα μας. Έτσι αν χρειαστεί να γίνει κάποια διόρθωση στη ιστοσελίδα, δεν θα έχουμε να αλλάξουμε σε όλα τα αρχεία που εμφανίζουν την σελίδα μας αλλά μόνο αυτά τα δύο που αναφέραμε. Επίσης δημιουργήσαμε μικρά αρχεία τα οποία τα χρησιμοποιούμε πολλές φορές. Κερδίζουμε έτσι χρόνο και δεν έχουμε πολλές γραμμές κώδικα στα script μας.

Επίσης η PHP δεν κοιτάζει την επέκταση του αρχείου στο αιτούμενο αρχείο. Αυτό σημαίνει ότι μπορείτε να ονομάσετε το αρχείο σας οτιδήποτε επιλέξετε, εφόσον δεν πρόκειται να το καλέσετε κατευθείαν. Όταν χρησιμοποιήσετε την require() για να φορτώσετε το αρχείο, τελικά θα γίνει μέρος ενός PHP αρχείου και θα εκτελείται έτσι.

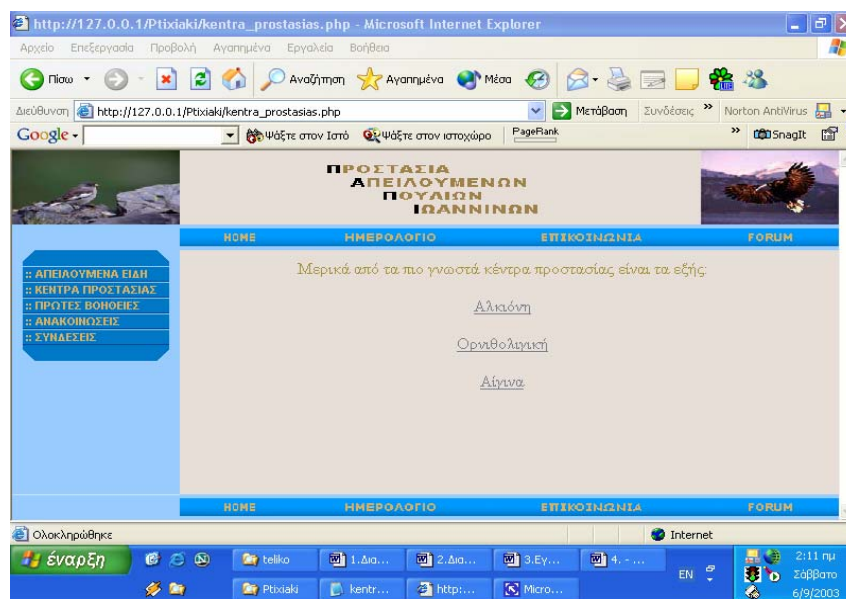
Κάτι επίσης που θα πρέπει να προσέξουμε είναι η εφαρμογή να είναι συμβατή με όλες τις εφαρμογές browser ή τουλάχιστον με τις εφαρμογές browser που χρησιμοποιούνται περισσότερο (δηλ του Internet Explorer 5.x και του Netscape Navigator 6.x). Η εφαρμογή μας είναι δοκιμασμένη και στις δύο εφαρμογές και δουλεύει κανονικά.

Τέλος θα πρέπει να προσέξουμε η εφαρμογή να είναι συμβατή με τις διάφορες αναλύσεις οθόνης. Ο κάθε χρήστης έχει διαφορετική οθόνη και συνεπώς διαφορετική ανάλυση οθόνης στο σύστημα που χρησιμοποιεί.

Εμφάνιση Κέντρων προστασίας

Θα ξεκινήσουμε από αυτά τα τρία κουμπιά, δηλ Κέντρα προστασίας, πρώτες βοήθειες και Links γιατί δεν χρειάζονται ιδιαίτερη εξήγηση. Κάποιος με απλές γνώσεις HTML θα μπορέσει να καταλάβει πως λειτουργούν αυτά τα κουμπιά.

Το πρώτο κουμπί εμφανίζει τα κέντρα προστασίας που βρίσκονται ανά την Ελλάδα. Η έξοδος αυτού του κουμπιού φαίνεται στην εικόνα 4.1

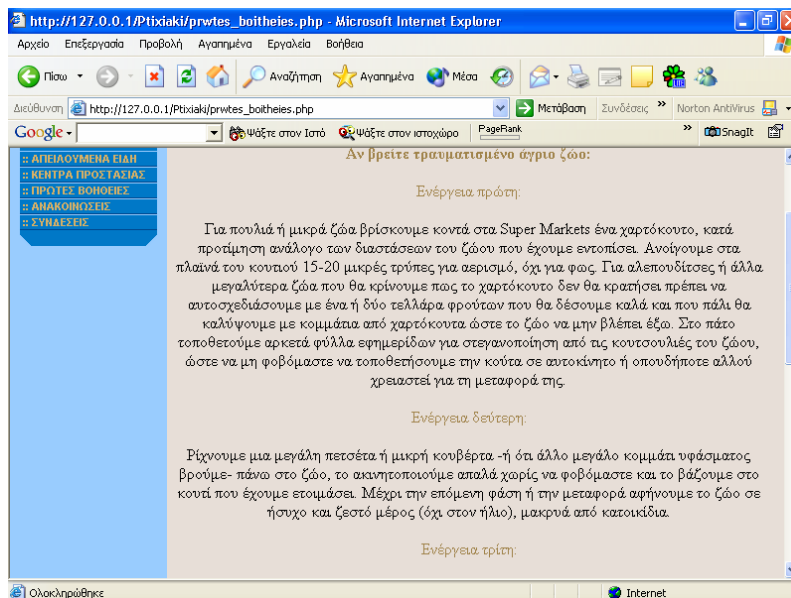


Εικόνα 4.1

Εμφάνιση του κουμπιού Κέντρα Προστασίας

Εμφάνιση Πρώτων Βοηθειών

Με το κουμπί αυτό εμφανίζεται τη πρέπει να κάνουμε σε περίπτωση που βρούμε ένα τραυματισμένο ζώο. Αυτό φαίνεται στην εικόνα 4.2

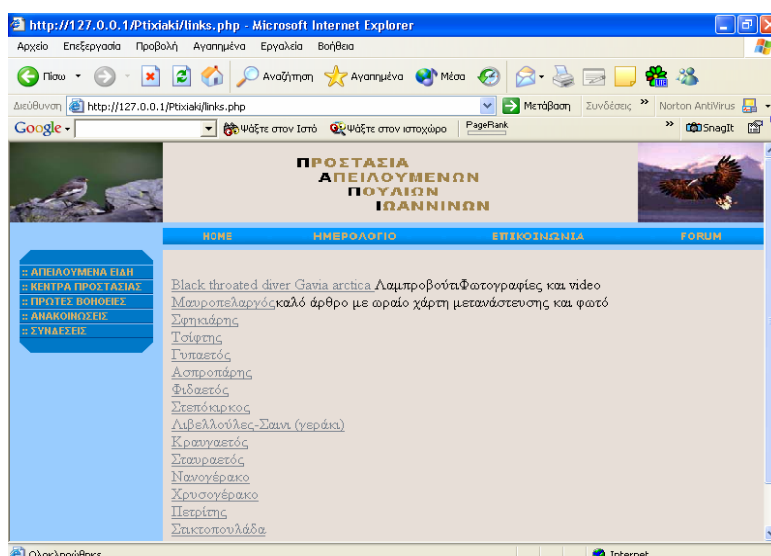


Εικόνα 4.2

Εμφάνιση του κουμπιού πρώτες βοήθειες

Εμφάνιση των Links

Τέλος το κουμπί Links εμφανίσει συνδέσεις προς άλλες ιστοσελίδες που έχουν σχέση με απειλούμενα πουλιά. Αυτό φαίνεται στην εικόνα 4.3.



Εικόνα 4.3

Εμφάνιση του κουμπιού Links

Κεφάλαιο 5

Εμφάνιση των απειλούμενων πουλιών και εμφάνιση των ανακοινώσεων

Στο κεφάλαιο αυτό θα εμφανίσουμε όλα τα απειλούμενα πουλιά που βρίσκονται στη βάση δεδομένων καθώς και όλες τις ανακοινώσεις.

Σημειώνω ότι ανάλογα με την θεματική ενότητα η οποία θα είχε αναπτυχθεί θα δημιουργούσαμε την αντίστοιχη βάση δεδομένων στην οποία θα καταχωρούνταν δυναμικά σχετικά πεδία πληροφοριών

Το πρώτο script `apeiloumena.php` ξεκινά περιλαμβάνοντας το `library_birds.php`, το οποίο περιέχει όλες τις βιβλιοθήκες συναρτήσεων για την εφαρμογή αυτή.

Υπάρχουν κάποιες κλήσεις σε HTML συναρτήσεις εξόδου, όπως η `do_html_header()` και η `do_html_footer()`. Αυτές οι συναρτήσεις περιέχονται στο `output_birds.php`.

Το πιο σημαντικό τμήμα του script είναι το εξής:

```
// λήψη πουλιών από τη βάση δεδομένων
$cat_array = get_categories();
// εμφάνιση συνδέσεων στις σελίδες πουλιών
```

display_categories(\$cat_array);

Οι συναρτήσεις `get_categories()` και `display_categories()` βρίσκονται στις βιβλιοθήκες συναρτήσεων `birds_fns.php` και `output_birds.php` αντίστοιχα. Η συνάρτηση `get_categories()` επιστρέφει ένα πίνακα με τις κατηγορίες, ο οποίος στη συνέχεια περνάει στη `display_categories()`. Ας δούμε τον κώδικα της `get_categories()`. Φαίνεται στη λίστα 5.1

Λίστα 5.1 `get_categories()` Συνάρτηση από το `birds_fns.php` – Συνάρτηση που ανακαλεί μία λίστα κατηγοριών από τη βάση δεδομένων.

```
function get_categories()  
{  
    // ερώτημα στη βάση για λίστα από τις διαθέσιμα πουλιά  
    $conn = db_connect();  
    $query = "select * from birds order by bird_id";  
    $result = @mysql_query($query);  
    if (!$result)  
        return false;  
    $num_cats = @mysql_num_rows($result);  
    if ($num_cats ==0)  
        return false;  
    $result = db_result_to_array($result);  
    return $result;  
}
```

Όπως μπορείτε να δείτε, αυτή η συνάρτηση συνδέεται με τη βάση δεδομένων και ανακτά μία λίστα όλων των στοιχείων των πουλιών. Έχουμε χρησιμοποιήσει μία συνάρτηση που ονομάζεται `db_result_to_array()`, η οποία βρίσκεται στο `db_news.php`. Αυτή η συνάρτηση φαίνεται στη Λίστα 5.2. Παίρνει ένα MySQL αναγνωριστικό αποτελεσμάτων και επιστρέφει ένα αριθμημένο πίνακα γραμμών, όπου κάθε γραμμή είναι ένας συσχετιζόμενος πίνακας.

Λίστα 5.2 db_result_to_array() Συνάρτηση που μετατρέπει ένα αναγνωριστικό αποτελέσματος της MySQL σε ένα πίνακα αποτελεσμάτων

```
function db_result_to_array($result)
{
    $res_array = array();

    for ($count=0; $row = @mysql_fetch_array($result); $count++)
        $res_array[$count] = $row;

    return $res_array;
}
```

Στην περίπτωση μας, θα επιστρέψουμε αυτόν τον πίνακα πίσω στο `apeiloumena.php` όπου το περνάμε στη συνάρτηση `display_categories()` από το `output_birds.php`. Αυτή η συνάρτηση εμφανίζει κάθε κατηγορία ως σύνδεση στη σελίδα που περιέχει τα πουλιά αυτής της κατηγορίας. Ο κώδικας αυτής της συνάρτησης φαίνεται στη Λίστα 5.3.

Λίστα 5.3 `display_categories()` Συνάρτηση που εμφανίζει ένα πίνακα από κατηγορίες ως λίστα με συνδέσεις στις κατηγορίες αυτές

```
function display_categories($cat_array)
{
    if (!is_array($cat_array))
    {
        echo "No birds currently available<br>";
        return;
    }
    echo "<ul>";
    foreach ($cat_array as $row)
    {
        $url = "show_birds.php?bird_id=" . ($row["bird_id"]);

```

Λίστα 5.3 Συνέχεια

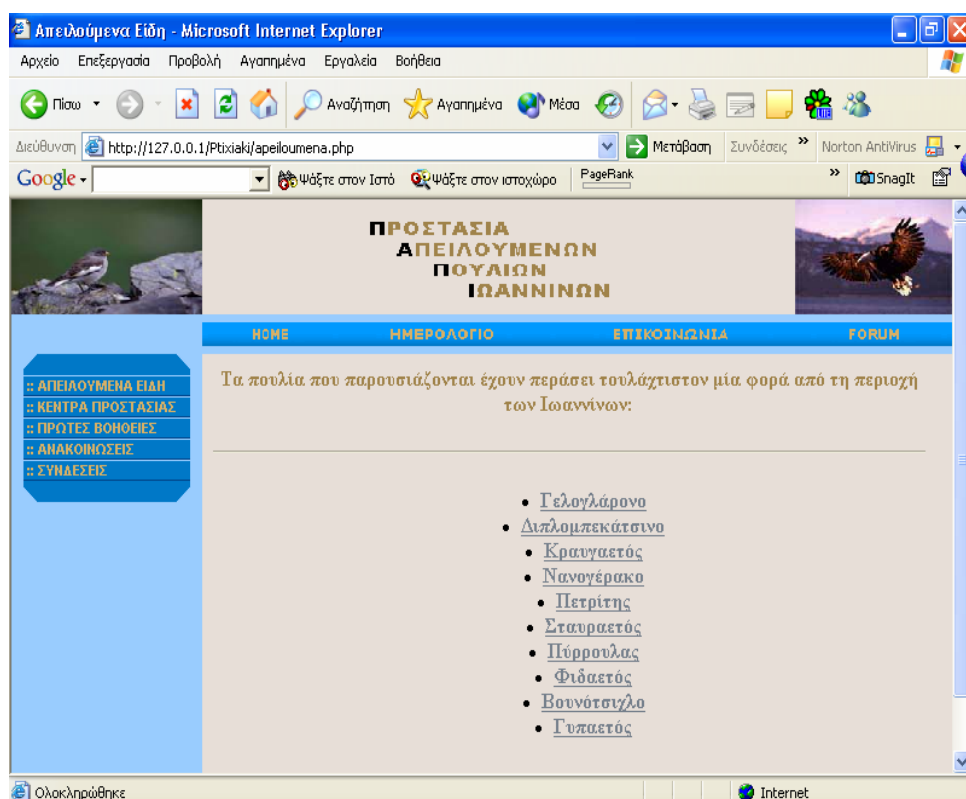
```

$title = $row["bird_title"];
    echo "<li>";
    do_html_url($url, $title);
}
echo "</ul>";
echo "<hr>";
}

```

Αυτή η συνάρτηση μετατρέπει κάθε κατηγορία της βάσης δεδομένων σε μία σύνδεση. Κάθε σύνδεση πηγαίνει στο επόμενο script – το show_birds.php – αλλά το καθένα έχει διαφορετική παράμετρο, το bird_id

Αυτή η παράμετρος προς το επόμενο script καθορίζει ποια κατηγορία θα δούμε τελικά. Το αποτέλεσμα της εξόδου του script birds.php φαίνεται στην εικόνα 5.1



Εικόνα 5.1

Η σελίδα birds.php εμφανίζει τις κατηγορίες των πουλιών που είναι διαθέσιμες

Εμφάνιση λεπτομερειών των πουλιών

Το script `show_birds.php` παίρνει το `bird_id` σαν παράμετρο και ανακαλεί και εμφανίζει τις λεπτομέρειες του πουλιού. Ο κώδικας του script φαίνεται στη Λίστα 5.3

Λίστα 5.3 Αυτό το script δείχνει τις λεπτομέρειες ενός συγκεκριμένου πουλιού

```
<?
    include('header.php');
    include ('library_birds.php');
    session_start();

    // παίρνει το πουλί από τη βάση
    $book = get_birds_details($bird_id);
    do_html_header($book["bird_tittle"]);
    display_birds_details($book);
    // Αν συνδέθηκε σαν administrator
    if(session_is_registered("admin_user"))
    {
        display_button("index.php", "next", "Next");
        display_button("admin.php", "admin-menu", "Admin Menu");
        display_button("edit_category_form.php?catid=$catid",          "edit-
category", "Edit Category");
    }
    else
    {
        $bird_id=$bird_id-1;
        display_button("show_birds.php?bird_id=$bird_id",          "continue-
shopping", "Previous");
    }
}
```

Λίστα 5.3 ΣΥΝΕΧΕΙΑ

```

$bird_id=$bird_id+2;
    display_button("show_birds.php?bird_id=$bird_id",      "continue-
shopping", "Previous");
    do_html_footer();
    include('footer.php');
?>

```

Και πάλι με αυτό το script, κάνουμε παρόμοια με εκείνα της προηγούμενης σελίδας. Ξεκινάμε χρησιμοποιώντας το

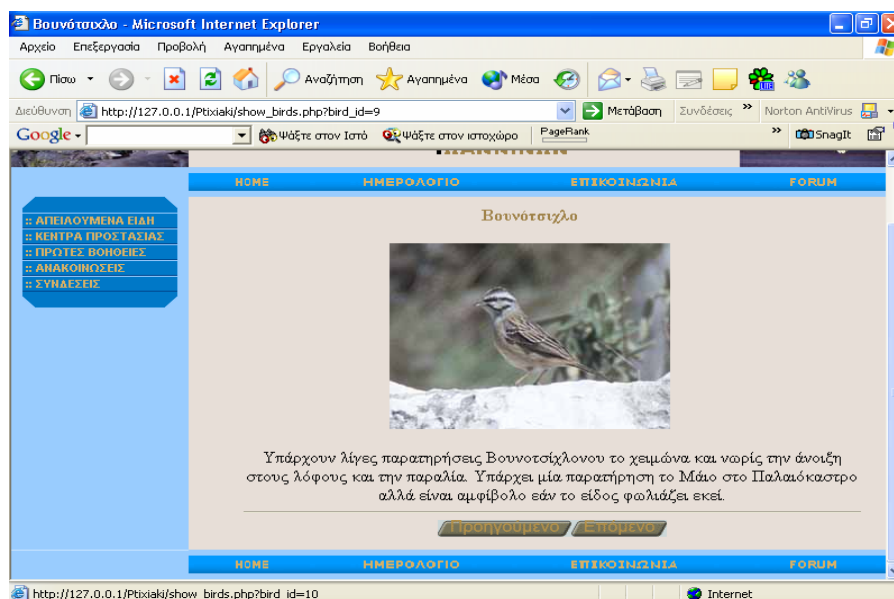
```
$book = get_birds_details($bird_id);
```

για να πάρουμε της πληροφορίες του πουλιού από τη βάση δεδομένων και

```
display_birds_details($book);
```

για να εμφανίσουμε τα δεδομένα σε HTML.

Αυτό που πρέπει να παρατηρήσετε εδώ, είναι ότι η `display_birds_details()` ψάχνει για ένα αρχείο εικόνας του πουλιού, με τη μορφή `images/$pictures.jpg`. Αν αυτό το αρχείο δεν υπάρχει, δεν θα εμφανιστεί καμία εικόνα. Η έξοδος από το script αυτό φαίνεται στην εικόνα 5.2



Εικόνα 5.2

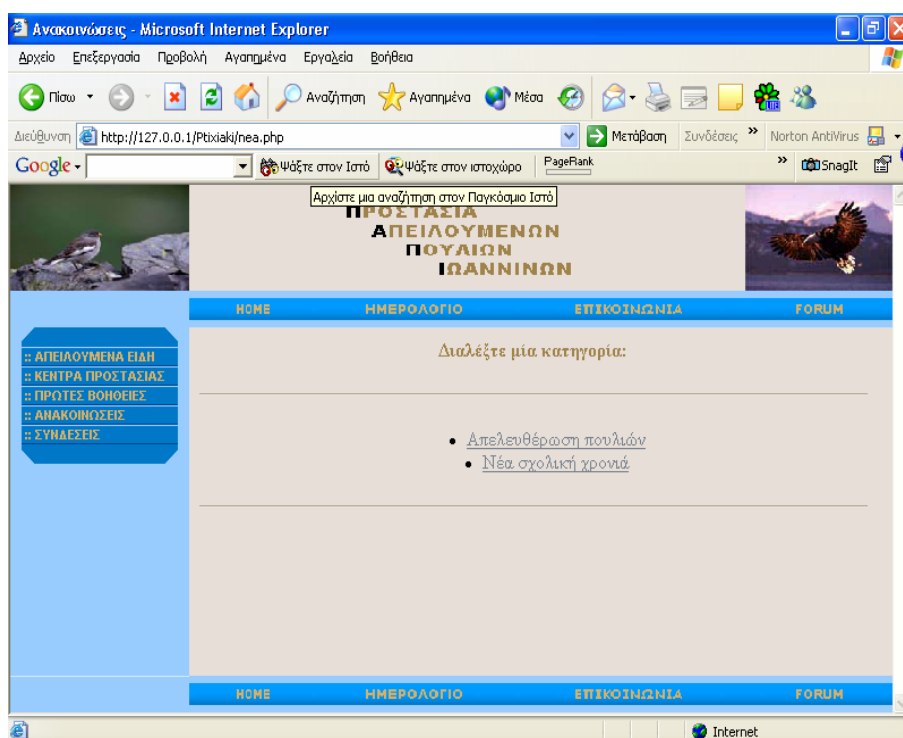
Έξοδος από το `show_birds.php`

Πρέπει επίσης να παρατηρήσουμε την ύπαρξη δύο κουμπιών στο τέλος της σελίδας – τα Next και Previous – τα οποία είναι για να μεταφέρουν το χρήστη στο επόμενο και προηγούμενο πουλί αντίστοιχα, χωρίς ο χρήστης να έχει να επιστρέφει στην προηγούμενη σελίδα και να πατάει στο επόμενο πουλί.

Εμφάνιση ανακοινώσεων

Πατώντας στο κουμπί ανακοινώσεις, καλούνται από τη βάση δεδομένων όλες οι ανακοινώσεις που είναι αποθηκευμένες εκεί.

Η λειτουργία εμφάνισης των ανακοινώσεων είναι παρόμοια με την εμφάνιση των πουλιών και δεν θα μπούμε στη διαδικασία να το εξηγήσουμε. Η εμφάνιση των ανακοινώσεων φαίνεται στην εικόνα 5.3



Εικόνα 5.3
Εμφάνιση μίας ανακοίνωσης

Κεφάλαιο 6

Δημιουργία ενός ημερολογίου

Σε αυτή την ενότητα θα ασχοληθούμε με την κατασκευή ενός ημερολογίου το οποίο θα μπορεί να εμφανίζει ημερομηνίες για οποιοδήποτε μήνα μεταξύ των ετών 1980 και 2010. Ο χρήστης θα έχει τη δυνατότητα να επιλέξει τόσο το μήνα όσο και το έτος από αναδιπλούμενες λίστες, και οι ημερομηνίες για τον επιλεγμένο μήνα θα εμφανίζονται οργανωμένες με βάση τις ημέρες της εβδομάδας. Θα το δημιουργήσουμε με βάση δύο μεταβλητές – μία για τον μήνα και μία για το έτος – τις τιμές των οποίων θα παρέχει ο χρήστης.

Έλεγχος της εισόδου του χρήστη

Όταν ο χρήστης ανοίγει την εφαρμογή ημερολογίου για πρώτη φορά, δεν εισάγει καμία πληροφορία. Συνεπώς, πρέπει να διασφαλίσουμε ότι το script μας θα μπορεί να χειρίζεται το γεγονός ότι οι μεταβλητές για το μήνα και το έτος μπορεί να μην έχουν καθορισμένη τιμή. Για το σκοπό αυτό χρησιμοποιούμε τη συνάρτηση `checkdate()` η οποία ελέγχει αν έχουν εισαχθεί οι πληροφορίες μήνα και έτους από μία φόρμα. Εάν οι πληροφορίες αυτές δεν

έχουν εισαχθεί, η `checkdate()` επιστρέφει την τιμή `false` επειδή το `script` δεν μπορεί να σχηματίσει μία έγκυρη ημερομηνία. Επιπλέον διασφαλίζει ότι τα δεδομένα που υποβάλει ο χρήστης μέσω μίας φόρμας, αποτελούν μία σωστή ημερομηνία. Αυτό φαίνεται στη λίστα 6.1

Λίστα 6.1 Έλεγχος της συνοδού του χρήστη για το `script` Ημερολογίου

```
<?
if (!checkdate($_POST[month], 1, $_POST[year])) {
    $nowArray = getdate();
    $month = $nowArray['mon'];
    $year = $nowArray['year'];
} else {
    $month = $_POST[month];
    $year = $_POST[year];
}
$start = mktime (12, 0, 0, $month, 1, $year);
$firstDayArray = getdate($start);
?>
```

Εάν η ημερομηνία δεν είναι έγκυρη χρησιμοποιούμε τη συνάρτηση `getdate()` για να δημιουργήσουμε μία διάταξη στοιχείων η οποία αποθηκεύει την τρέχουσα ημερομηνία/ώρα. Έπειτα ορίζουμε μόνοι μας τιμές στις μεταβλητές `$month` και `$year`, χρησιμοποιώντας τις τιμές των στοιχείων `mon` και `year` της διάταξης. Εάν ο χρήστης έχει εισάγει τιμές στη φόρμα, τις τοποθετούμε στις μεταβλητές `$month` και `$year` και διατηρούμε άθικτες τις τιμές της γενικής μεταβλητής `$_POST`.

Τώρα που είμαστε σίγουροι ότι έχουμε έγκυρα δεδομένα στις μεταβλητές `$month` και `$year`, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `mktime()` για να δημιουργήσουμε μία χρονική τιμή για την πρώτη μέρα του μήνα.

Κατασκευή HTML φόρμας

Εκτός από το `script`, θα πρέπει επίσης να δημιουργήσουμε ένα περιβάλλον μέσω του οποίου οι χρήστες θα μπορούν να ζητούν την εμφάνιση δεδομένων για ένα συγκεκριμένο μήνα και έτος. Για το σκοπό αυτό θα δημιουργήσουμε στοιχεία `SELECT` (λίστες επιλογών). Ο κώδικας για την δημιουργία της φόρμας φαίνεται στη λίστα 6.2

Λίστα 6.2 Κώδικας εμφάνισης HTML φόρμας

```
<html>
<head>
<title><?php      print      "Calendar:      ".$firstDayArray['month']."
".$firstDayArray['year'] ?></title>
<head>
<body>
<form method="post" action="<?php print "$_SERVER[PHP_SELF]";
?>">
<select name="month">
<?php
$months = Array("January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "November",
"December");
for ($x=1; $x <= count($months); $x++) {
    print "\t<option value=\"$x\"";
    print ($x == $month)? " SELECTED":"";
    print ">".$months[$x-1]."\n";
}
?>
</select>
<select name="year">
<?php
for ($x=1980; $x<=2010; $x++) {
    print "\t<option";
    print ($x == $year)? " SELECTED":"";
    print ">$x\n";
}
?>
</select>
<input type="submit" value="Go!">
</form>
```

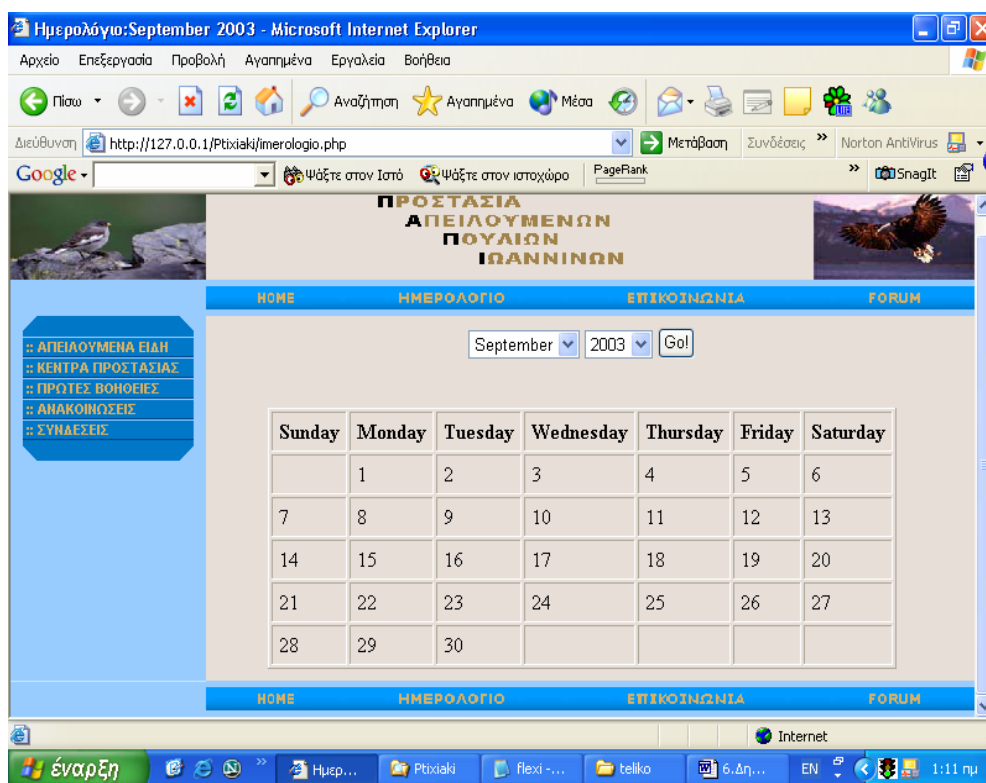

Λίστα 6.2 Συνέχεια

```
</body>
```

```
</html>
```

Στο σημείο αυτό έχουμε μία φόρμα η οποία μπορεί να στέλνει παραμέτρους για το μήνα και το έτος στον εαυτό της και εξ ορισμού χρησιμοποιεί τον τρέχοντα μήνα και έτος, είτε το μήνα και το έτος που επέλεξε ο προηγούμενος χρήστης.

Η έξοδος από αυτό το script φαίνεται στην εικόνα 6.1



Εικόνα 6.1
Εμφάνιση του ημερολογίου

Κεφάλαιο 7

Δημιουργία Φόρουμ

Σχεδίαση της βάσης δεδομένων

Υπάρχουν ορισμένα χαρακτηριστικά τα οποία πρέπει να αποθηκεύσουμε σχετικά με κάθε άρθρο που δημοσιεύεται στο φόρουμ: ποιος το έγραψε (poster), τον τίτλο του άρθρου, πότε δημοσιεύτηκε και το σώμα του άρθρου. Επομένως, θα χρειαστούμε ένα πίνακα για άρθρα. Θα δημιουργήσουμε ένα μοναδικό χαρακτηριστικό για κάθε άρθρο, που θα ονομάζεται posted

Κάθε άρθρο θα πρέπει να έχει μερικές πληροφορίες σχετικά με το που ανήκει στην ιεραρχία. Θα μπορούσαμε να αποθηκεύσουμε πληροφορίες σχετικά με τα “παιδιά” ενός άρθρου, μαζί με το άρθρο. Ωστόσο, κάθε άρθρο μπορεί να έχει πολλές απαντήσεις και αυτό μπορεί να οδηγήσει σε προβλήματα στη δομή της βάσης δεδομένων. Καθώς το κάθε άρθρο μπορεί να είναι μία απάντηση σε κάποιο άλλο, είναι πιο εύκολο να αποθηκεύσουμε

μία αναφορά στο γονικό άρθρο, δηλαδή στο άρθρο στο οποίο απαντά αυτό το άρθρο.

Επομένως, πρέπει να αποθηκεύσουμε τα παρακάτω δεδομένα για κάθε άρθρο:

- `postid`: Ένα μοναδικό αναγνωριστικό για κάθε άρθρο
- `parent`: Το `postid` του γονικού άρθρου
- `poster`: Το συντάκτη του άρθρου
- `title`: Τον τίτλο του άρθρου
- `posted`: Την ημερομηνία και την ώρα που δημοσιεύτηκε το άρθρο
- `message`: Το σώμα του άρθρου

Θα προσθέσουμε μερικές λεπτομέρειες σε αυτό.

Όταν προσπαθούμε να προσδιορίσουμε αν ένα άρθρο έχει απαντήσεις, θα πρέπει να εκτελούμε ένα ερώτημα για να δούμε ποια άλλα άρθρα έχουν το άρθρο αυτό ως γονέα. Θα χρειαστούμε αυτή την πληροφορία για κάθε δημοσίευσή μας. Όσο λιγότερα ερωτήματα πρέπει να εκτελέσουμε, τόσο πιο γρήγορα θα εκτελείται ο κώδικας μας. Μπορούμε να καταργήσουμε αυτά τα ερωτήματα προσθέτοντας ένα πεδίο που θα δείχνει αν υπάρχουν απαντήσεις. Αυτό το πεδίο θα το ονομάσουμε `children` και θα το κάνουμε δυαδικό. Η τιμή θα είναι 1, αν ο κόμβος έχει παιδιά και 0 αν δεν έχει.

Υπάρχει πάντα ένα τίμημα για τις βελτιστοποιήσεις. Εδώ επιλέγουμε να αποθηκεύσουμε περιττά δεδομένα. Καθώς αποθηκεύουμε τα δεδομένα με δύο τρόπους, θα πρέπει να είμαστε προσεκτικοί και να διασφαλίσουμε ότι οι δύο αναπαραστάσεις συμφωνούν μεταξύ τους. Όταν προσθέτουμε εξαρτημένα άρθρα, θα πρέπει να ενημερώσουμε τα γονικά. Αν επιτρέψουμε τη διαγραφή παιδιών, θα πρέπει να ενημερώσουμε το γονικό κόμβο για να βεβαιωθούμε ότι η βάση δεδομένων διαθέτει συνέπεια. Σε αυτό το έργο, δεν θα προσθέσουμε τη δυνατότητα διαγραφής άρθρων και έτσι θα αποφύγουμε αυτό το πρόβλημα κατά το ήμισυ.

Θα κάνουμε άλλη μία βελτιστοποίηση: θα διαχωρίσουμε τα σώματα των μηνυμάτων από άλλα δεδομένα και θα τα αποθηκεύσουμε σε ένα ξεχωριστό πίνακα. Αυτό θα γίνει γιατί αυτό το όρισμα θα έχει τύπο `text` της MySQL. Αν έχουμε αυτό τον τύπο σε ένα πίνακα, τα ερωτήματα του πίνακα γίνονται πιο αργά. Επειδή θα κάνουμε πολλά ερωτήματα για να δημιουργήσουμε τη δομή

του δέντρου, η επίδραση στην ταχύτητα θα είναι μεγάλη. Αν έχουμε τα σώματα των μηνυμάτων σε ένα ξεχωριστό πίνακα, μπορούμε να τα ανακαλέσουμε όταν ένας χρήστης θέλει να δει ένα συγκεκριμένο μήνυμα.

Η MySQL μπορεί να αναζητήσει εγγραφές συγκεκριμένου μεγέθους γρηγορότερα από τις εγγραφές με μεταβλητό μέγεθος. Αν πρέπει να χρησιμοποιήσουμε δεδομένα μεταβλητού μεγέθους, μπορούμε να βοηθήσουμε δημιουργώντας ευρετήρια στα πεδία, τα οποία θα χρησιμοποιηθούν για να κάνουμε αναζητήσεις στη βάση δεδομένων. Η δημιουργία ευρετηρίων παίρνει αρκετό χρόνο και τα δεδομένα στα φόρουμ μπορεί να αλλάζουν συνέχεια, άρα θα πρέπει να ξαναδημιουργούμε τα ευρετήρια μας συχνά.

Με βάση όλες αυτές τις ιδέες, βλέπουμε το αρχείο `create_database.sql` στη Λίστα 7.1 με το οποίο δημιουργούμε τη βάση δεδομένων για το φόρουμ.

Λίστα 7.1 `create_database.sql` SQL για τη δημιουργία βάσης δεδομένων συζήτησης

```
create database discussion;  
use discussion;  
create table header  
(  
  parent int not null,  
  poster char(20) not null,  
  title char(20) not null,  
  children int default 0 not null,  
  area int default 1 not null,  
  posted datetime not null,  
  postid int unsigned not null auto_increment primary key  
);  
create table body  
(  
  postid int unsigned not null primary key,  
  message text
```

Λίστα 7.1 Συνέχεια

);

grant select, insert, update, delete

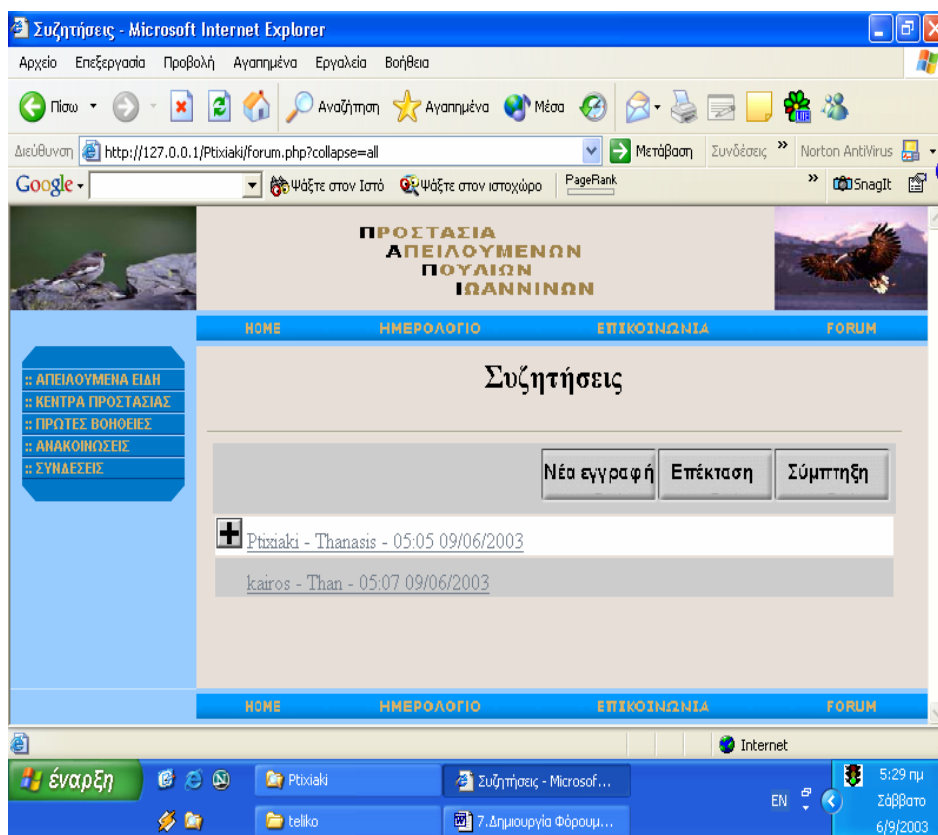
on discussion.*

to discussion@localhost identified by 'password';

Προβολή του Δέντρου των άρθρων

Στη συνέχεια, θα πρέπει να βρούμε ένα τρόπο για να παίρνουμε πληροφορίες από τη βάση δεδομένων και να τις αναπαραστήσουμε στη δομή δέντρου. Αυτό θα το κάνουμε στη βασική σελίδα forum.php. Για την παρουσίαση αυτή, έχουμε εισάγει μερικά δείγματα δημοσιεύσεων μέσω των script δημοσίευσης άρθρων new_post.php και store_new_post.php.

Η εικόνα 7.1 δείχνει την αρχική προβολή των άρθρων στην τοποθεσία όπως θα την βλέπει ο χρήστης.



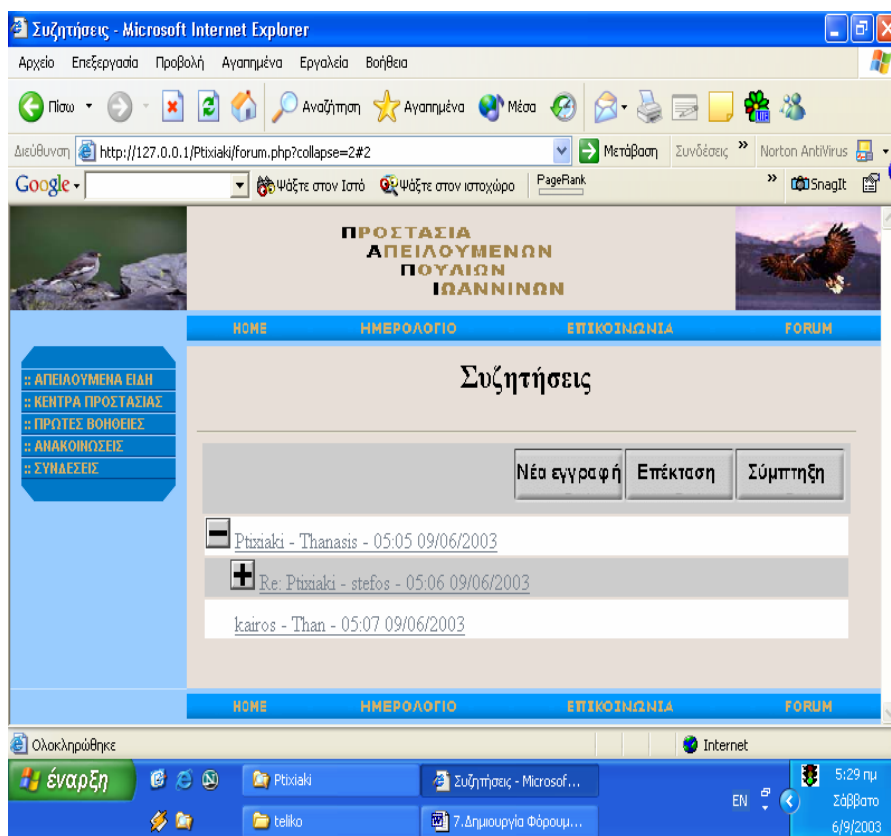
Εικόνα 7.1

Αρχική προβολή των άρθρων

Αυτά που βλέπουμε εδώ είναι όλα αρχικά άρθρα. Κανένα από αυτά δεν είναι απάντηση, είναι όλα τα πρώτα άρθρα σε ένα συγκεκριμένο θέμα.

Θα δείτε ότι έχουμε πολλές διαφορές. Υπάρχει μία γραμμή μενού, με την οποία θα προσθέσουμε μία νέα δημοσίευση και θα αναπτύσσουμε ή θα συμπύσσουμε την προβολή των άρθρων.

Για να καταλάβετε τι σημαίνει αυτό, κοιτάξτε τις δημοσιεύσεις. Κάποιες από αυτές έχουν ένα συν δίπλα τους. Αυτό σημαίνει ότι υπάρχουν απαντήσεις σε αυτά τα άρθρα. Για να δείτε τις απαντήσεις για ένα συγκεκριμένο άρθρο, μπορείτε να κάνετε κλικ πάνω στο συν. Το αποτέλεσμα αυτής της ενέργειας φαίνεται στην εικόνα 7.2.

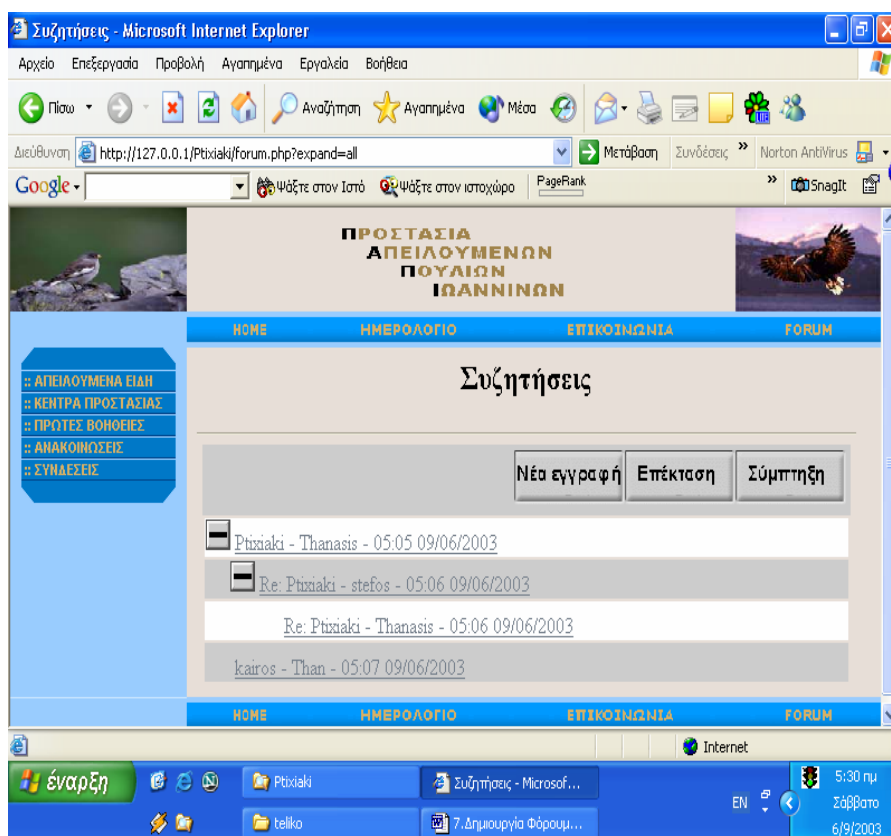


Εικόνα 7.2
Απαντήσεις για ένα συγκεκριμένο άρθρο

Όπως μπορείτε να δείτε, κάνοντας κλικ στο συν εμφανίζονται οι απαντήσεις σε αυτό το πρώτο άρθρο. Το συν μετατράπηκε σε μείων. Αν κάνουμε κλικ στο μείων, όλα τα άρθρα σε αυτή τη σειρά θα συμπτυχθούν και θα επανέλθουν στην αρχική προβολή.

Μπορείτε επίσης να παρατηρήσετε ότι μία από τις απαντήσεις έχει ένα συν δίπλα της. Αυτό σημαίνει ότι υπάρχουν απαντήσεις σε αυτή την απάντηση. Αυτό μπορεί να συνεχιστεί όσο θέλουμε και μπορείτε κάθε απάντηση κάνοντας κλικ στο κατάλληλο συν.

Οι δύο επιλογές από τη γραμμή μενού, το Expand και το Collapse, θα αναπτύξουν όλες τις σειρές και θα συμπύξουν όλες τις σειρές, αντίστοιχα. Στην εικόνα 7.3 βλέπουμε τα αποτελέσματα ενός κλικ Επέκταση.



Εικόνα 7.3
Τα αποτελέσματα ενός κλικ Επέκταση.

Επέκταση και σύμπτυξη

Ας δούμε πως γίνεται αυτό, κοιτάζοντας το script forum.php, που φαίνεται παρακάτω.

Λίστα 7.2 Το script forum.php

```
<?
include ('header.php');
include ('include_fns.php');
```

Λίστα 7.2 Συνέχεια

```
session_start();

// ελέγχουμε αν δημιουργήσαμε τη μεταβλητή συνοδού λειτουργίας
if(!session_is_registered('expanded'))
{
    $expanded = array();
    session_register('expanded');
}

// ελέγχουμε αν πατήθηκε το κουμπί επέκτασης
// η επέκταση μπορεί να αφορά όλα ή ένα postid ή τίποτα
if($expand)
{
    if($expand == 'all')
        expand_all($expanded);
    else
        $expanded[$expand] = true;
}

// ελέγχουμε αν πατήθηκε το κουμπί σύμπτυξης
// η σύμπτυξη μπορεί να αφορά όλα ή ένα postid ή τίποτα
if($collapse)
{
    if($collapse=="all")
        unset($expanded);
    else
        unset($expanded[$collapse]);
}

do_html_header("Discussion Posts");

display_index_toolbar();

// εμφάνιση της προβολής δέντρου συζητήσεων
display_tree($expanded);

do_html_footer();
include('footer.php ');
?>
```

Αυτό το script χρησιμοποιεί τρεις μεταβλητές συνοδού για να εκτελέσει το έργο του. Είναι οι εξής :

- Η μεταβλητή συνοδού λειτουργίας \$expanded η οποία παρακολουθεί τις σειρές που αναπτύσσονται. Αυτή μπορεί να διατηρηθεί από

προβολή σε προβολή και έτσι μπορούμε να αναπτύξουμε πολλές σειρές μαζί. Αυτή η μεταβλητή είναι ένας συσχετιζόμενος πίνακα, που περιέχει το postid των άρθρων των οποίων οι απαντήσεις θα εμφανιστούν αναπτυγμένες

- Η παράμετρος \$expand, που λέει στο script ποια νέα σειρά να αναπτύξει.
- Η παράμετρος \$collapse, που λέει στο script ποια νέα σειρά να αναπτύξει

Όταν κάνουμε κλικ σε ένα συν ή ένα μείον ή στο κουμπί Expand ή Collapse, θα ξανακληθεί το script forum.php με νέες παραμέτρους για το \$expand ή το \$collapse. Χρησιμοποιούμε το \$expanded από σελίδα σε σελίδα για να παρακολουθούμε ποιες σειρές θα πρέπει να αναπτυχθούν σε κάθε προβολή.

Το script ξεκινά αρχίζοντας μια σύνοδο λειτουργίας και προσθέτοντας τη μεταβλητή \$expanded ως μεταβλητή συνοδού λειτουργίας, αν αυτό δεν έχει γίνει ήδη.

Μετά από αυτό, το script ελέγχει αν έχει περάσει μια παράμετρος \$expanded ή \$collapse και τροποποιεί τον πίνακα \$expanded αντίστοιχα. Κοιτάξτε τον κώδικα για την παράμετρο \$expand:

```
if($expand)
{
  if($expand == 'all')
    expand_all($expanded);
  else
    $expanded[$expand] = true;
}
```

Αν κάνουμε κλικ στο κουμπί Expand, καλείται η συνάρτηση expand_all() για την προσθήκη όλων των σειρών που έχουν απαντήσεις στον πίνακα \$expanded.

Αν προσπαθήσουμε να αναπτύξουμε μία συγκεκριμένη σειρά, θα έχουμε περάσει ένα posted μέσω του \$expand. Επομένως προσθέτουμε ένα νέο στοιχείο στον πίνακα \$expanded.

Η συνάρτηση expand_all() φαίνεται στον παρακάτω πίνακα Λίστα (Λίστα 7.3)

Λίστα 7.3 Η συνάρτηση `expand_all()`

```
function expand_all(&$expanded)
{
  // σημειώνει να επεκταθούν όλες οι σειρές με παιδιά
  $conn = db_connect();
  $query = "select postid from header where children = 1";
  $result = mysql_query($query);
  $num = mysql_numrows($result);
  for($i = 0; $i<$num; $i++)
  {
    $expanded[mysql_result($result, $i, 0)]=true;
  }
}
```

Αυτή η συνάρτηση εκτελεί ένα ερώτημα στη βάση δεδομένων, για να δει ποια από τις σειρές του φόρουμ έχουν ως εξής:

```
select posted from header where children = 1
```

Καθένα από τα άρθρα που επιστρέφεται, προστίθεται στον πίνακα `$expanded`. Εκτελούμε αυτό το ερώτημα για να εξοικονομήσουμε χρόνο αργότερα. Θα μπορούσαμε απλώς να προσθέσουμε όλα τα άρθρα στην αναπτυγμένη λίστα, αλλά θα χάναμε χρόνο αν επεξεργαζόμασταν ερωτήσεις που δεν υπάρχουν.

Η σύμπτυξη των άρθρων εκτελείται με παρόμοιο, αλλά αντίθετο τρόπο, ως εξής:

```
if($collapse)
{
  if($collapse=="all")
    unset($expanded);
  else
    unset($expanded[$collapse]);
}
```

Μπορείτε να αφαιρέσετε αντικείμενα από τον πίνακα `$expanded`, αναιρώντας τα. Αφαιρούμε τη σειρά που πρόκειται να συμπυχθεί ή αφαιρούμε ολόκληρο τον πίνακα, αν πρόκειται να συμπυχθεί ολόκληρη η σελίδα.

Όλα αυτά είναι στο στάδιο της προ-επεξεργασίας, έτσι ώστε να ξέρουμε ποια άρθρα θα πρέπει να εμφανιστούν και ποια όχι. Το βασικό κομμάτι του script είναι η κλήση της

display_tree(\$expanded)

η οποία δημιουργεί στην πραγματικότητα, το δέντρο με τα άρθρα.

Εμφάνιση των άρθρων

Ας δούμε τη συνάρτηση `display_tree()`, που φαίνεται παρακάτω στη Λίστα 7.4

Λίστα 7.4 Η συνάρτηση `display_tree()`

```
function display_tree($expanded, $row = 0, $start = 0)
{
    // εμφάνιση της προβολής δέντρου με τις συζητήσεις

    global $table_width;
    echo "<table width = $table_width>";

    // κοιτάζουμε αν εμφανίζουμε ολόκληρη την λίστα ή μία υπολίστα
    if($start>0)
        $sublist = true;
    else
        $sublist = false;

    // κατασκευή δομής δέντρου για σύνοψη των συζητήσεων
    $tree = new treenode($start, " ", " ", 1, true, -1, $expanded, $sublist);

    // λέμε στο δέντρο να εμφανίσει τον εαυτό του
    $tree->display($row, $sublist);

    echo "</table>";
}
```

Ο βασικός ρόλος αυτής της συνάρτησης είναι η δημιουργία της ρίζας του δέντρου. Το χρησιμοποιούμε για να εμφανίσουμε ολόκληρο το ευρετήριο και να δημιουργήσουμε υπό-δέντρα από απαντήσεις στη σελίδα `view_post.php`. Χρειάζεται τρεις παραμέτρους. Η πρώτη, το `$expanded`, είναι η λίστα με τα `postid` των άρθρων, για την εμφάνιση τους σε ανεπτυγμένη μορφή. Η δεύτερη, η `$row`, είναι ένας δείκτης του αριθμού γραμμής που θα χρησιμοποιηθεί για να βρούμε τα εναλλασσόμενα χρώματα των γραμμών της λίστας.

Η τρίτη παράμετρος, η \$start, λέει στη συνάρτηση σε πιο σημείο να αρχίζει να εμφανίζει άρθρα. Αυτό είναι το postid της ρίζας του δέντρου που θα δημιουργηθεί και θα εμφανιστεί. Αν εμφανίσουμε ολόκληρο το δέντρο, όπως κάνουμε στην κύρια σελίδα, αυτό θα είναι 0, που σημαίνει ότι θα εμφανιστούν όλα τα άρθρα χωρίς το γονικό άρθρο. Αν αυτή η παράμετρος είναι 0, ορίζουμε το \$sublist σε false και εμφανίζουμε ολόκληρο το δέντρο.

Αν η παράμετρος είναι μεγαλύτερη από 0, τη χρησιμοποιούμε ως ρίζα του δέντρου που εμφανίζεται, ορίζουμε το \$sublist σε true και δημιουργούμε και εμφανίζουμε μόνο ένα κομμάτι του δέντρου.

Το πιο σημαντικό πράγμα που κάνει αυτή η συνάρτηση, είναι η δημιουργία ενός στιγμιότυπου της κλάσης treenode, η οποία αντιπροσωπεύει τη ρίζα του δέντρου. Αυτό δεν είναι ένα στην πραγματικότητα ένα άρθρο, αλλά συμπεριφέρεται ως γονικό άρθρο όλων των άρθρων πρώτου επιπέδου, που δεν έχουν γονικό άρθρο. Αφού δημιουργηθεί το δέντρο, απλώς καλούμε τη συνάρτηση εμφάνισης για να εμφανίσουμε τη λίστα με τα άρθρα.

Χρήση της κλάσης treenode

Ο κώδικας της κλάσης treenode φαίνεται παρακάτω στη Λίστα 7.5

Λίστα 7.5 Η κλάση treenode

```
<?
// οι συναρτήσεις για την φόρτωση, κατασκευή και εμφάνιση
// του δέντρου βρίσκονται σε αυτό το αρχείο

class treenode
{
// κάθε κόμβος του δέντρου έχει μεταβλητές μέλη που περιέχουν
// όλα τα δεδομένα για τη δημοσίευση εκτός από το μήνυμα
var $m_postid;
var $m_title;
var $m_poster;
var $m_posted;
var $m_children;
var $m_childlist;
var $m_depth;

function treenode($postid, $title, $poster, $posted, $children,
                 $expand, $depth, $expanded, $sublist)
{
// η συνάρτηση δημιουργίας ορίζει της μεταβλητές μέλη, αλλά το πιο
```

Λίστα 7.5 Συνέχεια

```

// σημαντικό δημιουργεί αναδρομικά τα κατώτερα τμήματα δέντρου
$this->m_postid = $postid;
$this->m_title = $title;
$this->m_poster = $poster;
$this->m_posted = $posted;
$this->m_children = $children;
$this->m_childlist = array();
$this->m_depth = $depth;

// μας νοιάζει μόνο τι είναι κάτω από τον κόμβο, αν έχει
// παιδιά και είναι σημειωμένος για επέκταση
// οι υπολίστες εκτείνονται πάντα
if(($sublist||$expand) && $children)
{
    $conn = db_connect();

    $query = "select * from header where parent = $postid order by
posted";
    $result = mysql_query($query);

    for ($count=0; $row = @mysql_fetch_array($result); $count++)
    {
        if($sublist||$expanded[ $row['postid'] ] == true)
            $expand = true;
        else
            $expand = false;
        $this->m_childlist[$count]= new treenode($row['postid'],$row['title'],
            $row['poster'],$row['posted'],
            $row['children'], $expand,
            $depth+1, $expanded, $sublist);
    }
}

function display($row, $sublist = false)
{
    // καθώς αυτό είναι ένα αντικείμενο, είναι υπεύθυνο για εμφάνιση του
    // εαυτού του

    // το $row μας λέει σε ποια γραμμή της εμφάνισης βρισκόμαστε
    // ώστε να ξέρουμε ποιο χρώμα θα έχει

    // το $sublist μας λέει αν είμαστε στην κύρια σελίδα
    // ή στη σελίδα μηνυμάτων. Οι σελίδες μηνυμάτων θα πρέπει να έχουν
    // το $sublist = true.
    // σε μία sublist, όλα τα μηνύματα επεκτείνονται και δεν υπάρχουν
    // σύμβολα "+" ή "-" .

```

Λίστα 7.5 Συνέχεια

```

// αν αυτός είναι ένας κόμβος, τον αγνοούμε
if($this->m_depth>-1)
{
// γραμμές με εναλλακτικά χρώματα
echo "<tr><td bgcolor = ";
if ($row%2)
echo "'#cccccc'>";
else
echo "'#ffffff'>";

// η εσοχή δείχνει το βάθος της ένθεσης
for($i = 0; $i<$this->m_depth; $i++)
{
echo "<img src = 'images/spacer.gif' height = 22
width = 22 alt = " valign = bottom>";
}

// εμφάνιση του + ή - ή ενός κενού
if ( !$sublist && $this->m_children && sizeof($this->m_childlist))
// είμαστε στην κύρια σελίδα, υπάρχουν κάποια παιδιά και είναι
// επεκταμένα
{
// είναι επεκταμένα – εμφάνιση κουμπιού σύμπτυξης
echo "<a href = 'forum.php?collapse="
$this->m_postid."#$this->m_postid'
><img src = 'images/minus.gif' valign = bottom
height = 22 width = 22 alt = 'Collapse Thread' border = 0></a>";
}
else if(!$sublist && $this->m_children)
{
// είναι συμπτυγμένα – εμφάνιση κουμπιού επέκτασης
echo "<a href = 'forum.php?expand="
$this->m_postid."#$this->m_postid'><img src = 'images/plus.gif'
height = 22 width = 22 alt = 'Expand Thread' border = 0></a>";
}
else
{
// δεν υπάρχουν παιδιά, ή είναι σε μία υπολίστα, να μην εμφανιστεί
// κουμπί
echo "<img src = 'images/spacer.gif' height = 22 width = 22
alt = " valign = bottom>";
}

echo " <a name = $this->m_postid ><a href =
'view_post.php?postid=$this->m_postid'>$this->m_title -
$this->m_poster - ".reformat_date($this->m_posted)."</a>";
echo "</td></tr>";

```

Λίστα 7.5 Συνέχεια

```

    // αύξηση μετρητή γραμμής σε εναλλακτικά χρώματα
    $row++;
  }
  // κλήση εμφάνισης στα παιδιά αυτού του κόμβου
  // ο κόμβος θα έχει παιδιά μόνο αν η λίστα του είναι επεκταμένη
  $num_children = sizeof($this->m_childlist);
  for($i = 0; $i<$num_children; $i++)
  {
    $row = $this->m_childlist[$i]->display($row, $sublist);
  }
  return $row;
}
};
?>

```

Αυτή η κλάση περιέχει τις λειτουργίες για την προβολή δέντρου αυτής της εφαρμογής

Ένα στιγμιότυπο της κλάσης `treenode` περιέχει πληροφορίες για μία μόνο δημοσίευση και συνδέσεις για όλες τις δημοσιεύσεις απαντήσεις αυτής της κλάσης. Έτσι έχουμε τις παρακάτω μεταβλητές μέλη:

```

var $m_postid;
var $m_title;
var $m_poster;
var $m_posted;
var $m_children;
var $m_childlist;
var $m_depth;

```

Παρατηρήστε ότι το `treenode` δεν περιέχει το σώμα του άρθρου. Δεν χρειάζεται να το φορτώσουμε, παρά μόνο αν ο χρήστης πάει στο `script view_post.php`. Θα πρέπει να το κάνουμε αυτό σχετικά γρήγορα, καθώς χρειαζόμαστε πολλά δεδομένα προκειμένου να εμφανίσουμε το δέντρο και θα πρέπει να κάνουμε ξανά υπολογισμούς όταν γίνεται ανανέωση της σελίδας ή όταν πατιέται ένα κουμπί.

Η συνάρτηση δημιουργίας διαμορφώνει τις τιμές όλων των μεταβλητών, ως εξής:

```

function treenode($postid, $title, $poster, $posted, $children,
                 $expand, $depth, $expanded, $sublist)
{
  // η συνάρτηση δημιουργίας ορίζει της μεταβλητές μέλη, αλλά το πιο
  // σημαντικό δημιουργεί αναδρομικά τα κατώτερα τμήματα δέντρου
  $this->m_postid = $postid;
  $this->m_title = $title;
  $this->m_poster = $poster;
  $this->m_posted = $posted;
  $this->m_children = $children;
  $this->m_childlist = array();
  $this->m_depth = $depth;

```

Όταν κατασκευάζουμε τη ρίζα `treenode` από την `display_tree()` της κύριας σελίδας, δημιουργούμε στην πραγματικότητα, ένα πρόχειρο κόμβο, στον οποίο θα συνδέονται άρθρα. Του περνάμε μερικές αρχικές τιμές:

```
$tree = new treenode($start, "", "", 1, true, -1, $expanded, $sublist)
```

Αυτό δημιουργεί ένα κόμβο ρίζας με μηδενικό `$postid`. Αυτό μπορεί να χρησιμοποιηθεί για να βρεθούν όλες οι δημοσιεύσεις πρώτου επιπέδου, γιατί έχουν γονικό άρθρο ίσο με μηδέν. Ορίζουμε το βάθος σε `-1`, γιατί αυτός ο κόμβος δεν θα εμφανιστεί. Όλες οι δημοσιεύσεις πρώτου επιπέδου έχουν βάθος ίσο με μηδέν και βρίσκονται στο αριστερό άκρο της οθόνης. Τα επόμενα βάθη πηγαίνουν προς τα δεξιά.

Το πιο σημαντικό που συμβαίνει σε αυτή την κατασκευή, είναι ότι δημιουργούνται οι κόμβοι παιδιά αυτού του κόμβου. Ξεκινάμε αυτή τη διαδικασία ελέγχοντας αν πρέπει να επεκτείνουμε τους κόμβους παιδιά. Αυτή τη διαδικασία την εκτελούμε μόνο αν ο κόμβος έχει κόμβους παιδιά και αν έχουμε επιλέξει να τα εμφανίσουμε:

```

if(($sublist||$expand) && $children)
{
  $conn = db_connect();

```

Στη συνέχεια συνδεόμαστε με τη βάση δεδομένων και καλούμε όλες τις εξαρτημένες δημοσιεύσεις ως εξής:

```

$query = "select * from header where parent = $postid order by posted";
$result = mysql_query($query);

```

Στη συνέχεια συμπληρώνουμε τον πίνακα `$m_childlist` με στιγμιότυπα της κλάσης `treenode`, που περιέχουν τις απαντήσεις στη δημοσίευση σε αυτό το `treenode`, ως εξής:

```

for ($count=0; $row = @mysql_fetch_array($result); $count++)
{

```



```

if($sublist||$expanded[ $row['postid'] ] == true)
  $expand = true;
else
  $expand = false;
$this->m_childlist[$count]= new treenode($row['postid'],$row['title'],
  $row['poster'],$row['posted'],
  $row['children'], $expand,
  $depth+1, $expanded, $sublist);

```

Αυτή η τελευταία γραμμή θα δημιουργήσει τα νέα treenode, ακολουθώντας ακριβώς τη διαδικασία που περιγράψαμε, αλλά για το επόμενο επίπεδο δέντρου. Αυτό είναι το κομμάτι που επαναλαμβάνεται αναδρομικά. Ένας γονικός κόμβος προκαλεί την κατασκευή του treenode, περνώντας το δικό του postid ως γονέας και προσθέτοντας ένα στο δικό του επίπεδο, πριν το περάσει.

Θα δημιουργηθεί κάθε treenode και αυτό με τη σειρά του, θα δημιουργήσει τα παιδιά του μέχρι να μην έχουμε άλλες απαντήσεις ή φύλλα που να θέλουμε να αναπτυχθούν.

Αφού γίνουν όλα αυτά, καλούμε τη συνάρτηση εμφάνιση της ρίζας treenode ως εξής:

```
$tree->display($row, $sublist);
```

Η συνάρτηση display() ξεκινά ελέγχοντας αν πρόκειται για πρόχειρη ρίζα:

```
if($this->m_depth>-1)
```

Με αυτόν τον τρόπο, η πρόχειρη ρίζα μπορεί να παραλειφθεί από την εμφάνιση. Όμως δεν θέλουμε να προσπεράσουμε τελείως τον κόμβο της ρίζας. Δεν θέλουμε να εμφανιστεί, αλλά θα πρέπει να ενημερώσει τα παιδιά του ότι πρέπει να εμφανίσουν τους εαυτούς τους.

Η συνάρτηση ξεκινά τότε το σχεδιασμό ενός πίνακα, που θα περιέχει τα άρθρα. Χρησιμοποιούμε τον τελεστή υπολοίπου (%) για να αποφασίσει τι χρώμα φόντου θα πρέπει να έχει αυτή η γραμμή:

```
// γραμμές με εναλλακτικά χρώματα
echo "<tr><td bgcolor = ";
if ($row%2)
  echo "#cccccc">";
else
  echo "#ffffff">";
```

Στη συνέχεια, χρησιμοποιεί τη μεταβλητή μέλος \$m_depth για να βρει το μέγεθος της εσοχής του τρέχοντος αντικειμένου. Αν ξανακοιτάξετε τις εικόνες

θα δείτε ότι, όσο μεγαλύτερο είναι το μέγεθος μιας απάντησης, τόσο μεγαλύτερη είναι η εσοχή. Αυτό γίνεται ως εξής:

```
// η εσοχή δείχνει το βάθος της ένθεσης
for($i = 0; $i<$this->m_depth; $i++)
{
    echo "<img src = 'images/spacer.gif' height = 22
        width = 22 alt = " valign = bottom>";
}
```

Το επόμενο τμήμα της συνάρτησης βρίσκει αν πρέπει να τοποθετήσει ένα κουμπί συν ή πλην ή και τίποτα:

```
// εμφάνιση του + ή - ή ενός κενού
if ( !$sublist && $this->m_children && sizeof($this->m_childlist))
// είμαστε στην κύρια σελίδα, υπάρχουν κάποια παιδιά και είναι
// επεκταμένα
{
// είναι επεκταμένα - εμφάνιση κουμπιού σύμπτυξης
echo "<a href = 'forum.php?collapse="
    $this->m_postid."#$this->m_postid'
    ><img src = 'images/minus.gif' valign = bottom
    height = 22 width = 22 alt = 'Collapse Thread' border = 0></a>";
}
else if(!$sublist && $this->m_children)
{
// είναι συμπτυγμένα - εμφάνιση κουμπιού επέκτασης
echo "<a href = 'forum.php?expand="
    $this->m_postid."#$this->m_postid"><img src = 'images/plus.gif'
    height = 22 width = 22 alt = 'Expand Thread' border = 0></a>";
}
else
{
// δεν υπάρχουν παιδιά, ή είναι σε μία υπολίστα, να μην εμφανιστεί
// κουμπί
echo "<img src = 'images/spacer.gif' height = 22 width = 22
    alt = " valign = bottom>";
}
}
```

Στη συνέχεια, εμφανίζουμε τις πληροφορίες για αυτόν τον κόμβο

```
echo " <a name = $this->m_postid ><a href =
    'view_post.php?postid=$this->m_postid'>$this->m_title -
    $this->m_poster - ".reformat_date($this->m_posted)."</a>";
echo "</td></tr>";
```

Αλλάζουμε το χρώμα της επόμενης γραμμής

```
// αύξηση μετρητή γραμμής σε εναλλακτικά χρώματα
$row++;
}
// κλήση εμφάνισης στα παιδιά αυτού του κόμβου
// ο κόμβος θα έχει παιδιά μόνο αν η λίστα του είναι επεκταμένη
$num_children = sizeof($this->m_childlist);
```

```

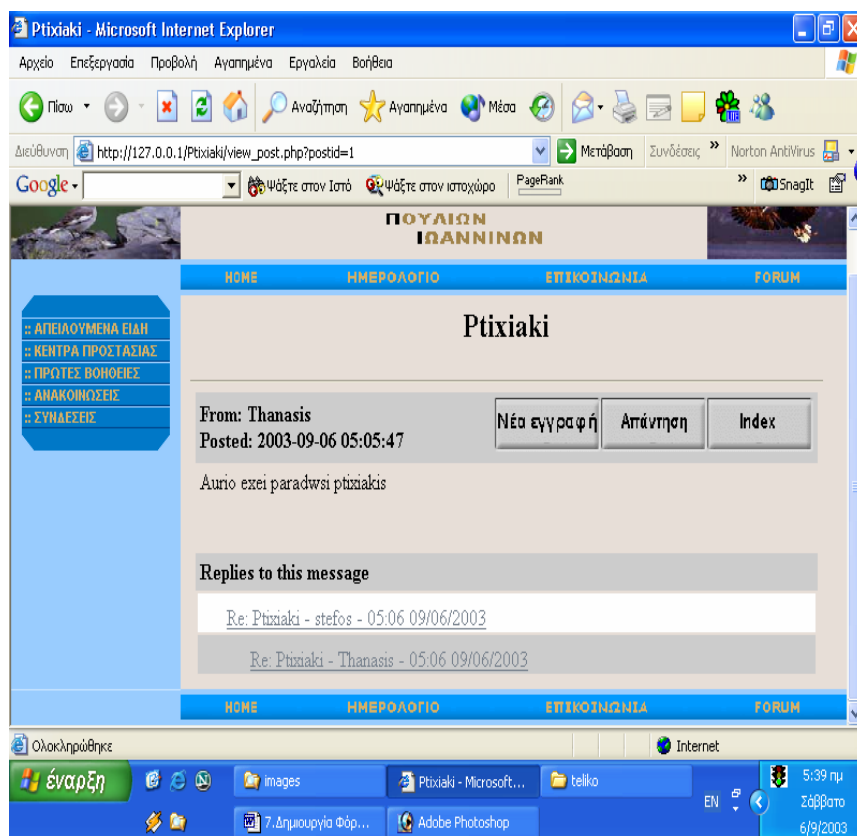
for($i = 0; $i<$num_children; $i++)
{
    $row = $this->m_childlist[$i]->display($row, $sublist);
}
return $row;

```

Και πάλι αυτή είναι μία αναδρομική κλήση συνάρτησης, που καλεί καθένα από τα παιδιά του κόμβου, για να εμφανίσουν τον εαυτό τους. Τους περνάμε το τρέχον χρώμα γραμμής και αυτά το παίρνουν με τη σειρά τους όταν τελειώσουν, έτσι ώστε να παρακολουθούμε το εναλλασσόμενο χρώμα.

Προβολή Συγκεκριμένων Άρθρων

Η `display_tree()` μας δίνει τελικά συνδέσεις για ένα σύνολο από άρθρα. Αν κάνουμε κλικ σε ένα από αυτά τα άρθρα, θα πάμε στο script `view_post.php`, με την παράμετρο `posted` του άρθρου που θα προβληθεί. Στην εικόνα 7.4 φαίνεται ένα δείγμα εξόδου από αυτό το script.



Εικόνα 7.4

Δείγμα εξόδου από αυτό το script `display_tree()`

Αυτό το script μας δείχνει το σώμα του μηνύματος, καθώς και τις απαντήσεις στο μήνυμα αυτό. Θα δείτε ότι οι απαντήσεις εμφανίζονται και

πάλι ως δέντρο, αλλά έχουν αναπτυχθεί πλήρως αυτή τη φορά και δεν έχουν κουμπιά συν και πλην. Αυτό είναι το αποτέλεσμα του διακόπτη \$sublist.

Ο κώδικας για το view_post.php, φαίνεται στη Λίστα 7.6

Λίστα 7.6 Το script view_post.php

```
<?
include ('header.php');
// συμπερίληψη συναρτήσεων βιβλιοθηκών
include ('include_fns.php');

// λήψη πληροφοριών δημοσίευσης
$post = get_post($postid);

do_html_header($post["title"]);

// εμφάνιση δημοσίευσης
display_post($post);

// αν η δημοσίευση έχει απαντήσεις, εμφάνιση της προβολής δέντρου
if($post['children'])
{
    echo "<br><br>";
    display_replies_line();
    display_tree($expanded, 0, $postid);
}

do_html_footer();
include ('footer.php');
?>
```

Το script χρησιμοποιεί τρεις βασικές κλήσεις συναρτήσεων για να εκτελεστεί η λειτουργία του: get_post(), display_post() και display_tree().

Η συνάρτηση `get_post()` παίρνει τις πληροφορίες της συνάρτησης από τη βάση δεδομένων. Ο κώδικας για αυτή τη συνάρτηση φαίνεται στη Λίστα 7.7

Λίστα 7.7 Η συνάρτηση `get_post()`

```
function get_post($postid)
{
    // εξαγωγή μίας δημοσίευσης από τη βάση και επιστροφή ως πίνακα

    if(!$postid) return false;

    $conn = db_connect();

    // λήψη όλων των πληροφοριών κεφαλίδας από το 'header'
    $query = "select * from header where postid = $postid";
    $result = mysql_query($query);
    if(mysql_numrows($result)!=1)
        return false;
    $post = mysql_fetch_array($result);

    // λήψη μηνύματος από το σώμα και πρόσθεσε το στο προηγούμενο
    // αποτέλεσμα
    $query = "select * from body where postid = $postid";
    $result2 = mysql_query($query);
    if(mysql_numrows($result2)>0)
    {
        $body = mysql_fetch_array($result2);
        if($body)
        {
            $post['message'] = $body['message'];
        }
    }
    return $post;
}
```

Αυτή η συνάρτηση, χρησιμοποιώντας ένα `postid`, θα εκτελέσει τα δύο ερωτήματα για την ανάκληση της κεφαλίδας και του σώματος αυτής της δημοσίευσης και θα τοποθετήσει μαζί σε ένα συσχετιζόμενο πίνακα, τον οποίο στη συνέχεια επιστρέφει.

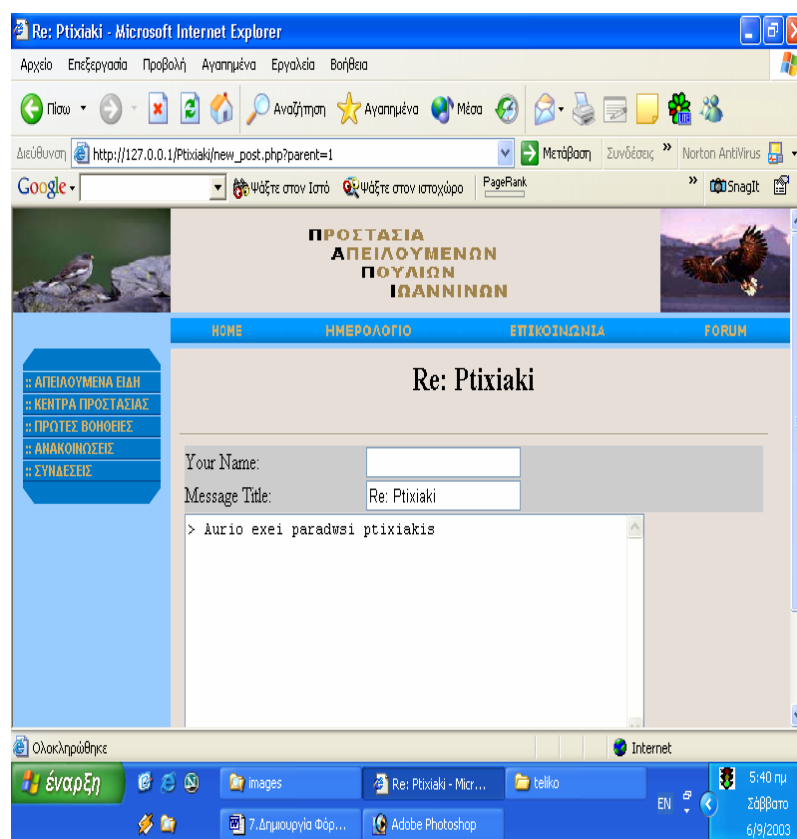
Τα αποτελέσματα αυτής της συνάρτησης περνούν στη συνέχεια στη συνάρτηση `display_post()` από το `output_fns.php`. Αυτή απλώς εμφανίζει τον πίνακα με HTML μορφοποίηση.

Τέλος το script `view_post.php` ελέγχει αν υπάρχουν απαντήσεις σε αυτό το άρθρο και καλεί την `display_tree()` για να εμφανίσει με μορφή υπό-λίστας, δηλαδή πλήρως ανεπτυγμένες, χωρίς συν και μείον.

Προσθήκη Νέων Άρθρων

Μετά από όλα αυτά, μπορούμε να δούμε πως προστίθεται μία νέα δημοσίευση στο φόρουμ. Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι να κάνει κλικ στο κουμπί `New Post` στη σελίδα `index` και ο δεύτερος είναι να κάνει κλικ στο κουμπί `Reply`, στη σελίδα `view_post.php`.

Και οι δύο αυτές ενέργειες, ενεργοποιούν το ίδιο script, το `new_post.php` αλλά με διαφορετικούς παραμέτρους. Η εικόνα 7.5 δείχνει την έξοδο από το `new_post.php` στο οποίο φτάσαμε πατώντας το κουμπί `Reply`.



Εικόνα 7.5
Η έξοδος από το κουμπί Απάντηση

Ας δούμε τον κώδικα που δίνει αυτή την έξοδο – φαίνεται στη λίστα 7.8

Λίστα 7.8 To script new_post.php

```
<?
include ('header.php');
include ('include_fns.php');

if(!$area)
    $area = 1;

if(!$error)
{
    if(!$parent)
    {
        $parent = 0;
        if(!$title)
            $title = "New Post";
    }
    else
    {
        // λήψη ονόματος δημοσίευσης
        $title = get_post_title($parent);

        // προσάρτηση του Re:
        if(strpos($title, "Re: ") == false )
            $title = "Re: ".$title;

        // βεβαιωθείτε ότι ο τίτλος χωρά στη βάση δεδομένων
        $title = substr($title, 0, 20);

        //προσαρτήστε εισαγωγικά στην δημοσίευση στην οποία απαντάτε
        $message = add_quoting(get_post_message($parent));
    }
}
do_html_header("$title");

display_new_post_form($parent, $area, $title, $message, $name);

if($error)
    echo "Your message was not stored. Make sure you have filled in all
fields and try again.";

do_html_footer();
include ('footer.php');
?>
```

Μετά από μερικές ρυθμίσεις, αυτό το script ελέγχει αν ο γονέας είναι μηδέν ή όχι. Αν είναι μηδέν πρόκειται για νέο θέμα και χρειάζεται επιπλέον δουλειά.

Αν αυτή είναι μία απάντηση, τότε το script προχωρά και ορίζει τον τίτλο και το κείμενο του αρχικού μηνύματος, ως εξής:

```
// get post name
$title = get_post_title($parent);

// append Re:
if(strpos($title, "Re: ") == false )
    $title = "Re: ".$title;

//make sure title will still fit in db
$title = substr($title, 0, 20);

//prepend a quoting pattern to the post you are replying to
$message = add_quoting(get_post_message($parent));
```

Οι συναρτήσεις που χρησιμοποιεί εδώ είναι οι `get_post_title()`, `get_post_message()` και `add_quoting()`. Αυτές οι συναρτήσεις βρίσκονται στη βιβλιοθήκη `discussion_fns.php`. Φαίνονται στη λίστα 7.9 και 7.10 και 7.11 αντίστοιχα.

Λίστα 7.9 Η συνάρτηση `get_post_title()`,

```
function get_post_title($postid)
{
    // εξαγωγή του μηνύματος μίας δημοσίευσης από τη βάση
    if(!$postid) return "";
    $conn = db_connect();

    //λήψη όλων των πληροφοριών κεφαλίδας από το 'header'
    $query = "select title from header where postid = $postid";
    $result = mysql_query($query);
    if(mysql_numrows($result)!=1)
        return "";
    return mysql_result($result, 0, 0);
}
```

Λίστα 7.10 Η συνάρτηση get_post_message()

```
function get_post_message($postid)
{
    // εξαγωγή του μηνύματος μίας δημοσίευσης από τη βάση δεδομένων

    if(!$postid) return "";

    $conn = db_connect();

    $query = "select message from body where postid = $postid";
    $result = mysql_query($query);
    if(mysql_numrows($result)>0)
    {
        return mysql_result($result,0,0);
    }
}
```

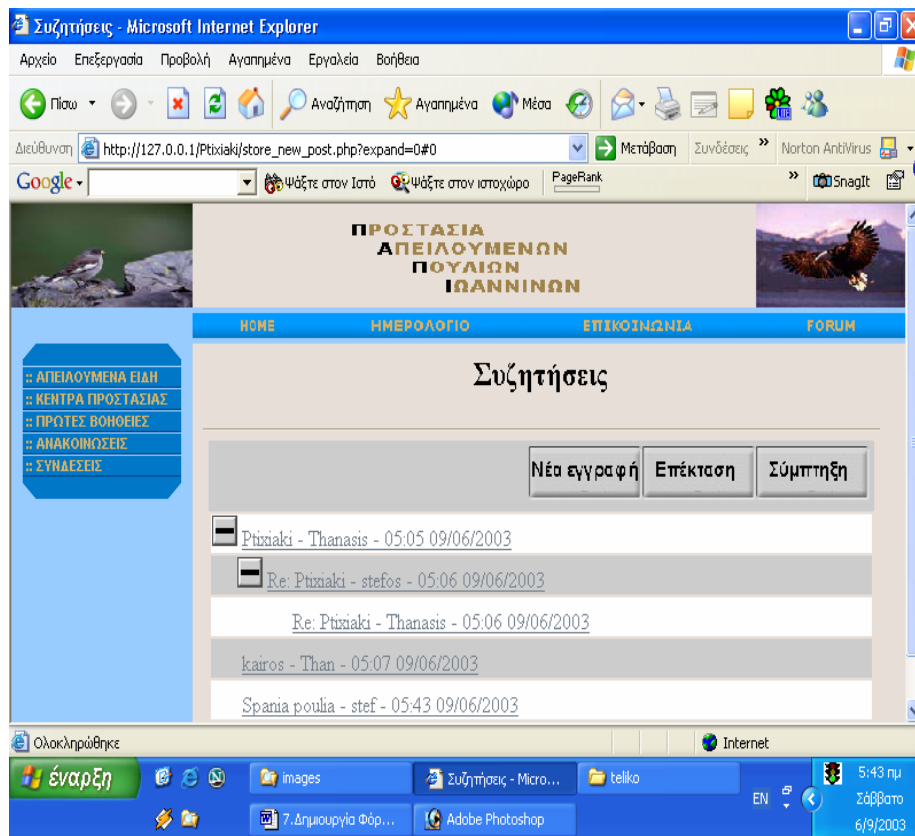
Οι δύο πρώτες συναρτήσεις ανακαλούν την κεφαλίδα και το σώμα ενός μηνύματος

Λίστα 7.11 Η συνάρτηση add_quoting().

```
function add_quoting($string, $pattern = "> ")
{
    // προσάρτηση εισαγωγικών για να σημειωθεί το κείμενο στο οποίο
    // απαντάτε
    return $pattern.str_replace("\n", "\n$pattern", $string);
}
```

Η συνάρτηση add_quoting() μορφοποιεί τη συμβολοσειρά, ώστε να ξεκινά το αρχικό κείμενο με ένα σύμβολο, το οποίο από προεπιλογή είναι το >.

Αφού εισάγει ο χρήστης την απάντηση του και κάνει κλικ στο κουμπί Post, θα μεταφερθεί στο script `store_new_post.php`. Ένα παράδειγμα της εξόδου από το script φαίνεται στην εικόνα 7.6.



Εικόνα 7.6

Παράδειγμα της εξόδου από το script `store_new_post.php`

Ο κώδικας του `store_new_post.php` φαίνεται στη Λίστα 7.12

Λίστα 7.12 το script `store_new_post.php`

```
<?
include ("include_fns.php");
if($id = store_new_post($HTTP_POST_VARS))
{
    include ("forum.php");
}
else
{
    $error = true;
```

Λίστα 7.12 Συνέχεια

```
include ("new_post.php");  
}  
  
?>
```

Όπως βλέπεται αυτό είναι ένα σύντομο script. Η βασική του λειτουργία είναι ότι καλεί τη συνάρτηση `store_new_post.php`. Αυτή η σελίδα δεν έχει δικά της οπτικά περιεχόμενα. Αν η αποθήκευση επιτύχει, βλέπουμε τη σελίδα `forum`. Διαφορετικά επιστρέφουμε στη σελίδα `new_post.php`, έτσι ώστε ο χρήστης να προσπαθήσει ξανά.

Η συνάρτηση `store_new_post()` φαίνεται στη λίστα 7.13

Λίστα 7.13 Η συνάρτηση `store_new_post()`

```
function store_new_post($post)  
{  
    // επικυρώνει και αποθηκεύει μία δημοσίευση  
  
    $conn = db_connect();  
    // έλεγχος ότι δεν υπάρχουν κενά πεδία  
    if(!filled_out($post))  
        return false;  
  
    $post = clean_all($post);  
  
    //έλεγχος ότι υπάρχει γονέας  
    if($post["parent"]!=0)  
    {  
        $query = "select postid from header where postid =  
        ".$post['parent']."";  
        $result = mysql_query($query);
```

Λίστα 7.13 Συνέχεια

```
if(mysql_numrows($result)!=1)
{
    return false;
}
}

// έλεγχος ότι δεν είναι διπλότυπη εγγραφή
$query = "select header.postid from header, body where
        header.postid = body.postid and
        header.parent = ".$post['parent']." and
        header.poster = ".$post['poster']."' and
        header.title = ".$post['title']."' and
        header.area = ".$post['area']."' and
        body.message = ".$post['message']."'";
$result = mysql_query($query);
if (!$result)
{
    return false;
}
if(mysql_numrows($result)>0)
    return mysql_result($result, 0, 0);

$query = "insert into header values
        (".$post['parent'].",
        ".$post['poster'].",
        ".$post['title'].",
        0,
        ".$post['area'].",
        now(),
        NULL
```

Λίστα 7.13 Συνέχεια

```
    )";  
    $result = mysql_query($query);  
    if (!$result)  
    {  
        return false;  
    }  
  
    // σημειώστε ότι ο γονέας μας έχει ένα παιδί τώρα  
    $query = "update header set children = 1 where postid =  
    ".$post['parent'];  
    $result = mysql_query($query);  
    if (!$result)  
    {  
        return false;  
    }  
  
    // find εύρεση του id της δημοσίευσης, όπου μπορούν να υπάρχουν  
    // πολλές κεφαλίδες που να είναι ίδιες, εκτός από το id και πιθανόν την  
    // ώρα δημοσίευσης  
    $query = "select header.postid from header left join body on  
header.postid = body.postid  
        where parent = ".$post["parent"]."  
        and poster = ".$post["poster"]."  
        and title = ".$post["title"]."  
        and body.postid is NULL";  
    $result = mysql_query($query);  
    if (!$result)  
    {  
        return false;  
    }  
    if(mysql_numrows($result)>0)
```

Λίστα 7.13 Συνέχεια

```
$id = mysql_result($result, 0, 0);  
if($id)  
{  
  
    $query = "insert into body values ($id, ".$post["message"].")";  
    $result = mysql_query($query);  
    if (!$result)  
    {  
        return false;  
    }  
  
    return $id;  
}  
  
}
```

Κεφάλαιο 8

Στο κεφάλαιο αυτό θα ασχοληθούμε με το πως ένας χρήστης θα μπορεί να μπει στη σελίδα και να εισάγει διάφορα στοιχεία στη βάση δεδομένων, να σβήσει διάφορα στοιχεία από τη βάση, να δημιουργήσει νέους χρήστες καθώς και να αλλάξει το password του και να κάνει logout

Χειρισμός του περιβάλλοντος διαχείρισης

Η πρώτη σελίδα που δημιουργήσαμε ονομάζεται login.php γιατί παρέχει στους χρήστες τη δυνατότητα να συνδεθούν στο σύστημα. Ο κώδικας της πρώτης σελίδας φαίνεται στη Λίστα 8.1

Λίστα 8.1 login.php

```
<?
require_once("bookmark_fns.php");
include "header.php";
do_html_header("login for administration");
```

Λίστα 8.1 Συνέχεια

```
display_login_form();
do_html_footer();
include "footer.php";
?>
```

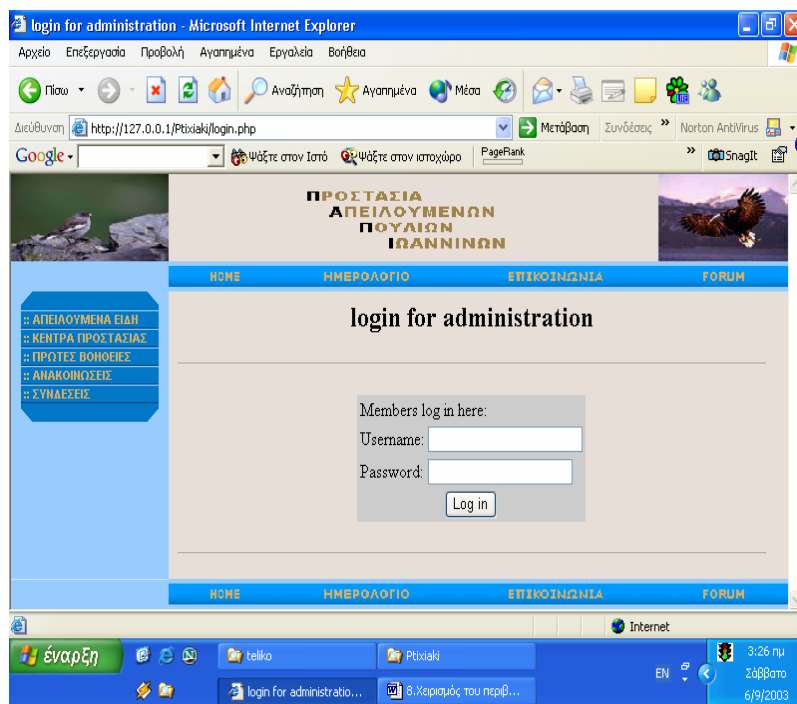
Αυτός ο κώδικας καλεί συναρτήσεις από τη συνάρτηση `bookmark_fns` που θα δημιουργήσουμε για αυτή την εφαρμογή. Κοιτάζοντας αυτό το αρχείο, βλέπουμε ότι συμπεριλαμβάνουμε ένα αρχείο (που περιέχει τις συναρτήσεις) και στη συνέχεια καλούμε ορισμένες συναρτήσεις για να εμφανίσουμε μία HTML κεφαλίδα, να εμφανίσουμε κάποια περιεχόμενα και να εμφανίσουμε ένα HTML υποσέλιδο.

Η έξοδος αυτού του script φαίνεται στην εικόνα 8.1

Οι συναρτήσεις του συστήματος συμπεριλαμβάνονται στο αρχείο `bookmark_fns.php` που φαίνεται στη Λίστα 8.2

Λίστα 8.2 `bookmark_fns.php`

```
<?
// Μπορούμε να συμπεριλάβουμε αυτό το αρχείο σε όλα τα αρχεία μας
// και με αυτόν τον τα τρόπο κάθε αρχείο θα περιέχει όλες τις συναρτήσεις
require_once("data_valid_fns.php");
require_once("db_fns.php");
require_once("user_auth_fns.php");
require_once("output_fns.php");
require_once("url_fns.php");
?>
```

**Εικόνα 8.1**

Η πρώτη σελίδα για να συνδεθεί σαν administrator

Όπως μπορείτε να δείτε, αυτό το αρχείο απλώς περιέχει τα πέντε άλλα αρχεία που θα χρησιμοποιήσουμε σε αυτή της εφαρμογή. Δημιουργήσαμε το αρχείο bookmark_fns.php, γιατί θα χρησιμοποιήσουμε τα περισσότερα από τα πέντε αρχεία συναρτήσεων στα περισσότερα script μας. Είναι ευκολότερο να συμπεριλάβουμε αυτό το αρχείο σε κάθε script, παρά να έχουμε πέντε εντολές include.

Στη συγκεκριμένη περίπτωση, χρησιμοποιούμε συναρτήσεις από το αρχείο output_fns.php. Αυτές είναι όλες απλές συναρτήσεις, που δίνουν ως αποτέλεσμα απλή HTML. Αυτό το αρχείο περιλαμβάνει τις τρεις συναρτήσεις που έχουμε χρησιμοποιήσει στο login.php, δηλαδή τις do_html_header(), display_login_form() και do_html_footer(), μεταξύ άλλων.

Ο κώδικας της do_html_header() φαίνεται στη Λίστα 8.3

Λίστα 8.3 Εμφανίζει την επικεφαλίδα της σελίδας

```
function do_html_header($title)
{
    // print an HTML header
    ?>
    <html>
```

Λίστα 8.3 Συνέχεια

```
<head>
  <title><?=$title?></title>
</head>
<body>
<?
  if($title)
    do_html_heading($title);
    echo "<hr>";
}
```

Το μόνο που κάνει αυτή η συνάρτηση είναι η προσθήκη του κατάλληλου τίτλου και της επικεφαλίδας στη σελίδα. Οι άλλες συναρτήσεις που έχουμε χρησιμοποιήσει στο login.php είναι παρόμοιες. Η συνάρτηση display_login_form() εμφανίζει τη φόρμα εγγραφής που φαίνεται στη εικόνα 8.1 και η do_html_footer() προσθέτει ένα τυπικό υποσέλιδο στη σελίδα.

Όταν ο χρήστης συμπληρώσει τα στοιχεία και κάνει κλικ στο κουμπί login καλείται η συνάρτηση member.php. που φαίνεται στην παρακάτω λίστα.

Λίστα 8.4 member.php

```
<?
include "header.php";

// include function files for this application
require_once("bookmark_fns.php");
session_start();

if ($username && $passwd)
// they have just tried logging in
{
  if (login($username, $passwd))
  {
    // if they are in the database register the user id
    $valid_user = $username;
    session_register("valid_user");
```

Λίστα 8.4 Συνέχεια

```
}
else
{
// unsuccessful login
do_html_header("Problem:");
echo "You could not be logged in.
    You must be logged in to view this page.";
do_html_url("login.php", "Login");
do_html_footer();
exit;
}
}

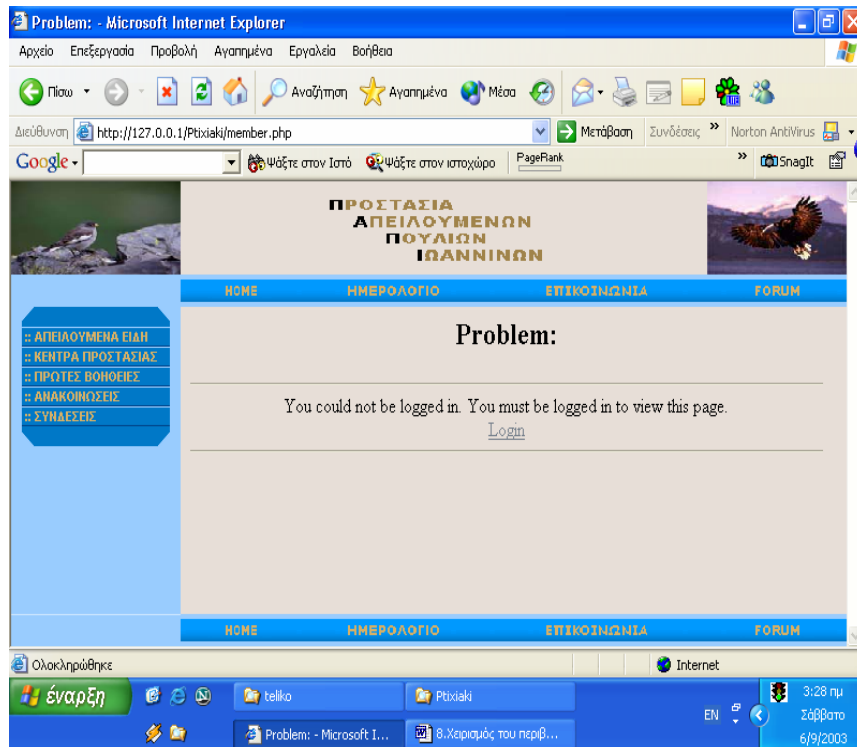
do_html_header("Home");
check_valid_user();

// give menu of options
display_user_menu();

do_html_footer();
include "footer.php";

?>
```

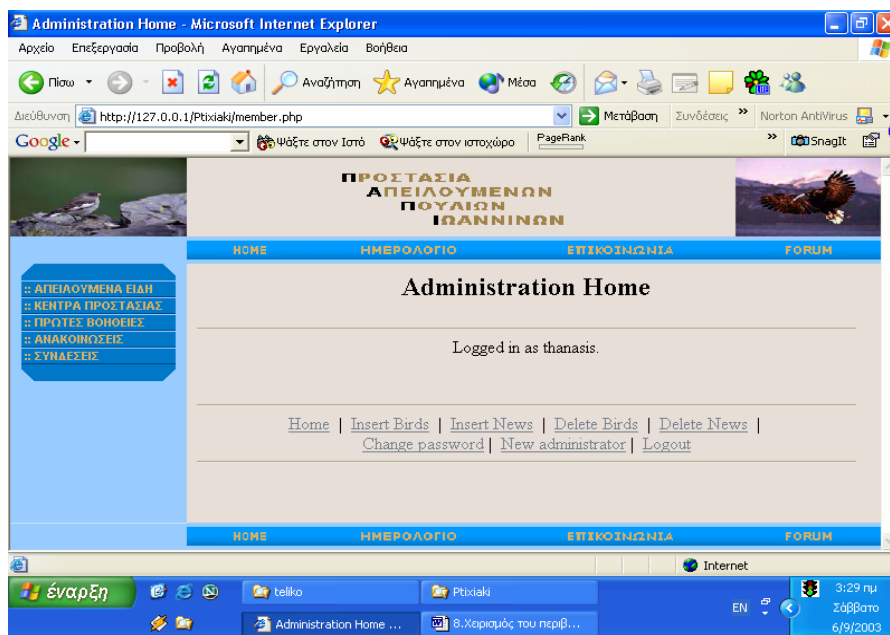
Αυτή η συνάρτηση κάνει έναν έλεγχο αν τα στοιχεία που έδωσε ο χρήστης είναι εγγεγραμμένα στη βάση δεδομένων και αν τα στοιχεία είναι σωστά. Αν ένα από τα στοιχεία που έδωσε ο χρήστης είναι λανθασμένα τότε ο χρήστης πηγαίνει σε μία σελίδα που τον πληροφορεί ότι δεν μπορεί να τον ανακατευθύνει στην κεντρική σελίδα διαχείρισης και του δίνει μία επιλογή να προσπαθήσει ξανά. Η σελίδα αυτή φαίνεται στην Εικόνα 8.2



Εικόνα 8.2

Η εικόνα που εμφανίζεται αν ο χρήστης δώσει λάθος στοιχεία

Αν τα στοιχεία είναι σωστά τότε εμφανίζεται η αρχική σελίδα διαχείρισης που φαίνεται στη εικόνα 8.3



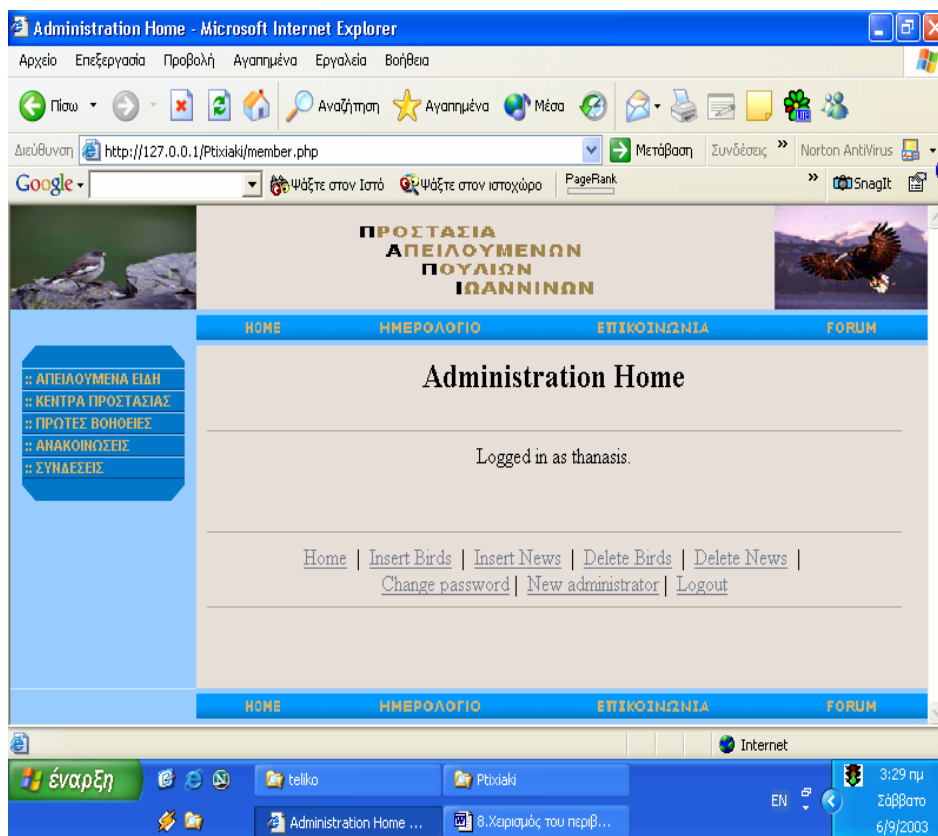
Εικόνα 8.3

Αρχική σελίδα διαχειρίσεις

Σε αυτή τη σελίδα εμφανίζονται όλες οι επιλογές, (σαν διαχειριστής πλέον), που έχει ο χρήστης.

Η πρώτη επιλογή είναι η Home με την οποία ο χρήστης σε οποιαδήποτε σελίδα διαχείρισης κι αν βρίσκεται, μπορεί πατώντας σε αυτό το κουμπί να επιστρέψει στην αρχική σελίδα διαχείρισης.

Πατώντας στη δεύτερη επιλογή καλείται το script insert_birds.php το οποίο εμφανίζει αυτό που πραγματικά έχει σημασία στη σελίδα διαχείρισης. Εμφανίζεται μια σελίδα στη οποία ο χρήστης μπορεί να εισάγει στοιχεία στη βάση δεδομένων. Στη προκειμένη περίπτωση να εισάγει ένα είδος πουλιού, εισάγοντας το όνομα του, τη φωτογραφία του και κάποιες πληροφορίες για το συγκεκριμένο πουλί. Αυτό φαίνεται στην εικόνα 8.4



Εικόνα 8.4

Σελίδα για να εισάγουμε στοιχεία στη βάση

Εισάγοντας τα στοιχεία στα πεδία που εμφανίζονται και κάνοντας κλικ στο Insert καλείται το script insert_birds_to_db.php το οποίο φαίνεται στη Λίστα 8.5

Λίστα 8.5

```
<?
require_once("bookmark_fns.php");
include "header.php";
do_html_header($bird_title);
if (!$bird_title || !$bird_picture || !$bird_text)
{
    echo "You have not entered all the required details.<br>"
        ."Please go back and try again.";
    exit;
}

$bird_title = addslashes($bird_title);
$bird_picture = addslashes($bird_picture);
$bird_text = addslashes($bird_text);
@ $db = mysql_pconnect("localhost", "thanstef", "ampelia");

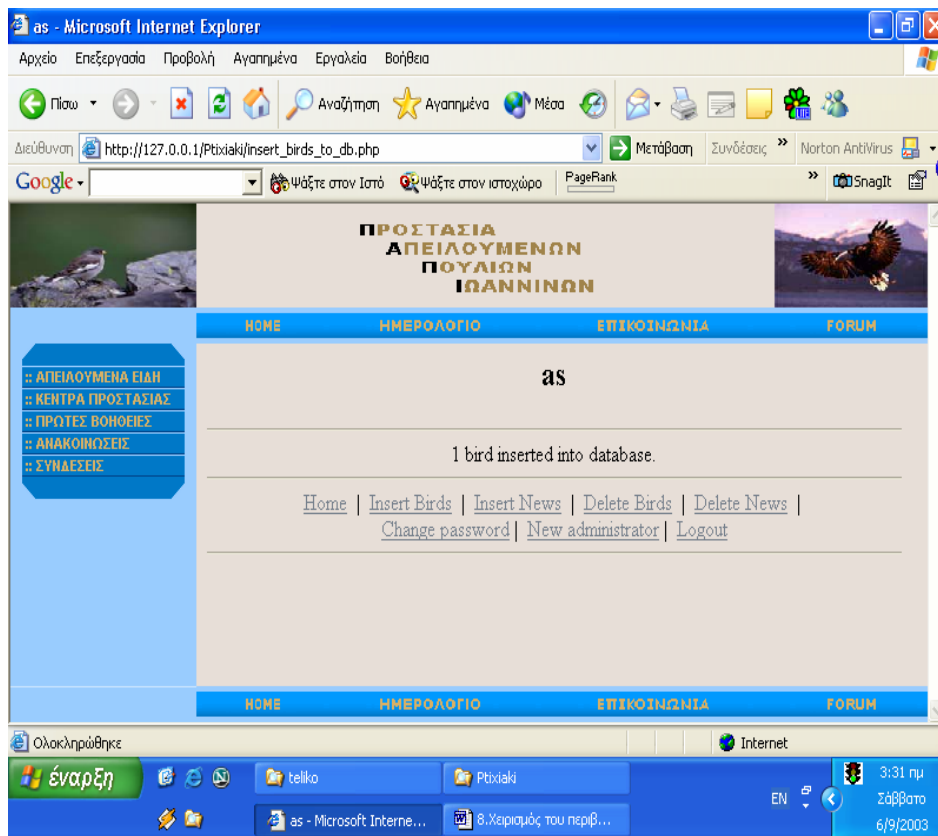
if (!$db)
{
    echo "Error: Could not connect to database. Please try again later.";
    exit;
}

mysql_select_db("birds");
$query = "insert into birds values ('','$bird_title.', '$bird_picture.',
'$bird_text.')";
$result = mysql_query($query);
if ($result)
    echo mysql_affected_rows()." bird inserted into database.";
display_user_menu();
include "footer.php";

?>
```

Αυτό το script κάνει ένα έλεγχο αν όλα τα στοιχεία στη προηγούμενη σελίδα έχουν συμπληρωθεί σωστά. Αν όχι, εμφανίζεται ένα μήνυμα λάθους ότι δεν έχουν συμπληρωθεί όλες οι απαραίτητες λεπτομέρειες και μας πληροφορεί να επιστρέψουμε στην προηγούμενη σελίδα για να συμπληρώσουμε όλες τις απαραίτητες λεπτομέρειες.

Στην περίπτωση που όλα τα στοιχεία έχουν συμπληρωθεί σωστά, εμφανίζεται μία σελίδα που μας πληροφορεί ότι ένα στοιχείο εισήχθηκε στη βάση δεδομένων. Αυτή η σελίδα φαίνεται στην εικόνα 8.5



Εικόνα 8.5
Εισαγωγή ενός στοιχείου στη βάση

Η επόμενη επιλογή είναι η Insert news. Η διαδικασία εισαγωγής νέων είναι παρόμοια με τη Insert Birds και δεν χρειάζεται περισσότερη εξήγηση.

Επόμενη επιλογή είναι η Delete Birds. Κάνοντας κλικ σε αυτή την επιλογή καλείται το αρχείο delete_birds.php. και εμφανίζεται ένας πίνακας με όλα τα πουλιά που υπάρχουν στη βάση δεδομένων. Αυτό φαίνεται στη εικόνα ...

Το αρχείο delete_birds.php φαίνεται στη Λίστα 8.6

Λίστα 8.6

<?

```
include "header.php";
```

Λίστα 8.6 Συνέχεια

```
require_once("output_fns.php");
do_html_header("Delete Birds");
@ $db = mysql_pconnect("localhost","thanstef","ampelia");

if (!$db)
{
    echo "error";
    exit;
}

mysql_select_db("birds");
$query = "select * from birds";
$result = mysql_query($query);
$num_results = mysql_num_rows($result);
echo "news found:" . $num_results;
?>
<form method=post action="delete_birds_from_db.php">
    <table bgcolor=#cccccc>

<?
for ($i=0; $i< $num_results; $i++)
{
    $row = mysql_fetch_array($result);
    echo("<tr>");
    echo "<td><input type=checkbox name=\"del_me[] \" value=\"\${row[bird_id]}\"";
";
    echo "</td>";
    echo "<td> \${row[bird_title]} ";
    echo("</td>");
    echo "</tr>";
```


Λίστα 8.6 Συνέχεια

}

?>

`<tr><td><input type=submit value="delete"></td></tr>``</table></form>`

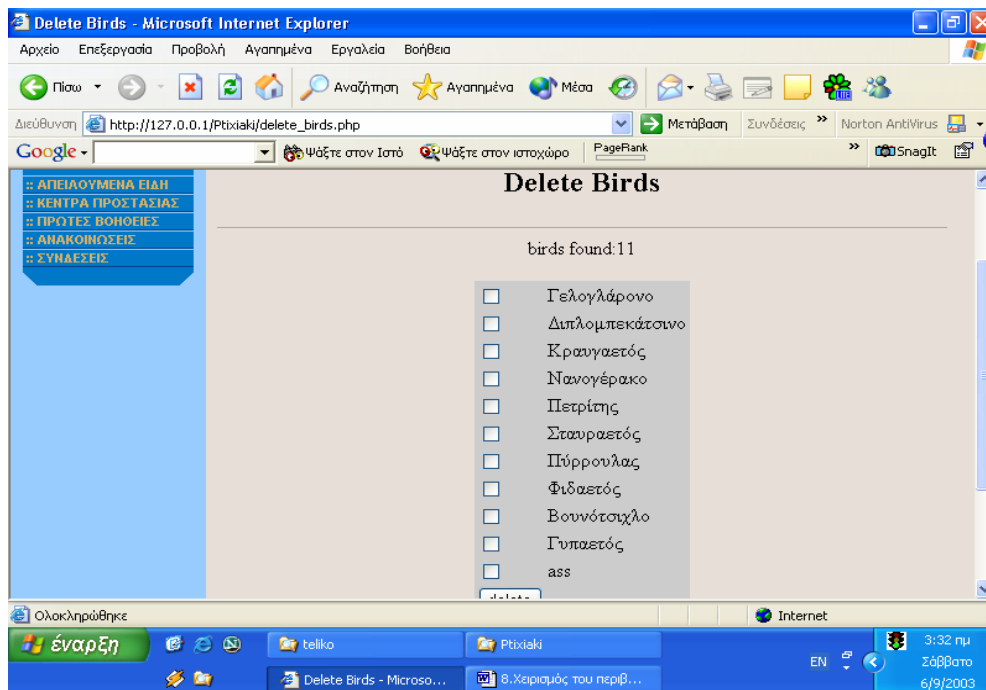
<?

`display_user_menu();``include "footer.php";`

?>

Η λειτουργία του script δεν χρειάζεται ιδιαίτερη εξήγηση εκτός από το σημείο που εμφανίζει τα checkboxes. Εκεί δημιουργείται ένας πίνακας (del_me[]) στον οποίο εισάγεται ο αριθμός bird_id ο οποίος είναι μοναδικός στη βάση δεδομένων για κάθε πουλί.

```
echo "<td><input type=checkbox name=\"del_me[]\" value=\"\${row[bird_id]}\"
```

**Εικόνα 8.6**

Διαγραφή ενός πουλιού από τη βάση

Όπως βλέπουμε, εμφανίζονται όλα τα πουλιά που είναι γραμμένα στη βάση με ένα checkbox μπροστά. Αυτό δίνει τη δυνατότητα στο χρήστη να επιλέξει ποιο πουλί θέλει να διαγράψει (από ένα μέχρι και όλα τα πουλιά). Αυτό φαίνεται στην εικόνα 8.6

Όταν επιλέξουμε ποιο πουλί θέλουμε να διαγράψουμε (μαρκάροντας το κατάλληλο checkbox), και κάνουμε κλικ στο κουμπί Delete, καλείται το αρχείο delete_birds_from_db.php.

Το αρχείο delete_birds_from_db.php φαίνεται στη λίστα 8.7

Λίστα 8.7 Το αρχείο delete_birds_from_db.php

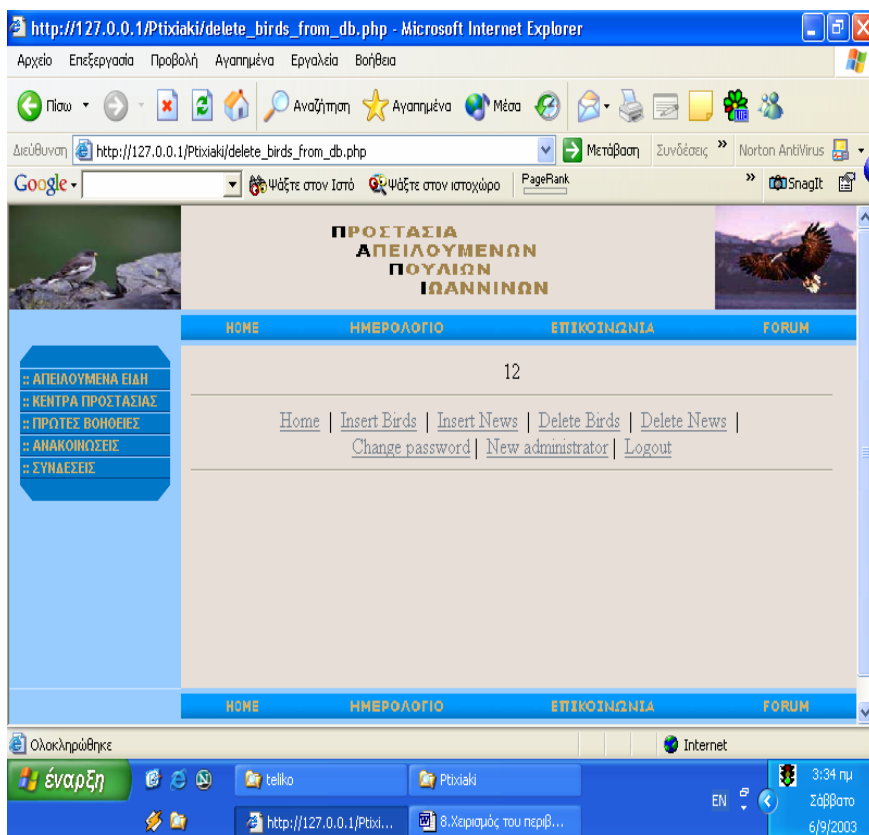
```
<?

include "header.php";
require_once("output_fns.php");
require_once ("url_fns.php");

foreach($del_me as $bird_id)
{
    if (delete_bm_birds($bird_id))
        echo $bird_id;
    else
        echo "Could not delete ";
}
display_user_menu();
include "footer.php";
?>
?>
```

Το αρχείο αυτό παίρνει σαν είσοδο το bird_id (που όπως είπαμε είναι μοναδικό) και καλεί τη συνάρτηση delete_bm_birds(). Αυτή με τη σειρά της

ανοίγει μία σύνδεση με τη βάση δεδομένων και διαγράφει το πουλί από τη βάση με αυτό το id. Τελικά το πουλί διαγράφεται από τη βάση και η σελίδα μας δίνει την πληροφορία ότι ένα πουλί διαγράφηκε. Αυτό φαίνεται στην εικόνα 8.7

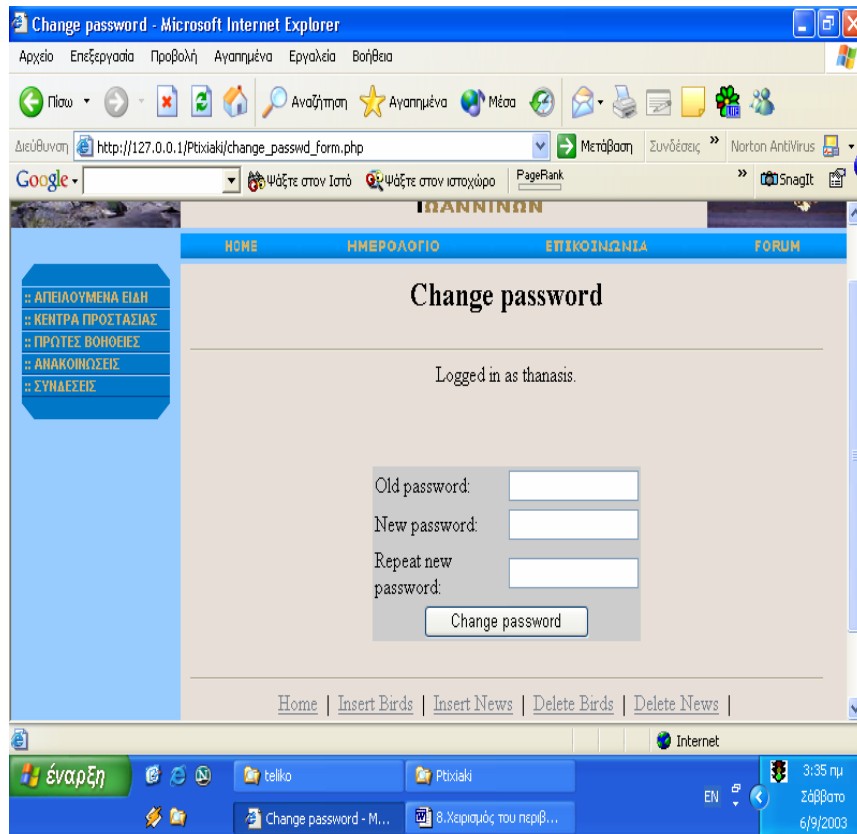


Εικόνα 8.7
Ένα πουλί διαγράφηκε

Με παρόμοιο τρόπο λειτουργεί και η Delete News και δεν μπαίνουμε στη διαδικασία να το εξηγήσουμε.

Αλλαγή κωδικών πρόσβασης

Το επόμενο κουμπί είναι το Change Password. Πατώντας σε αυτό το κουμπί ενεργοποιείται το αρχείο change_passwd_form.php. Αυτό το script καλεί της βιβλιοθήκης εξόδου για να εμφανίσει μια φόρμα στη οποία ο χρήστης μπορεί να αλλάξει το password. Αυτή η φόρμα φαίνεται στην εικόνα 8.8

**Εικόνα 8.8**

Το script `change_passwd_form.php` εμφανίζει μία φόρμα, όπου οι χρήστες μπορούν να αλλάξουν τους κωδικούς πρόσβασης

Όταν στέλνεται η φόρμα, ξεκινά το script `change_passwd.php` που φαίνεται στη Λίστα 8.8

Λίστα 8.8 το script `change_passwd.php`

```
<?
include "header.php";
require_once("bookmark_fns.php");
session_start();
do_html_header("Changing password");
check_valid_user();
if (!filled_out($_HTTP_POST_VARS))
{
    echo "You have not filled out the form completely.
    Please try again.";
```

Λίστα 8.8 Συνέχεια

```
display_user_menu();
do_html_footer();
    exit;
}
else
{
    if ($new_passwd!=$new_passwd2)
        echo "Passwords entered were not the same. Not changed.";
    else if (strlen($new_passwd)>16 || strlen($new_passwd)<6)
        echo "New password must be between 6 and 16 characters. Try again.";
    else
    {
        // προσπάθεια ενημέρωσης
        if (change_password($valid_user, $old_passwd, $new_passwd))
            echo "Password changed.";
        else
            echo "Password could not be changed.";
    }
}
display_user_menu();
do_html_footer();
include "footer.php";
?>
```

Αυτό το script ελέγχει αν ο χρήστης έχει συνδεθεί (χρησιμοποιώντας το `check_valid_user()`), αν έχει συμπληρώσει τη φόρμα για τον κωδικό πρόσβασης (χρησιμοποιώντας το `filled_out()`) και αν οι νέοι κωδικοί πρόσβασης είναι ίδιοι και έχουν το σωστό μέγεθος.

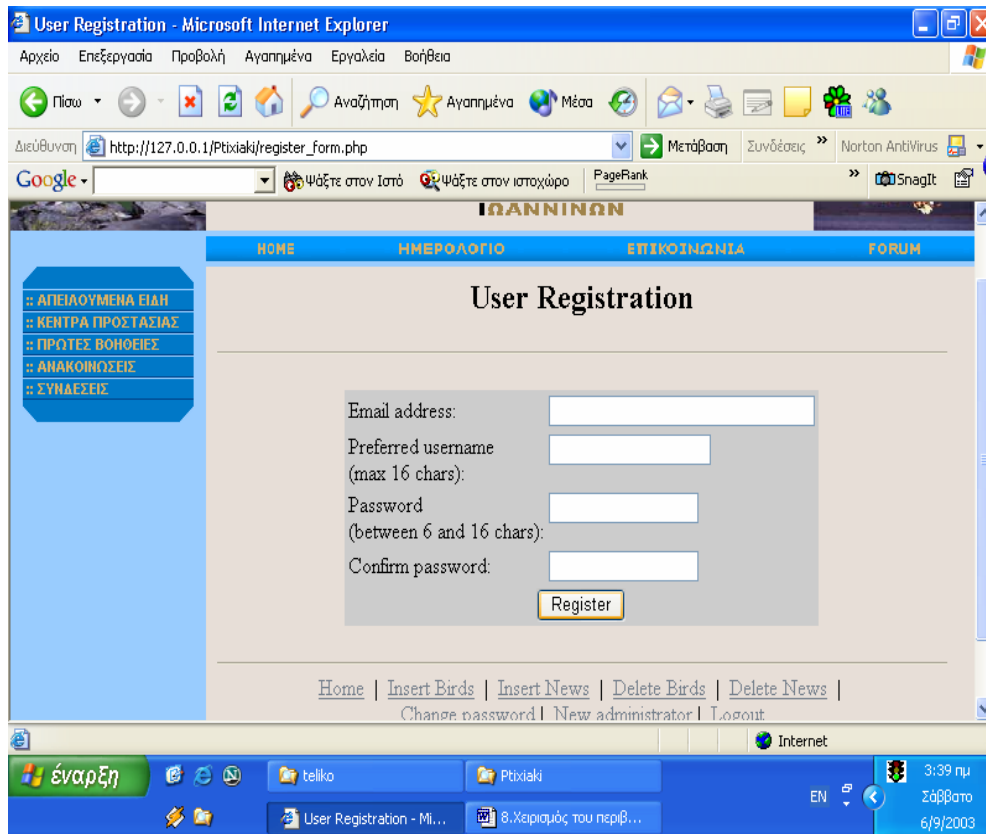
Η συνάρτηση `check_valid_user()` φαίνεται στη Λίστα ... και ελέγχει αν ο χρήστη είναι συνδεδεμένος ώστε να μπορεί να αλλάξει το password.

Λίστα 8.9 Η συνάρτηση `check_valid_user()`

```
function check_valid_user()
// βλέπει αν κάποιος είναι συνδεδεμένος και ειδοποιεί αν όχι
{
    global $valid_user;
    if (session_is_registered("valid_user"))
    {
        echo "Logged in as $valid_user.";
        echo "<br>";
    }
    else
    {
        // δεν είναι συνδεδεμένος
        do_html_heading("Problem:");
        echo "You are not logged in.<br>";
        do_html_url("login.php", "Login");
        do_html_footer();
        exit;
    }
}
```

Εγγραφή ενός νέου administrator

Για να εγγραφεί ένας νέος χρήστης, θα πρέπει να πάρουμε στοιχεία του μέσω μίας φόρμας και να τα εισάγουμε στη βάση δεδομένων. Όταν ο χρήστης κάνει κλικ στο κουμπί New Administrator στη σελίδα `login.php` θα μεταφερθεί σε μία φόρμα εγγραφής που δημιουργείται από το `register_form.php`. Η έξοδος από αυτό το αρχείο φαίνεται στην εικόνα 8.9

**Εικόνα 8.9**

Έξοδος από το script register_new.php

Σε αυτή τη σελίδα ο χρήστης, σαν administrator, μπορεί να δημιουργήσει κι άλλους χρήστες οι οποίοι θα μπορούν να διαχειρίζονται τη βάση. Συμπληρώνοντας τα στοιχεία και κάνοντας κλικ στο κουμπί register μπαίνουν σε λειτουργία διάφορες συνθήκες που έχουμε δημιουργήσει.

Πρώτη συνθήκη είναι η filled_out(). Αυτή η συνάρτηση ελέγχει αν η φόρμα έχει συμπληρωθεί. Η συνάρτηση filled_out() φαίνεται στη Λίστα 8.10

Λίστα 8.10 filled_out.php

```
function filled_out($form_vars)
{
    // έλεγχος ότι κάθε μεταβλητή έχει τιμή
    foreach ($form_vars as $key => $value)
    {
        if (!isset($key) || ($value == ""))
            return false;
    }
}
```

Λίστα 8.10 Συνέχεια

```
}  
    return true;  
}
```

Εφόσον ελεγχθεί ότι όλα τα στοιχεία είναι σωστά συμπληρωμένα ελέγχουμε αν η διεύθυνση ηλεκτρονικού ταχυδρομείου είναι σωστή. Αυτό γίνεται με την συνάρτηση `valid_email()` η οποία φαίνεται στη Λίστα 8.11

Λίστα 8.11

```
function valid_email($address)  
{  
    // ελέγχει αν η διεύθυνση ηλεκτρονικού ταχυδρομείου είναι έγκυρη  
    if (ereg("^[a-zA-Z0-9_\.\\-]+@[a-zA-Z0-9\\-]+\.[a-zA-Z0-9\\-\.]+$", $address))  
        return true;  
    else  
        return false;  
}
```

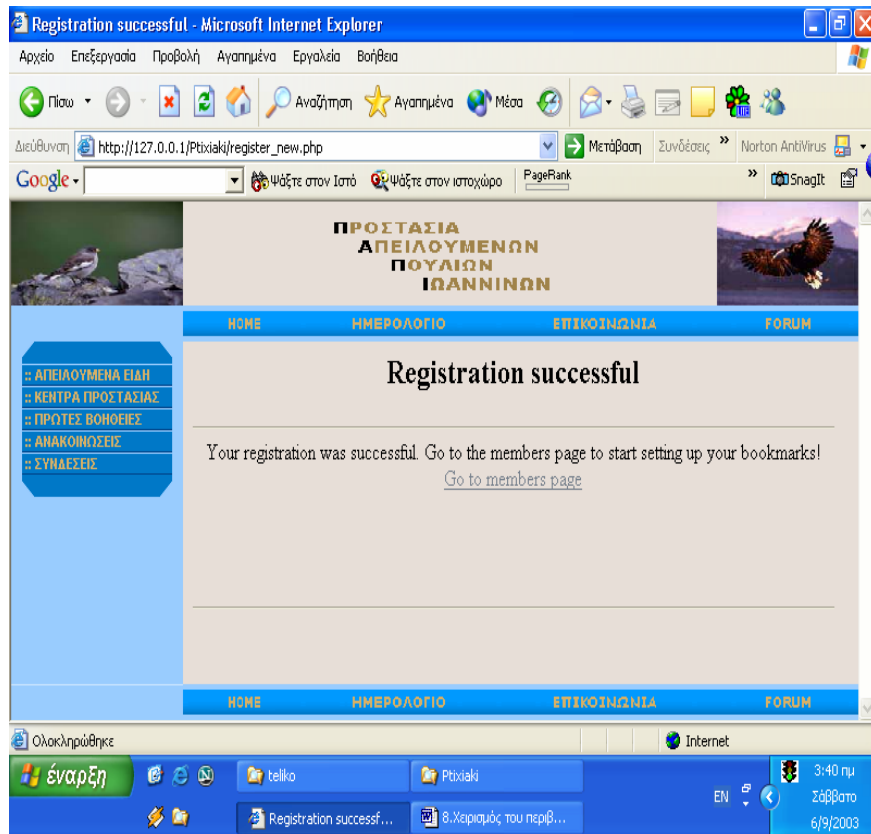
Τρίτο ελέγχουμε αν τα passwords που δόθηκαν ταιριάζουν μεταξύ τους με την εξής συνθήκη:

```
if ($passwd != $passwd2)
```

και τέλος ελέγχουμε αν ο κωδικός πρόσβασης έχει το κατάλληλο μέγεθος ως εξής:

```
if (strlen($passwd)<6 || strlen($passwd) >16)
```

Αν περάσει όλες τις συνθήκες τότε δημιουργείτε ένας νέος χρήστης με τα στοιχεία που δόθηκαν παραπάνω και εμφανίζεται μία εικόνα όπως αυτή της εικόνας 8.10



Εικόνα 8.10
Η εγγραφή ήταν επιτυχής

Τέλος υπάρχει ένα κουμπί logout το οποίο συνδέεται με το script `logout.php`. Ο κώδικας αυτού του script φαίνεται στη Λίστα 8.12

Λίστα 8.12 `logout.php`

```
<?
```

```
// συμπεριλαμβάνει τα αρχεία συναρτήσεων γι' αυτή την εφαρμογή
```

```
require_once("bookmark_fns.php");
```

```
session_start();
```

```
$old_user = $valid_user; // store to test if they *were* logged in
```

```
$result_unreg = session_unregister("valid_user");
```

```
$result_dest = session_destroy();
```

```
// αρχίζει η έξοδος html
```

```
do_html_header("Logging Out");
```

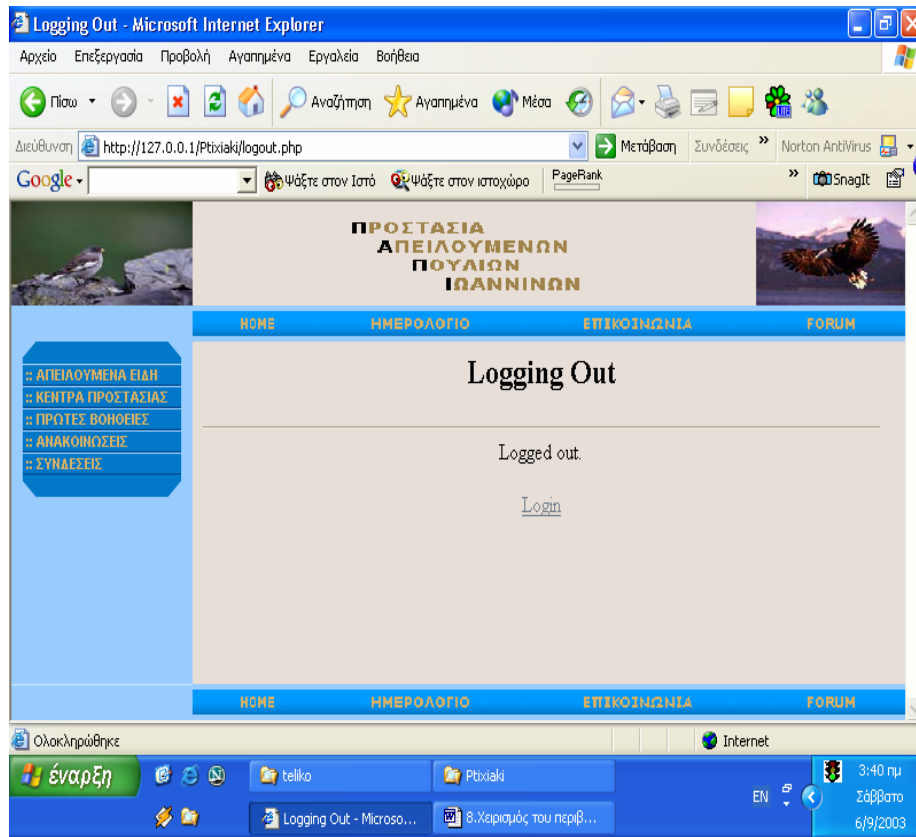
Λίστα 8.12 Συνέχεια

```
if (!empty($old_user))
{
    if ($result_unreg && $result_dest)
    {
        // αν ήταν συνδεδεμένος και τώρα αποσυνδέθηκε
        echo "Logged out.<br>";
        do_html_url("login.php", "Login");
    }
    else
    {
        // είναι συνδεδεμένος και δεν μπορεί να αποσυνδεθεί
        echo "Could not log you out.<br>";
    }
}
else
{
    // αν δεν είναι συνδεδεμένος και ήρθε στη σελίδα κάπως
    echo "You were not logged in, and so have not been logged out.<br>";
    do_html_url("login.php", "Login");
}

do_html_footer();

?>
```

Με αυτό το script ο χρήστης αποσυνδέεται και θα πρέπει να ξανακάνει τη διαδικασία login που περιγράψαμε πιο πριν ώστε να μπορέσει να κάνει της λειτουργίες που είχε σαν administrator. Η σελίδα που εμφανίζεται όταν ο χρήστης κάνει logout είναι αυτή που φαίνεται στην εικόνα 8.11



Εικόνα 8.11
Αποσύνδεση

Βιβλιογραφία

Για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκαν κάποια βιβλία από τα οποία πήρα κάποια ιδέα η και χρησιμοποιήθηκαν κάποια κομμάτια κώδικα από αυτά τα βιβλία. Τα βιβλία αυτά είναι:

- Πλήρες εγχειρίδιο της HTML 4 Laura Lemay
- PHP , MySQL και Apache Εκδόσεις Julie C. Meloni
- PHP Developers cookbook Sterling Hughes
- E-business & e-Commerce How to program Deitel,Deitel & Nieto
- MySQL-PHP Database Applications
- PHP Manual Stig Bakken
- Php-Mysql_tutorial Graeme Merrall
- Ανάπτυξη Web εφαρμογών με PHP και MySQL Εκδόσεις SAMS