



**ΑΤΕΙ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΣΤΕΦ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΠΟΛΥΜΕΣΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΘΕΜΑ: ΣΥΣΤΗΜΑ ΑΠΟΘΗΚΗΣ
«SUPER MARKET»**

**ΕΙΣΗΓΗΤΗΣ κ. ΚΛΕΙΣΑΡΧΑΚΗΣ ΜΙΧΑΗΛ
ΣΠΟΥΔΑΣΤΗΣ ΜΠΕΝΟΣ ΘΕΟΔΩΡΟΣ
Α.Μ 1177**

ΗΡΑΚΛΕΙΟ 2010

ΤΕΚΜΗΡΙΩΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΙΣΤΟΤΟΠΟΥ ΜΕ ΤΙΤΛΟ ΗΛΕΚΤΡΟΝΙΚΗ ΣΥΓΚΡΙΤΙΚΗ ΑΓΟΡΑ

ΠΕΡΙΕΧΟΜΕΝΑ

1. [ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ](#)
2. [ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ](#)
 1. [ΑΡΧΙΚΗ ΣΕΛΙΔΑ](#)
 2. [ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ](#)
 - 1) [ΚΑΤΑΣΤΗΜΑΤΑ](#)
 - 2) [ΚΑΤΗΓΟΡΙΕΣ](#)
 - 3) [ΠΡΟΪΟΝΤΑ](#)
 - 4) [ΠΑΡΑΓΓΕΛΙΕΣ](#)
 - 5) [ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ](#)
 - 6) [ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ](#)
 - 7) [ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ](#)
3. [ΑΝΑΛΥΣΗ ΒΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ](#)
 1. [ΠΙΝΑΚΕΣ ΒΑΣΗΣ](#)
 2. [ΠΑΡΟΥΣΙΑΣΗ ΠΙΝΑΚΩΝ](#)
 - 1) [Categories](#)
 - 2) [Customers](#)
 - 3) [orders](#)
 - 4) [Passlist](#)
 - 5) [Products](#)
 - 6) [Sessions](#)
 - 7) [Stores](#)
 3. [ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ](#)
4. [ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΠΡΟΓΡΑΜΜΑΤΟΣ](#)
 1. [ΠΕΡΙΓΡΑΦΗ ΚΟΙΝΟΤΥΠΩΝ ΤΜΗΜΑΤΩΝ ΚΩΔΙΚΑ](#)
 2. [ΠΕΡΙΓΡΑΦΗ ΑΡΧΕΙΟΥ products.php](#)
 3. [ΤΥΠΟΙ ΕΡΩΤΗΜΑΤΩΝ SQL](#)

1. [ΕΡΩΤΗΜΑ SELECT](#)
2. [ΕΡΩΤΗΜΑ INSERT](#)
3. [ΕΡΩΤΗΜΑ DELETE](#)
4. [ΕΡΩΤΗΜΑ UPDATE](#)
4. [ΣΥΝΔΕΣΗ JAVASCRIPT ΜΕ ΤΗΝ PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ urlencode](#)
5. [ΕΚΚΙΝΗΣΗ ΕΦΑΡΜΟΓΗΣ](#)
6. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEX.PHP](#)
7. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEXLEFT.PHP](#)
 1. [ΚΩΔΙΚΑΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΕΙΣΑΓΩΓΗΣ ΚΩΔΙΚΟΥ](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ PassKeyPress](#)
 2. [ΤΟ ΑΡΧΕΙΟ validate.php](#)
 2. [ΚΩΔΙΚΑΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΠΙΤΥΣΣΟΜΕΝΗΣ ΛΙΣΤΑΣ](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ FillProducts](#)
 2. [Η ΣΥΝΑΡΤΗΣΗ Compare](#)
 3. [Η ΣΥΝΑΡΤΗΣΗ FillStores](#)
 4. [Η ΣΥΝΑΡΤΗΣΗ FillCat](#)
8. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ COMPARE.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ AddItem](#)
 2. [ΤΟ ΑΡΧΕΙΟ additem.php](#)
9. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEXRIGHT.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ RemoveItem](#)
 2. [ΤΟ ΑΡΧΕΙΟ removeitem.php](#)
 3. [ΤΟ ΑΡΧΕΙΟ delorder.php](#)
10. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ CUSTFORM.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ SendOrder](#)
 2. [ΤΟ ΑΡΧΕΙΟ regcustomer.php](#)
11. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ ADMIN.PHP](#)
 1. [ΤΟ ΑΡΧΕΙΟ left.js](#)
12. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ STORES.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ Edit](#)
 2. [Η ΣΥΝΑΡΤΗΣΗ Delete](#)
 3. [ΤΟ ΑΡΧΕΙΟ delstor.php](#)
 4. [Η ΣΥΝΑΡΤΗΣΗ Save](#)

5. [Η ΣΥΝΑΡΤΗΣΗ UpdateData](#)
6. [ΤΟ ΑΡΧΕΙΟ upstor.php](#)
7. [Η ΣΥΝΑΡΤΗΣΗ SendData](#)
8. [ΤΟ ΑΡΧΕΙΟ regstor.php](#)
9. [Η ΣΥΝΑΡΤΗΣΗ EditProd](#)
10. [ΤΟ ΑΡΧΕΙΟ getrec.php](#)
13. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ ORDERS.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ back](#)
14. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ EMPTYDATABASE.PHP](#)
15. [ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ CREATEDATA.PHP](#)
 1. [Η ΣΥΝΑΡΤΗΣΗ rand_except](#)

ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

Ο σκοπός της εφαρμογής είναι η προβολή προϊόντων επιχειρήσεων και η πώληση τους μέσω διαδικτύου.

Από την πλευρά του ο χρήστης έχει την δυνατότητα να κάνει έρευνα αγοράς για το τι προϊόντα είναι διαθέσιμα σε κάθε επιχείρηση, καθώς και να προβεί σε σύγκριση τιμών μεταξύ αυτών.

Τα προϊόντα είναι καταχωρημένα ανά κατηγορία και κατάσταση.

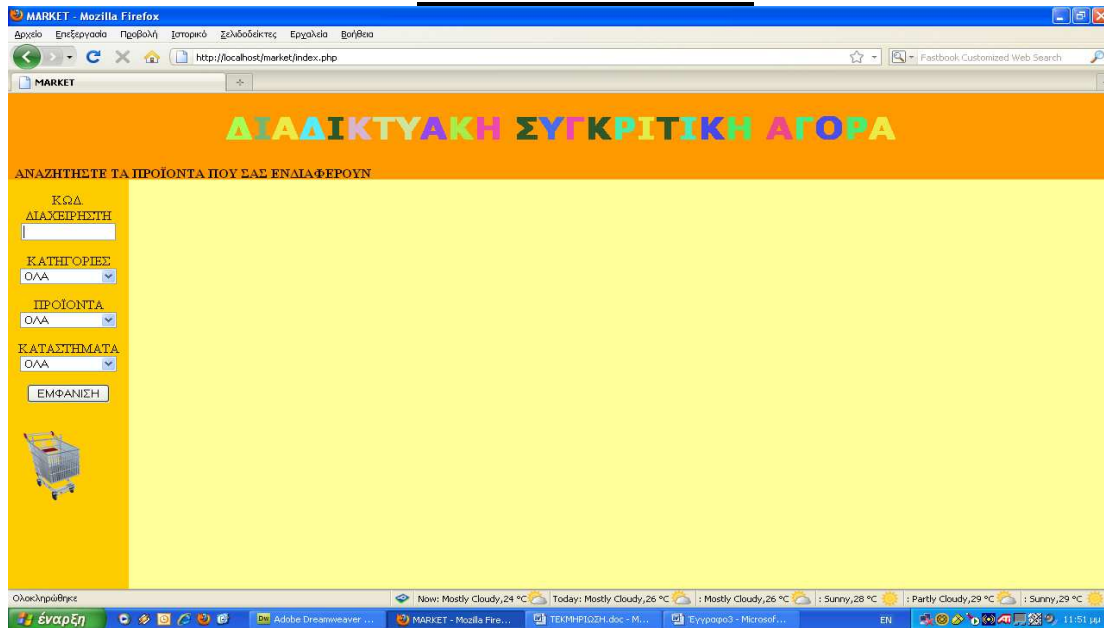
Ο χρήστης έχει επίσης την δυνατότητα να κάνει ηλεκτρονική παραγγελία και αγορά προϊόντων επιλέγοντας τα προϊόντα που θέλει και συμπληρώνοντας στην συνέχεια την φόρμα με τα στοιχεία του. Τα προϊόντα μπορεί να είναι από διαφορετικά καταστήματα.

Η εφαρμογή παρέχει τόσο στους εξουσιοδοτημένους υπευθύνους των καταστημάτων όσο και στον διαχειριστή του ιστοτόπου διαχειριστικά εργαλεία όπως καταχώρηση, αλλαγή και διαγραφή δεδομένων με ιεραρχία πρόσβασης που βασίζεται σε κωδικό εισόδου στην σελίδα διαχείρισης. Οι κωδικοί των υπευθύνων των καταστημάτων έχουν την δυνατότητα να επέμβουν μόνο σε προϊόντα που αφορούν αποκλειστικά το κατάστημα τους, ενώ ο διαχειριστής έχει πλήρη δικαιώματα πρόσβασης σε όλες τις λειτουργίες καθώς και την δυνατότητα καταχώρισης νέων καταστημάτων και χρηστών αλλά και διαγραφή όλων των δεδομένων τις βάσης και δημιουργία τυχαίων δεδομένων για λόγους επίδειξης.

ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ

Η λειτουργία της εφαρμογής βασίζεται σε δύο κύριες ιστοσελίδες που περιέχουν πλαίσια (FRAMES).

ΑΡΧΙΚΗ ΙΣΤΟΣΕΛΙΔΑ



Η πρώτη ιστοσελίδα απαρτίζεται από τέσσερα πλαίσια.

Το επάνω οριζόντιο πλαίσιο που καταλαμβάνει όλο το διαθέσιμο πλάτος περιλαμβάνει τον τίτλο του ιστοτόπου.

Το αριστερό πλαίσιο περιλαμβάνει ένα αντικείμενο εισαγωγής κειμένου τύπου κωδικού (Password), τρία πλαίσια πτυσσόμενης λίστας επιλογής (Combo Boxes) και ένα κουμπί αποστολής αιτημάτων (Push Button).

Στο αντικείμενο εισαγωγής κωδικού εισάγεται ο κωδικός πρόσβασης αλλά στη θέση του κωδικού εμφανίζονται αστεράκια για λόγους ασφαλείας.

Στη συνέχεια ο κωδικός πρόσβασης αποστέλλεται στον εξυπηρέτη (SERVER), ελέγχεται με τους αντίστοιχους που είναι καταχωρημένοι στη βάση και αναλόγως ο χρήστης αποκτά τα αντίστοιχα δικαιώματα πρόσβασης και προωθείται στη δεύτερη ιστοσελίδα που γίνεται η διαχείριση των δεδομένων και της βάσης του προγράμματος.

Το πρώτο από τα πτυσσόμενα πλαίσια επιλογής απεικονίζει τις διαθέσιμες κατηγορίες, το δεύτερο τα διαθέσιμα προϊόντα και το τρίτο τα διαθέσιμα καταστήματα.

Η προβολή των διαθέσιμων κατηγοριών εξαρτάται από την επιλογή του πλαισίου καταστημάτων και η προβολή των προϊόντων από τις επιλογές των πλαισίων κατηγοριών και καταστημάτων.

Αν σε κάποια επιλογή ή αν δεν υπάρχουν καταχωρημένα δεδομένα, δεν υπάρχει αντίστοιχο προϊόν ή κατηγορία, το πλαίσιο προβάλλει την επιλογή «ΔΕΝ ΥΠΑΡΧΟΥΝ».

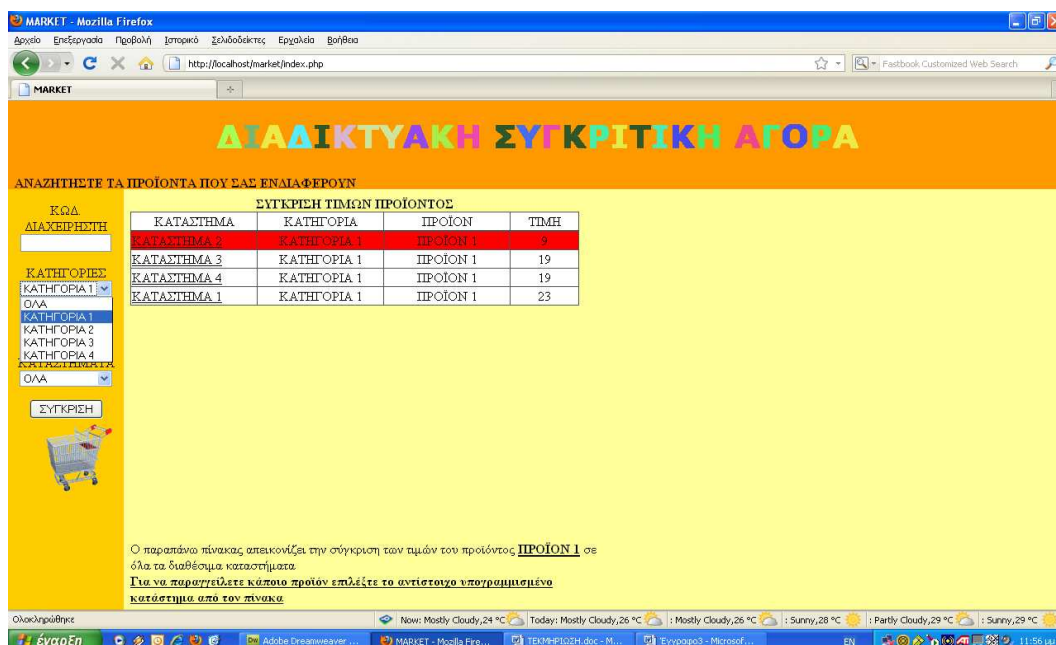
Υπάρχει επιπλέον εκτός από τις διαθέσιμες προβολές στα πλαίσια λίστας και η επιλογή «ΟΛΑ» που σημαίνει ότι κατά την υποβολή αιτήματος αναζήτησης ή σύγκρισης, να συμπεριληφθούν όλα τα δεδομένα της ή των εν λόγω κατηγοριών (Κατηγορίες, Προϊόντα, Καταστήματα).

Κατά την φόρτωση της ιστοσελίδας, αν υπάρχουν δεδομένα στη βάση, όλα τα πλαίσια έχουν αρχική επιλογή την τιμή «ΟΛΑ» και το κουμπί την ένδειξη «ΕΜΦΑΝΙΣΗ».

Αν επιλεγεί συγκεκριμένη κατηγορία και συγκεκριμένο προϊόν και η επιλογή καταστημάτων είναι «ΟΛΑ» τότε το κουμπί παίρνει την ένδειξη «ΣΥΓΚΡΙΣΗ».

Κατά την αποστολή αιτήματος με την ένδειξη «ΕΜΦΑΝΙΣΗ» γίνεται απλή προβολή δεδομένων ενώ με την ένδειξη «ΣΥΓΚΡΙΣΗ» γίνεται προβολή δεδομένων με αύξουσα σειρά ταξινόμησης με καλύτερη τιμή προϊόντος ανά κατάστημα και η πρώτη επιλογή προβάλλεται με διαφορετικό χρώμα.

Οι προβολές και οι συγκρίσεις προϊόντων εμφανίζονται στο κεντρικό πλαίσιο με τη μορφή πίνακα.



MARKET - Mozilla Firefox

http://localhost/market/index.php

ΔΙΑΔΙΚΤΥΑΚΗ ΣΥΓΚΡΙΤΙΚΗ ΑΓΟΡΑ

ΑΝΑΖΗΤΗΣΤΕ ΤΑ ΠΡΟΪΟΝΤΑ ΠΟΥ ΣΑΣ ΕΝΔΙΑΦΕΡΟΥΝ

ΚΩΔ
ΔΙΑΣΦΕΡΣΗ

ΚΑΤΗΓΟΡΙΕΣ
ΚΑΤΗΓΟΡΙΑ 1
ΟΛΑ

ΚΑΤΗΓΟΡΙΑ 1
ΚΑΤΗΓΟΡΙΑ 2
ΚΑΤΗΓΟΡΙΑ 3
ΚΑΤΗΓΟΡΙΑ 4
ΟΛΑ

ΣΥΓΚΡΙΣΗ

ΣΥΓΚΡΙΣΗ ΤΙΜΩΝ ΠΡΟΪΟΝΤΟΣ			
ΚΑΤΑΣΤΗΜΑ	ΚΑΤΗΓΟΡΙΑ	ΠΡΟΪΟΝ	ΤΙΜΗ
ΚΑΤΑΣΤΗΜΑ 2	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 1	9
ΚΑΤΑΣΤΗΜΑ 3	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 1	19
ΚΑΤΑΣΤΗΜΑ 4	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 1	19
ΚΑΤΑΣΤΗΜΑ 1	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 1	23

Ο παραπάνω πίνακας απεικονίζει την σύγκριση των τιμών των προϊόντος **ΠΡΟΪΟΝ 1** σε όλα τα διαθέσιμα καταστήματα.
Για να παραγγείλετε κάποιο προϊόν επιλέξτε το αντίστοιχο υπογραμμισμένο κατάστημα από τον πίνακα

Now: Mostly Cloudy, 24 °C Today: Mostly Cloudy, 26 °C ! Mostly Cloudy, 26 °C ! Sunny, 28 °C ! Partly Cloudy, 29 °C ! Sunny, 29 °C

Στο δεξιό πλαίσιο, το οποίο είναι κρυφό, όταν επιλεγεί κάποιο προϊόν, εμφανίζεται και το προσθέτει στην λίστα παραγγελιών.

Αν κάποιο προϊόν επιλεγεί περισσότερο από μια φορά τότε αυξάνεται η ποσότητα του επιλεγμένου προϊόντος. Αντίστοιχα αν επιλεγεί ένα προϊόν από τη λίστα

παραγγελιών μειώνεται η ποσότητα του και αν αυτή ήταν ένα τότε διαγράφεται από τη λίστα. Αν η λίστα μείνει κενή τότε το δεξιό πλαίσιο ξανά γίνεται αόρατο. Στο δεξιό πλαίσιο επίσης εκτός από τη λίστα παραγγελιών υπάρχουν πάνω από αυτή δύο κουμπιά, το ένα έχει την ένδειξη «ΠΡΟΩΘΗΣΗ» και το άλλο την ένδειξη «ΑΚΥΡΩΣΗ».

Αν επιλεγεί το κουμπί «ΑΚΥΡΩΣΗ» τότε η λίστα διαγράφεται και το πλαίσιο γίνεται κρυφό, αν επιλεγεί το κουμπί «ΠΡΟΩΘΗΣΗ» το πλαίσιο γίνεται κρυφό και στο κεντρικό πλαίσιο εμφανίζεται η φόρμα καταχώρισης στοιχείων πελάτη. Η φόρμα στοιχείων πελάτη αποτελείται από οκτώ αντικείμενα εισαγωγής κειμένου και δύο κουμπιά.

Τα αντικείμενα κειμένου είναι τα εξής:

Όνομα

Επώνυμο

Διεύθυνση

Πόλη

Ταχ. Κώδικας

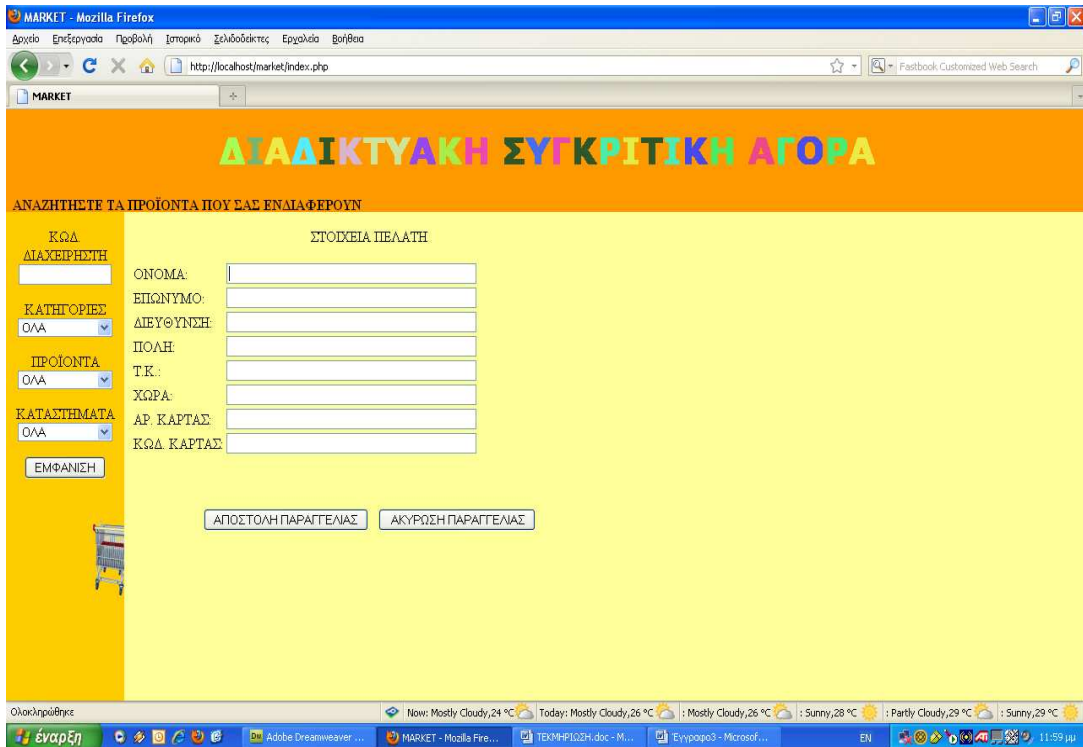
Χώρα

Αρ.κάρτας

Κωδικός κάρτας

Το ένα κουμπί έχει την ένδειξη «ΑΠΟΣΤΟΛΗ ΠΑΡΑΓΓΕΛΙΑΣ» και το άλλο «ΑΚΥΡΩΣΗ ΠΑΡΑΓΓΕΛΙΑΣ».

Αν επιλεγεί η αποστολή της παραγγελίας τότε καταχωρείται η παραγγελία και επιστρέφουμε στην αρχική σελίδα, ενώ με την ακύρωση διαγράφεται και επιστρέφουμε στην αρχική σελίδα.



ΙΣΤΟΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ

Η δεύτερη ιστοσελίδα απαρτίζεται από τρία πλαίσια.

Το πρώτο είναι οριζόντιο και καταλαμβάνει όλο το πλάτος της σελίδας, περιέχει τον τίτλο διαχείριση βάσης.

Το αριστερό πλαίσιο με τίτλο «ΠΙΝΑΚΕΣ ΒΑΣΗΣ» περιλαμβάνει μία λίστα επιλογών διαχείρισης των πινάκων της βάσης καθώς και την διαγραφή και την δημιουργία δεδομένων, που με την σειρά από πάνω προς τα κάτω είναι η εξής:

Καταστήματα

Κατηγορίες προϊόντων

Προϊόντα

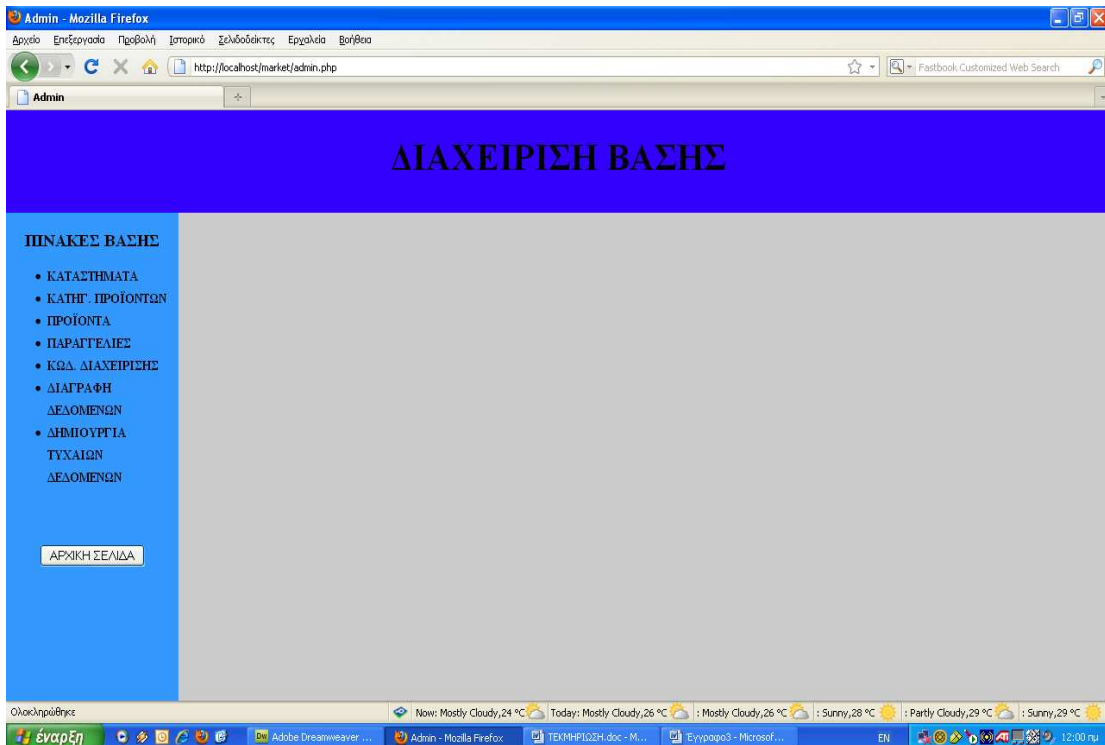
Παραγγελίες

Κωδικοί διαχείρισης

Διαγραφή δεδομένων

Δημιουργία τυχαίων δεδομένων.

Μετά τη λίστα υπάρχει και ένα κουμπί με την ένδειξη «ΑΡΧΙΚΗ ΣΕΛΙΔΑ» που αν επιλεγεί επιστρέφει ο χρήστης στην αρχική σελίδα.



ΦΟΡΜΕΣ ΠΙΝΑΚΩΝ ΒΑΣΗΣ

ΚΑΤΑΣΤΗΜΑΤΑ.

Η επιλογή αυτή είναι διαθέσιμη μόνο για τον διαχειριστή του ιστοτόπου.

Εμφανίζεται στο κεντρικό πλαίσιο ένας πίνακας με δύο πεδία που είναι τα εξής:

Κωδικός καταστήματος.

Επωνυμία καταστήματος.

Επίσης κάτω από τον πίνακα έχουμε τρία κουμπιά ορατά και ένα αόρατο κατά αρχάς.

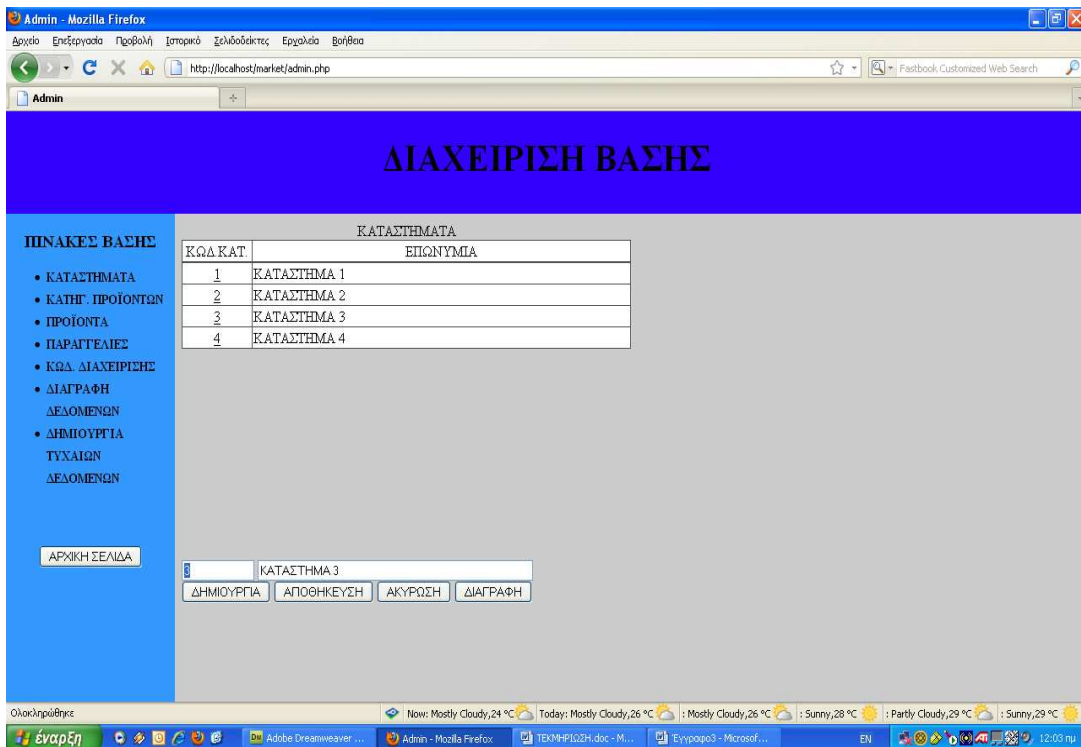
Οι ενδείξεις των κουμπιών από αριστερά προς τα δεξιά είναι οι εξής:

«ΔΗΜΙΟΥΡΓΙΑ»

«ΑΠΟΘΗΚΕΥΣΗ»

«ΑΚΥΡΩΣΗ»

«ΔΙΑΓΡΑΦΗ»



Το κουμπί «ΔΙΑΓΡΑΦΗ» εμφανίζεται όταν έχει επιλεγεί κάποια εγγραφή προς επεξεργασία.

Η επεξεργασία κάποιας εγγραφής γίνεται εάν επιλεγεί από τον πίνακα που εμφανίζεται κάποια εγγραφή πατώντας το πρώτο πεδίο της που είναι και υπογραμμισμένο.

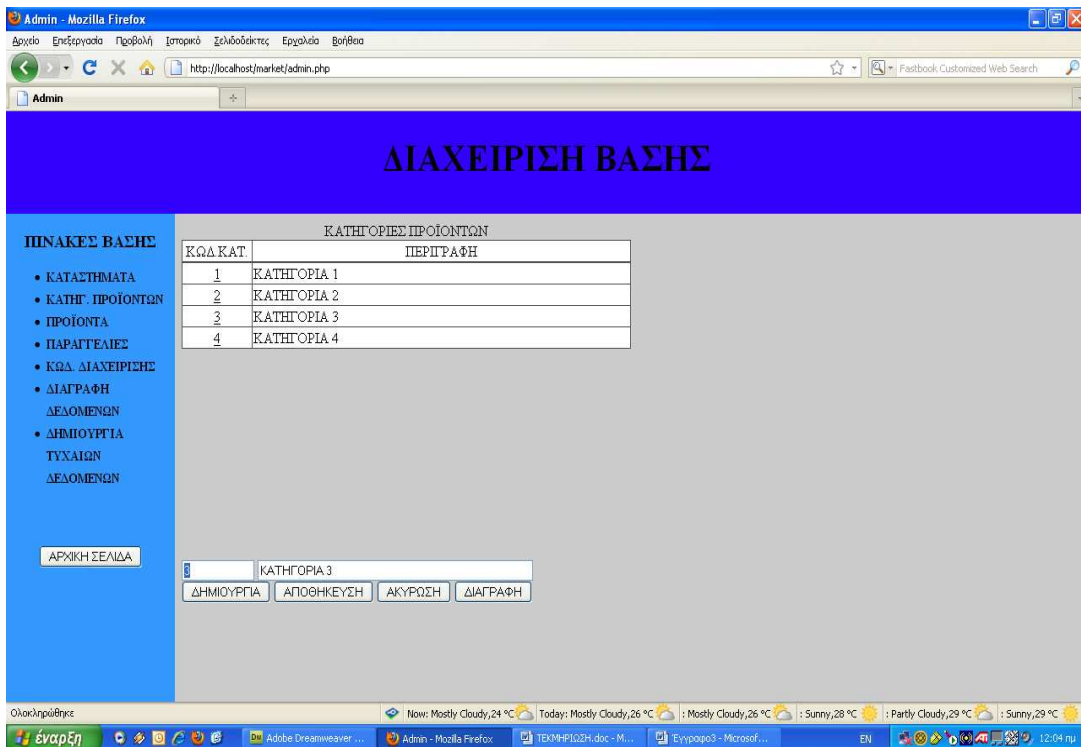
Όταν επιλεγεί προς επεξεργασία κάποια εγγραφή εμφανίζονται πάνω απ' τα κουμπιά δύο αντικείμενα κειμένου με προεπιλεγμένες τις τιμές της επιλεγμένης εγγραφής.

Τα αντικείμενα κειμένου αυτά εμφανίζονται επίσης και με την επιλογή του κουμπιού «ΔΗΜΙΟΥΡΓΙΑ».

Το κουμπί «ΑΚΥΡΩΣΗ» ακυρώνει την διαδικασία επεξεργασίας ή δημιουργίας νέας καταχώρισης, ενώ το κουμπί «ΑΠΟΘΗΚΕΥΣΗ» καταχωρεί την νέα εγγραφή ή τις όποιες αλλαγές έγιναν κατά την διαδικασία επεξεργασίας.

ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΪΟΝΤΩΝ

Η επιλογή αυτή είναι διαθέσιμη μόνο για τον διαχειριστή του ιστοτόπου. Ισχύουν τα ίδια που αναφέρθηκαν στην παράγραφο για τα καταστήματα.



ΠΡΟΪΟΝΤΑ

Με την επιλογή αυτή οι χρήστες έχουν δικαίωμα μόνο σε ότι αφορά τα δικά τους καταστήματα ενώ ο διαχειριστής σε όλα.

Εμφανίζεται στο κεντρικό πλαίσιο ένας πίνακας με πέντε πεδία που είναι τα εξής:

Κωδικός προϊόντος.

Περιγραφή.

Κατηγορία.

Κατάστημα.

Τιμή.

Επίσης κάτω από τον πίνακα έχουμε τρία κουμπιά ορατά και ένα αόρατο κατά αρχάς.

Οι ενδείξεις των κουμπιών από αριστερά προς τα δεξιά είναι οι εξής:

«ΔΗΜΙΟΥΡΓΙΑ»

«ΑΠΟΘΗΚΕΥΣΗ»

«ΑΚΥΡΩΣΗ»

«ΔΙΑΓΡΑΦΗ»

ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΗΣ

ΠΙΝΑΚΕΣ ΒΑΣΗΣ

- ΚΑΤΑΣΤΗΜΑΤΑ
- ΚΑΤΗΓ. ΠΡΟΪΟΝΤΩΝ
- ΠΡΟΪΟΝΤΑ
- ΠΑΡΑΓΓΕΛΙΕΣ
- ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ
- ΔΙΑΓΡΑΦΗ
- ΔΕΔΟΜΕΝΩΝ
- ΔΗΜΙΟΥΡΓΙΑ
- ΤΥΧΑΙΩΝ
- ΔΕΔΟΜΕΝΩΝ

ΠΡΟΪΟΝΤΑ				
ΚΩΔ.	ΠΕΡΙΓΡΑΦΗ	ΚΑΤΗΓΟΡΙΑ	ΚΑΤΑΣΤΗΜΑ	ΤΙΜΗ
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 1	ΚΑΤΑΣΤΗΜΑ 4	14
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 2	ΚΑΤΑΣΤΗΜΑ 1	23
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 2	ΚΑΤΑΣΤΗΜΑ 2	18
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 2	ΚΑΤΑΣΤΗΜΑ 3	22
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 2	ΚΑΤΑΣΤΗΜΑ 4	16
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 3	ΚΑΤΑΣΤΗΜΑ 1	13
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 3	ΚΑΤΑΣΤΗΜΑ 2	18
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 3	ΚΑΤΑΣΤΗΜΑ 3	23
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 3	ΚΑΤΑΣΤΗΜΑ 4	21
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 4	ΚΑΤΑΣΤΗΜΑ 1	14
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 4	ΚΑΤΑΣΤΗΜΑ 2	25
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 4	ΚΑΤΑΣΤΗΜΑ 3	15
9	ΠΡΟΪΟΝ 9	ΚΑΤΗΓΟΡΙΑ 4	ΚΑΤΑΣΤΗΜΑ 4	10

ΑΡΧΙΚΗ ΣΕΛΙΔΑ

ΠΡΟΪΟΝ 9 ΚΑΤΗΓΟΡΙΑ 4 ΚΑΤΑΣΤΗΜΑ 1 14

ΠΡΟΣΘΗΚΗ ΑΠΟΘΗΚΕΥΣΗ ΑΚΥΡΩΣΗ ΔΙΑΓΡΑΦΗ

Το κουμπί «ΔΙΑΓΡΑΦΗ» εμφανίζεται όταν έχει επιλεγεί κάποια εγγραφή προς επεξεργασία.

Η επεξεργασία κάποιας εγγραφής γίνεται εάν επιλεγεί από τον πίνακα που εμφανίζεται κάποια εγγραφή πατώντας το πρώτο πεδίο της που είναι και υπογραμμισμένο.

Όταν επιλεγεί προς επεξεργασία κάποια εγγραφή εμφανίζονται πάνω απ' τα κουμπιά τρία αντικείμενα κειμένου και δύο πλαίσια πτυσσόμενης λίστας με προεπιλεγμένες τις τιμές της επιλεγμένης εγγραφής.

Τα αντικείμενα κειμένου και πτυσσόμενης λίστας εμφανίζονται επίσης και με την επιλογή του κουμπιού «ΔΗΜΙΟΥΡΓΙΑ».

Το κουμπί «ΑΚΥΡΩΣΗ» ακυρώνει την διαδικασία επεξεργασίας ή δημιουργίας νέας καταχώρισης, ενώ το κουμπί «ΑΠΟΘΗΚΕΥΣΗ» καταχωρεί την νέα εγγραφή ή τις όποιες αλλαγές έγιναν κατά την διαδικασία επεξεργασίας.

ΠΑΡΑΓΓΕΛΙΕΣ

Με την επιλογή αυτή παρακολουθούμε τις παραγγελίες που έχουν καταχωρηθεί. Κάθε υπεύθυνος καταστήματος μπορεί να δει μόνο τις παραγγελίες που περιέχουν δικά του προϊόντα, ενώ ο διαχειριστής μπορεί να τις δει όλες.

Στο κεντρικό πλαίσιο εμφανίζονται δύο πίνακες.

Ο πρώτος εμφανίζει τα στοιχεία του πελάτη και ο δεύτερος τα προϊόντα που έχουν επιλεγεί.

Κάτω από τον πίνακα αυτό υπάρχουν δύο κουμπιά και ένα αντικείμενο κειμένου που είναι μόνο για ανάγνωση.

Το ένα κουμπί έχει την ένδειξη «ΠΡΟΗΓΟΥΜΕΝΗ» και το άλλο «ΕΠΟΜΕΝΗ» και χρησιμοποιούνται για την μετακίνηση από την μία παραγγελία στην άλλη μπρος και πίσω.

Το αντικείμενο κειμένου απεικονίζει τον συνολικό αριθμό παραγγελιών και τον αριθμό της τρέχουσας παραγγελίας.

Πάνω δεξιά υπάρχουν επίσης δύο κουμπιά με ενδείξεις «ΔΙΑΓΡΑΦΗ» και «ΕΚΤΕΛΕΣΗ».

Με την επιλογή «ΔΙΑΓΡΑΦΗ» διαγράφεται η παραγγελία ενώ με την «ΕΚΤΕΛΕΣΗ» αποστέλλονται υποθετικά τα προϊόντα και διαγράφονται.

ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΗΣ

ΠΙΝΑΚΕΣ ΒΑΣΗΣ

- ΚΑΤΑΣΤΗΜΑΤΑ
- ΚΑΤΗΓ. ΠΡΟΪΟΝΤΩΝ
- ΠΡΟΪΟΝΤΑ
- ΠΑΡΑΓΓΕΛΙΕΣ
- ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ
- ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ
- ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ

ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΗ

ΟΝΟΜΑ:	ΟΝΟΜΑ 7	ΕΠΩΝΥΜΟ:	ΕΠΩΝΥΜΟ 7
ΔΙΕΥΘΥΝΣΗ:	ΔΙΕΥΘΥΝΣΗ 7	ΠΟΛΗ:	ΠΟΛΗ 7
Τ.Κ.:	7	ΧΩΡΑ:	ΧΩΡΑ 7
ΑΡ.ΚΑΡΤΑΣ:	7	ΚΩΔ.ΚΑΡΤΑΣ:	7

ΣΤΟΙΧΕΙΑ ΠΑΡΑΓΓΕΛΙΑΣ

ΚΑΤΑΣΤΗΜΑ	ΚΑΤΗΓΟΡΙΑ	ΠΡΟΪΟΝ	ΤΕΜ.	ΤΙΜΗ	ΑΞΙΑ
ΚΑΤΑΣΤΗΜΑ 1	ΚΑΤΗΓΟΡΙΑ 4	ΠΡΟΪΟΝ 1	3	28	84
ΚΑΤΑΣΤΗΜΑ 4	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 1	1	19	19
ΚΑΤΑΣΤΗΜΑ 4	ΚΑΤΗΓΟΡΙΑ 4	ΠΡΟΪΟΝ 1	1	20	20
ΚΑΤΑΣΤΗΜΑ 1	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 2	2	27	54
ΚΑΤΑΣΤΗΜΑ 2	ΚΑΤΗΓΟΡΙΑ 1	ΠΡΟΪΟΝ 2	2	25	50
ΣΥΝΟΛΙΚΗ ΑΞΙΑ					227

ΑΡΧΙΚΗ ΣΕΛΙΔΑ

ΠΡΟΗΓΟΥΜΕΝΗ 1 ΑΠΟ 8 ΕΠΟΜΕΝΗ

ΚΩΔΙΚΟΙ ΔΙΑΧΕΙΡΙΣΗΣ

Οι χρήστες βλέπουν και μπορούν μόνο να αλλάξουν τους δικούς τους κωδικούς αντίθετα ο διαχειριστής έχει πλήρη πρόσβαση.

Εμφανίζεται ένας πίνακας με τρία πεδία που είναι τα εξής:

Κωδικός

Ονοματεπώνυμο

Κατάστημα

Κάτω από τον πίνακα αυτό υπάρχει αρχικά ένα κουμπί με ένδειξη «ΔΗΜΙΟΥΡΓΙΑ» ενώ υπάρχουν και τρία κουμπιά που είναι αρχικά κρυφά με ενδείξεις «ΑΠΟΘΗΚΕΥΣΗ», «ΑΚΥΡΩΣΗ», «ΔΙΑΓΡΑΦΗ» και τα οποία εμφανίζονται αν επιλεγεί η δημιουργία νέας εγγραφής ή επεξεργασία υπάρχουσας.

Η διαδικασία επεξεργασίας γίνεται όπως και στους προηγούμενους πίνακες δηλαδή με την επιλογή υπογραμμισμένου πεδίου κάποιας εγγραφής.

Ομοίως η διαδικασία αποθήκευσης, ακύρωσης και διαγραφής είναι ίδιες με τους προηγούμενους πίνακες.

ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΗΣ

ΚΩΔΙΚΟΣ ΔΙΑΧΕΙΡΙΣΗΣ

ΚΩΔΙΚΟΣ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΚΑΤΜΑ
0	ΔΙΑΧΕΙΡΗΣΤΗΣ	0
1	ΧΡΗΣΤΗΣ 1	ΚΑΤΑΣΤΗΜΑ 1
2	ΧΡΗΣΤΗΣ 2	ΚΑΤΑΣΤΗΜΑ 2
3	ΧΡΗΣΤΗΣ 3	ΚΑΤΑΣΤΗΜΑ 3
4	ΧΡΗΣΤΗΣ 4	ΚΑΤΑΣΤΗΜΑ 4

ΠΙΝΑΚΕΣ ΒΑΣΗΣ

- ΚΑΤΑΣΤΗΜΑΤΑ
- ΚΑΤΗΓ. ΠΡΟΪΟΝΤΩΝ
- ΠΡΟΪΟΝΤΑ
- ΠΑΡΑΓΓΕΛΙΕΣ
- ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ
- ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ
- ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ

ΑΡΧΙΚΗ ΣΕΛΙΔΑ

ΧΡΗΣΤΗΣ 2 ΚΑΤΑΣΤΗΜΑ 2

ΔΗΜΙΟΥΡΓΙΑ ΑΠΟΘΗΚΕΥΣΗ ΑΚΥΡΩΣΗ ΔΙΑΓΡΑΦΗ

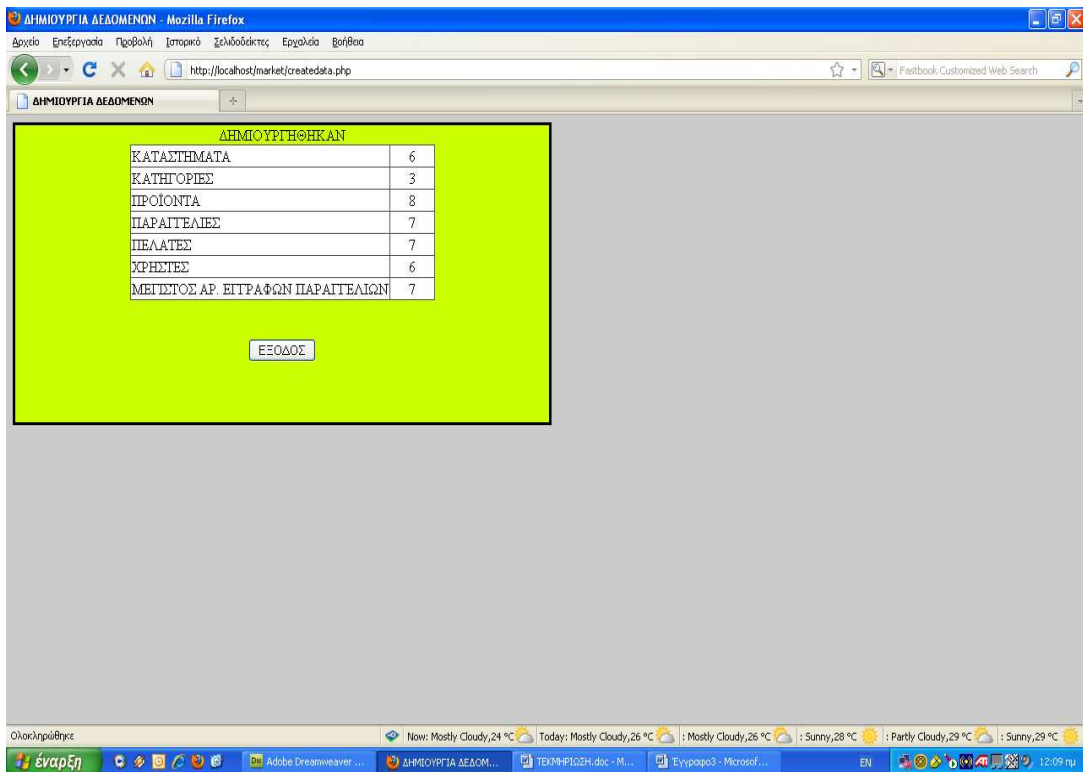
ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ

Με την επιλογή αυτή διαγράφονται όλα τα δεδομένα της βάσης κατόπιν επιβεβαίωσης.

ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ

Με την επιλογή αυτή δημιουργούνται κάθε φορά τυχαίες σε αριθμό εγγραφές σε όλους τους πίνακες της βάσης για λόγους επίδειξης.

Μετά την δημιουργία των εγγραφών αυτών εμφανίζεται ιστοσελίδα που περιέχει πίνακα ο οποίος αναφέρει τον αριθμό των εγγραφών ανά πίνακα που δημιουργήθηκαν και από κάτω ένα κουμπί με ένδειξη «ΕΞΟΔΟΣ» το οποίο μας επαναφέρει στην αρχική σελίδα.



ΑΝΑΛΥΣΗ ΒΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Η εφαρμογή χρησιμοποιεί για την αποθήκευση και την ανάκτηση δεδομένων την σχεσιακή βάση δεδομένων **MySQL v5.0.67**.

Το όνομα της βάσης είναι «stores» και ο λογαριασμός διαχείρισης της έχει οριστεί ως εξής:

Username: admin

Password: 1234

ΠΙΝΑΚΕΣ ΒΑΣΗΣ

Η βάση απαρτίζεται από επτά πίνακες.

Οι πίνακες αυτοί ονομαστικά είναι οι εξής:

categories

customers

orders

passlist

products

sessions

stores

ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΠΙΝΑΚΩΝ

categories

ο πίνακας categories (κατηγορίες) αποτελείται από τα εξής πεδία:

CategoryId	τύπου	UNSIGNED INTEGER
Name	»»»	CHAR(30)

Στο πρώτο πεδίο καταχωρείτε ένας αριθμητικός κωδικός μοναδικός και χαρακτηριστικός για κάθε κατηγορία.

Στο δεύτερο πεδίο καταχωρείτε η περιγραφή της κατηγορίας.

Το πεδίο **CategoryId** είναι και το **PRIMARY KEY** (πρωτεύον κλειδί) για τον πίνακα αυτόν.

customers

ο πίνακας customers (πελάτες) αποτελείται από τα εξής πεδία:

OrderSid	τύπου	UNSIGNED INTEGER
FirstName	»»»	CHAR(20)
SurName	»»»	CHAR(20)
Address	»»»	CHAR(45)
City	»»»	CHAR(25)
ZipCode	»»»	UNSIGNED INTEGER
Country	»»»	CHAR(25)
CardNo	»»»	UNSIGNED INTEGER
CardCode	»»»	UNSIGNED INTEGER
Stamp	»»»	UNSIGNED INTEGER

Στο πρώτο πεδίο καταχωρείτε ένας αριθμητικός κωδικός μοναδικός και χαρακτηριστικός για κάθε παραγγελία πελάτη.

Το πεδίο αυτό χρησιμοποιείτε για συσχέτιση της εγγραφής κάποιου πελάτη με τις αντίστοιχες εγγραφές στο αρχείο παραγγελιών.

Η σχέση συσχέτισης είναι τύπου ONE TO MANY δηλαδή για μία μοναδική εγγραφή στο αρχείο πελατών μπορούν να αντιστοιχούν μία ή περισσότερες εγγραφές στο αρχείο παραγγελιών αλλά οι τιμές των πεδίων OrderSid να είναι ίδιες με αυτές του πεδίου OrderSid του αρχείου customers.

Στο δεύτερο πεδίο καταχωρείτε το όνομα του πελάτη.
Στο τρίτο πεδίο καταχωρείτε το επώνυμο του πελάτη.
Στο τέταρτο πεδίο καταχωρείτε η διεύθυνση του πελάτη.
Στο πέμπτο πεδίο καταχωρείτε η πόλη του πελάτη.
Στο έκτο πεδίο καταχωρείτε ο Τ.Κ. του πελάτη.
Στο έβδομο πεδίο καταχωρείτε η χώρα του πελάτη.
Στο όγδοο πεδίο καταχωρείτε ο αριθμός κάρτας του πελάτη.
Στο ένατο πεδίο καταχωρείτε το PIN της κάρτας του πελάτη.
Στο δέκατο πεδίο καταχωρείτε η ημερομηνία και ώρα της καταχώρισης με την μορφή **TIMESTAMP** της γλώσσας προγραμματισμού PHP.
Το πεδίο **Stamp** είναι επίσης και **INDEX** (δείκτης ευρετηρίου) για τον πίνακα αυτόν.
Το πεδίο **OrderSid** είναι και το **PRIMARY KEY** (πρωτεύον κλειδί) για τον πίνακα αυτόν.

orders

ο πίνακας orders (παραγγελίες) αποτελείτε από τα εξής πεδία:

OrderSid	τύπου UNSIGNED INTEGER
ProductId	τύπου UNSIGNED INTEGER
StoreId	τύπου UNSIGNED INTEGER
CategoryId	τύπου UNSIGNED INTEGER
Price	τύπου FLOAT
Quantity	τύπου UNSIGNED INTEGER
Stamp	τύπου UNSIGNED INTEGER

Στο πρώτο πεδίο καταχωρείτε ένας αριθμητικός κωδικός μοναδικός και χαρακτηριστικός για κάθε ομάδα εγγραφών του αρχείου παραγγελιών.
Το πεδίο αυτό χρησιμοποιείτε για συσχέτιση της εγγραφής κάποιου πελάτη με τις αντίστοιχες εγγραφές στο αρχείο orders (παραγγελιών).
Η σχέση συσχέτισης έχει περιγραφή στην περιγραφή του πίνακα customers (πελάτες).
Στο δεύτερο πεδίο καταχωρείτε ο κωδικός του προϊόντος.
Στο τρίτο πεδίο καταχωρείτε ο κωδικός του καταστήματος.
Στο τέταρτο πεδίο καταχωρείτε ο κωδικός της κατηγορίας του προϊόντος.

Στο πέμπτο πεδίο καταχωρείτε η τιμή του προϊόντος.

Στο έκτο πεδίο καταχωρείτε η ποσότητα παραγγελίας του προϊόντος.

Στο έβδομο πεδίο καταχωρείτε η ημερομηνία και ώρα της καταχώρισης με την μορφή **TIMESTAMP** της γλώσσας προγραμματισμού PHP.

Τα πεδία OrderSid, ProductId, StoreId, CategoryId μαζί απαρτίζουν το **PRIMARY KEY** (πρωτεύον κλειδί) του πίνακα.

Τα πεδία OrderSid, ProductId, Stamp είναι επίσης και **INDEXES** (δείκτες ευρετηρίου) του πίνακα.

passlist

ο πίνακας passlist (λίστα κωδικών) αποτελείτε από τα εξής πεδία:

Pass	τύπου CHAR(20)
Name	τύπου CHAR(30)
StoreId	τύπου UNSIGNED INTEGER

Στο πρώτο πεδίο καταχωρείτε ένας αλφαριθμητικός κωδικός χρήστη.

Στο δεύτερο πεδίο καταχωρείτε η επωνυμία του χρήστη.

Στο τρίτο πεδίο καταχωρείτε ο κωδικός του καταστήματος τον οποίο αφορά ο κωδικός χρήστη.

Ο κωδικός του διαχειριστή πρέπει να έχει κωδικό καταστήματος 0.

Τα πεδία Pass, StoreId αποτελούν μαζί το **PRIMARY KEY** (πρωτεύον κλειδί) του πίνακα.

products

ο πίνακας products (προϊόντα) αποτελείτε από τα εξής πεδία:

ProductId	τύπου UNSIGNED INTEGER
Product	τύπου CHAR(30)
CategoryId	τύπου UNSIGNED INTEGER
StoreId	τύπου UNSIGNED INTEGER
Price	τύπου FLOAT

Ο πίνακας αυτός είναι ένας πίνακας κοινής αποθήκης για όλα τα καταστήματα.

Τα προϊόντα καταχωρούνται ανά κατηγορία και ανά κατάστημα.

Για αυτό και δεν μπορούν να υπάρξουν περισσότερες από μία εγγραφές με κοινά τα πεδία ProductId, CategoryId, StoreId τα οποία για αυτόν το λόγο αποτελούν και το PRIMARY KEY (πρωτεύον κλειδί) του πίνακα.

Στο πρώτο πεδίο καταχωρείτε ο αριθμητικός κωδικός του προϊόντος.

Στο δεύτερο πεδίο καταχωρείτε περιγραφή του προϊόντος.

Στο τρίτο πεδίο καταχωρείτε ο αριθμητικός κωδικός της κατηγορίας του προϊόντος.

Στο τέταρτο πεδίο καταχωρείτε ο αριθμητικός κωδικός του καταστήματος του προϊόντος.

Στο πέμπτο πεδίο καταχωρείτε η τιμή του προϊόντος.

Τα πεδία CategoryId, StoreId είναι επίσης και INDEXES (δείκτες ευρετηρίου) του πίνακα.

sessions

ο πίνακας sessions αποτελείται από τα εξής πεδία:

sessionid τύπου UNSIGNED INTEGER

stamp τύπου UNSIGNED INTEGER

Ο πίνακας αυτός αποθηκεύει τυχαίους κωδικούς αριθμούς μεγάλου μήκους που δημιουργούνται όταν ένας επιτυχής κωδικός εισόδου εισαχθεί και ο οποίος αποθηκεύεται μέσω Cookie και στον υπολογιστή του χρήστη.

Τα Cookies και οι αντίστοιχες εγγραφές στον πίνακα διαγράφονται όταν ο χρήστης βγει από την σελίδα διαχείρισης ή μετά το πέρας 1800 δευτερολέπτων (μισή ώρα) για λόγους ασφαλείας.

Αν κάποιος χρήστης παραμείνει στη σελίδα διαχείρισης περισσότερο από μισή ώρα θα χρειαστεί να επιστρέψει στην αρχική σελίδα και να ξανά εισάγει τον κωδικό του.

Οι κωδικοί αυτοί χρησιμεύουν για να επαληθεύουν την ταυτότητα του χρήστη.

Στο πρώτο πεδίο καταχωρείτε ο αριθμητικός κωδικός της συνεδρίας (session).

Στο δεύτερο πεδίο καταχωρείτε η ημερομηνία και ώρα της δημιουργίας της συνεδρίας με την μορφή **TIMESTAMP** της γλώσσας προγραμματισμού PHP.

Το πεδίο **sessionid** είναι και το **PRIMARY KEY** (πρωτεύον κλειδί) για τον πίνακα αυτόν.

stores

ο πίνακας stores (καταστήματα) αποτελείται από τα εξής πεδία:

Storeid τύπου UNSIGNED INTEGER

Firm τύπου CHAR(45)

Στο πρώτο πεδίο καταχωρείτε ο αριθμητικός κωδικός του καταστήματος.

Στο δεύτερο πεδίο καταχωρείτε η επωνυμία του καταστήματος.

Το πεδίο **StoreId** είναι και το **PRIMARY KEY** (πρωτεύον κλειδί) για τον πίνακα αυτόν.

ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Κάθε πίνακας έχει τουλάχιστον ένα πρωτεύον κλειδί. (**PRIMARY KEY**) και από κανένα έως όσα πεδία έχει δευτερεύοντα κλειδιά (**INDEXES**).

Τα κλειδιά αυτά μπορεί να αποτελούνται από ένα πεδίο ή και περισσότερα.

Ο λόγος δημιουργίας τους έχει σκοπό, πρώτον να διασφαλίσει ότι δεν μπορούν να καταχωρηθούν διπλές εγγραφές (**PRIMARY KEYS**) και δεύτερον να δίνουν την δυνατότητα συσχετισμών μεταξύ των πινάκων κατά την εκτέλεση ερωτημάτων **SQL** και η επεξεργασία των δεδομένων από τη βάση να γίνεται γρηγορότερα χρησιμοποιώντας ένα πεδίο **INDEX** για εύρεση ή σύγκριση και όχι όλα τα πεδία της εγγραφής.

Η σχέσεις συσχέτισης μπορεί να είναι:

PRIMARY KEY -> PRIMARY KEY

*Με αυτή την σχέση επεξεργάζονται δεδομένα μεταξύ δύο ή περισσότερων πινάκων που τα πεδία **PRIMARY KEY** όλων είναι ίσα.*

*Η σχέση αυτή αποκαλείτε **ONE TO ONE** (ΜΙΑ ΠΡΟΣ ΜΙΑ)*

*Επειδή τα πεδία **PRIMARY KEY** είναι μοναδικά πάντα το αποτέλεσμα είναι μια κοινή εγγραφή από όλους τους εμπλεκόμενους πίνακες.*

PRIMARY KEY -> INDEX

*Με αυτή την σχέση επεξεργάζονται δεδομένα μεταξύ δύο ή περισσότερων πινάκων που τα πεδία **PRIMARY KEY** του ενός είναι ίσο με τα πεδία **INDEX** των άλλων.*

*Η σχέση αυτή αποκαλείτε **ONE TO MANY** (ΜΙΑ ΠΡΟΣ ΠΟΛΛΕΣ)*

Το αποτέλεσμα αυτής της σχέσης είναι για κάθε πεδίο **PRIMARY KEY** της κάθε εγγραφής του πρώτου πίνακα να επεξεργάζονται δεδομένα εγγραφών στους άλλους πίνακες που τα πεδία **INDEX** έχουν την ίδια τιμή με το **PRIMARY KEY**.

Επειδή τα πεδία **INDEX** δεν είναι απαραίτητος μοναδικά στον πίνακα τους το αποτέλεσμα είναι για κάθε **PRIMARY KEY** να επεξεργάζονται πολλές αντίστοιχες εγγραφές στους άλλους πίνακες που τα πεδία **INDEX** τους είναι ίσα με το πεδίο **PRIMARY KEY** του πρώτου πίνακα.

Για παράδειγμα αν σε έναν πίνακα **A** κάθε εγγραφή έχει τα στοιχεία ενός διαφορετικού πελάτη με **PRIMARY KEY** το ΑΦΜ του και σε έναν άλλο πίνακα **B** έχουμε σε κάθε εγγραφή τα προϊόντα που αγόρασε κάποιος πελάτης με πεδίο **INDEX** το ΑΦΜ του, τότε για κάθε πελάτη του πίνακα **A** με ΑΦΜ **a**, επεξεργάζονται όλες η εγγραφές του πίνακα **B** που το ΑΦΜ τους είναι **a**.

INDEX -> INDEX

Με αυτή την σχέση επεξεργάζονται δεδομένα μεταξύ δύο ή περισσότερων πινάκων που τα πεδία **INDEX** τους είναι ίσα.

Το αποτέλεσμα είναι πολλές εγγραφές από όλους τους πίνακες που τα πεδία **INDEX** τους είναι ίσα.

Στην πράξη δεν χρησιμοποιείτε συχνά αυτή η σχέση.

Οι συσχετισμοί των πινάκων βοηθούν τη βάση δεδομένων να διασφαλίσει την αξιοπιστία και την ακεραιότητα των δεδομένων καθώς και την αποφυγή πολλαπλών καταχωρίσεων και ανακτήσεων ομοίων δεδομένων σε διαφορετικούς ή από διαφορετικούς πίνακες αντίστοιχα, που θα καταλάμβαναν περισσότερο χώρο στον δίσκο και την μνήμη και θα μεγάλωναν αρκετά τον χρόνο καταχώρισης και προσπέλασης των δεδομένων στην βάση.

Για παράδειγμα, δεν μπορεί να καταχωρηθεί προϊόν σε κάποια κατηγορία που δεν υπάρχει, όπως και δεν μπορεί να καταχωρηθεί προϊόν σε κατάσταση που δεν υπάρχει.

ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΕΡΙΓΡΑΦΗ ΚΟΙΝΟΤΥΠΩΝ ΤΜΗΜΑΤΩΝ ΤΟΥ ΚΩΔΙΚΑ

Η εφαρμογή εκτός της βάσης MySQL χρησιμοποιεί την γλώσσα PHP που τρέχει στον SERVER και παρέχει την επικοινωνία του χρήστη με τη βάση.

Για λόγους αποφυγής επαναλήψεων στην επεξήγηση του κώδικα της PHP στην παρούσα εφαρμογή παραθέτεται η επεξήγηση ενός τμήματος κώδικα που είναι κοινός προγραμματιστικά, σε όλα τα αρχεία PHP τα οποία επικοινωνούν με την βάση MySQL στην παρούσα εφαρμογή.

Παρακάτω βρίσκεται το βασικό κομμάτι κώδικα του αρχείου **products.php**, το οποίο ανακτά όλα τα καταχωρημένα προϊόντα που βρίσκονται στον πίνακα **products** της βάσης MySQL.

```
<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "SELECT p.`ProductId`, p.`Product`, c.`Name`, s.`Firm`, p.`Price`,
p.`CategoryId`, p.`StoreId` FROM products p, categories c, stores s WHERE
p.CategoryId=c.CategoryId AND p.StoreId=s.StoreId ORDER BY p.`ProductId`,
p.`CategoryId`, p.`StoreId`";

$result = mysqli_query($Connection,$query);
echo "<table border='1'>";
echo "<caption>ΠΡΟΪΟΝΤΑ</caption>";
echo "<tr>";
echo "<td style='width:80px;text-align:center'>" . "ΚΩΔ." . "</td>";
echo "<td style='width:300px;text-align:center'>" . "ΠΕΡΙΓΡΑΦΗ" . "</td>";
echo "<td style='width:200px;text-align:center'>" . "ΚΑΤΗΓΟΡΙΑ" . "</td>";
echo "<td style='width:200px;text-align:center'>" . "ΚΑΤΑΣΤΗΜΑ" . "</td>";
```

```

echo "<td style='width:80px;text-align:center>" . "TIMH" . "</td>";
echo "</tr>";
echo "</table>";
if($result && mysqli_num_rows($result))
{
    echo "<table border='1'>";
    for($i=0;$i<mysqli_num_rows($result);$i++)
    {
        echo "<tr>";
        $arr = mysqli_fetch_row($result);
        for($k=0;$k<mysqli_num_fields($result)-2;$k++)
        {
            if($k==0)
                echo "<td id='" . $arr[$k] . "' onclick='EditProd(" .
                $arr[$k] . "," . $arr[mysqli_num_fields($result)-1] . "," .
                $arr[mysqli_num_fields($result)-2] . ") 'style='width:80px;text-
                align:center;cursor:pointer;text-decoration:underline'>";
            else if($k==1)
                echo "<td style='width:300px' id='f' . $arr[0] . "'>";
            else if($k==2)
                echo "<td style='width:200px;text-align:center' id='c' .
                $arr[0] . "'>";
            else if($k==3)
                echo "<td style='width:200px;text-align:center' id='s' .
                $arr[0] . "'>";
            else if($k==4)
                echo "<td style='width:80px;text-align:center' id='p' .
                $arr[0] . "'>";
            echo $arr[$k];
            echo "</td>";
        }
        echo "</tr>";
    }
}

```



```
echo "</table>";  
echo"<script>document.getElementById('scroll').scrollTop=100900</script>";  
}  
mysqli_free_result($result);  
mysqli_close($Connection);  
?>
```

Όλες η εντολές της **PHP** που επικοινωνούν με την βάση επιστρέφουν κάποια τιμή που μπορεί να είναι δεδομένα ή αν εκτελέσθηκε ή όχι επιτυχώς η εντολή (**true/false**). Για λόγους λοιπόν ελέγχου θέτουμε πάντα μια μεταβλητή να παίρνει την επιστρεφόμενη αυτή τιμή.

Η μεταβλητή **\$Connection** παίρνει την τιμή που επιστρέφει η εντολή [mysqli_connect](#). Αν η τιμή είναι **false** τότε η σύνδεση με τη βάση απέτυχε, αλλιώς η τιμή είναι ένας μοναδικός αριθμός ταυτότητας της σύνδεσης.

Η εντολή [mysqli_connect](#) παίρνει τέσσερις παραμέτρους που είναι οι εξής:

Όνομα SERVER (\$host) που είναι εγκατεστημένη η βάση.

Όνομα χρήστη (\$user) που θέλει να συνδεθεί με τη βάση.

Ο κωδικός (\$password) του χρήστη.

Το όνομα της βάσης (“stores” στην παρούσα εφαρμογή) που θέλει να συνδεθεί.

Η μεταβλητή [\\$query](#) παίρνει σαν αλφαριθμητική τιμή, το ερώτημα σε μορφή εντολών **SQL**.

Η μεταβλητή [\\$result](#) παίρνει την τιμή που επιστρέφει η εντολή [mysqli_query](#) που είναι υπεύθυνη για την εκτέλεση του ερωτήματος.

Αν η τιμή είναι **false** τότε είτε δεν υπάρχουν εγγραφές προς επιστροφή ή για κάποιο λόγο η εντολή απέτυχε.

Η εντολή [mysqli_query](#) παίρνει δύο παραμέτρους.

Η πρώτη είναι ο χαρακτηριστικός αριθμός της σύνδεσης (**\$Connection**) που έχει δημιουργηθεί και η άλλη η μεταβλητή που περιέχει τις εντολές **SQL** προς εκτέλεση ([\\$query](#)).

Αν υπάρχουν εγγραφές που επιστρέφονται μετά από εκτέλεση ερωτήματος **SQL**, αυτά βρίσκονται στην μεταβλητή [\\$result](#) με την μορφή αντικειμένου πίνακα.

Η ανάγνωση των εγγραφών του αποτελέσματος γίνεται με την βοήθεια της εντολής

[mysqli_fetch_row\(\\$result\)](#) της οποίας το αποτέλεσμα είναι ένας πίνακας τα στοιχεία του οποίου είναι οι τιμές των πεδίων της τρέχουσας εγγραφής.

Η εντολή ξεκινά δίνοντας τα πεδία της πρώτης εγγραφής και κατόπιν αυξάνει το δείκτη της θέσης της στον πίνακα αποτελεσμάτων, αυτό σημαίνει ότι όταν ξανά εκτελεστεί θα δώσει τα αποτελέσματα της δεύτερης εγγραφής κοκ, μέχρι να φτάσει στην τελευταία.

Για τον λόγο αυτό αν θέλουμε να διαβάσουμε όλες τις εγγραφές χρησιμοποιούμε ένα βρόγχο **for{}**, έχοντας αρχική τιμή **0** (πρώτη εγγραφή) και τελευταία τον συνολικό αριθμό εγγραφών του αποτελέσματος **μείων 1**.

Η εντολή [mysqli_num_rows\(\\$result\)](#) επιστρέφει το πλήθος των εγγραφών του αποτελέσματος. Επειδή οι πίνακες ξεκινούν την αρίθμηση των θέσεων τους από τον αριθμό **0** και όχι από το **1** η τελευταία θέση έχει τον αριθμό **(πλήθος εγγραφών - 1)**.

Αν ο πίνακας αποτελεσμάτων περιέχει περισσότερες από μία εγγραφές τότε χρειαζόμαστε δύο βρόγχους **for{}**, ο ένας μέσα στον άλλον, για να διαβάσουμε τα αποτελέσματα επειδή ο πίνακας είναι μονοδιάστατος.

Έστω η κάθε εγγραφή περιέχει τρία πεδία και το σύνολο των εγγραφών είναι δύο, τότε αν ο πίνακας αποτελεσμάτων είναι **A** έχουμε:

A[0] το πρώτο πεδίο της πρώτης εγγραφής, **A[2]** το τελευταίο πεδίο της πρώτης εγγραφής και **A[3]** το πρώτο πεδίο της δεύτερης εγγραφής κοκ.

Ο δεύτερος βρόγχος μας δίνει τα πεδία (στήλες) κάθε εγγραφής και το πλήθος των πεδίων (στηλών) το βρίσκουμε με την εντολή [mysqli_num_fields\(\\$result\)](#).

Τα αποτελέσματα που επιστρέφονται στον χρήστη εξάγονται με την εντολή **echo**.

Τα αποτελέσματα μπορεί να είναι είτε κείμενο είτε κώδικας **HTML**.

Για παράδειγμα, οι παρακάτω εντολές εξάγουν μία σελίδα **HTML** που περιέχει έναν πίνακα δύο εγγραφών και τριών πεδίων η κάθε μία.

```
echo"<HTML>";
echo"<BODY>";
echo"<TABLE>";
echo"<TR>";
echo"<TD>ΚΑΤΑΣΤΗΜΑΤΑ</TD><TD>ΠΡΟΪΟΝΤΑ</TD><TD>ΤΙΜΕΣ</TD>";
echo"</TR>";
echo"<TR>";
```

```

echo"<TD>ΚΑΤΑΣΤΗΜΑ1</TD><TD>ΠΡΟΪΟΝ1</TD><TD>10</TD>";
echo"</TR>";
echo"<TR>";
echo"<TD>ΚΑΤΑΣΤΗΜΑ2</TD><TD>ΠΡΟΪΟΝ2</TD><TD>11</TD>";
echo"</TR>";
echo"</TABLE>";
echo"</BODY>";
echo"</HTML>";

```

Το αποτέλεσμα είναι:

ΚΑΤΑΣΤΗΜΑΤΑ	ΠΡΟΪΟΝΤΑ	ΤΙΜΕΣ
ΚΑΤΑΣΤΗΜΑ1	ΠΡΟΪΟΝ1	10
ΚΑΤΑΣΤΗΜΑ2	ΠΡΟΪΟΝ2	11

Μετά το τέλος της εξαγωγής των δεδομένων πρέπει να ελευθερωθεί η μνήμη που καταλαμβάνει ο πίνακας αποτελεσμάτων και να κλείσει η σύνδεση με την βάση.

Το πρώτο επιτυγχάνεται με την εντολή [mysql_free_result\(\\$result\)](#) και το δεύτερο με την εντολή [mysql_close\(\\$Connection\)](#).

Αν δεν ελευθερωθεί η μνήμη που καταλαμβάνουν τα παραπάνω σύντομα θα καταρρεύσει ο **SERVER** από έλλειψη μνήμης.

Η εντολή [mysql_free_result\(\\$result\)](#) πρέπει να εκτελείτε μόνο όταν τα ερωτήματα **SQL** επιστρέφουν δεδομένα, δηλαδή ξεκινούν με την εντολή **SELECT**, αλλιώς παράγεται από την **PHP** προειδοποίηση (κάτι σαν λάθος αλλά δεν διακόπτεται η εκτέλεση του επόμενου κώδικα).

ΤΥΠΟΙ ΕΡΩΤΗΜΑΤΩΝ SQL

Οι βασικοί και συνηθέστεροι τύποι ερωτημάτων **SQL** είναι:

Ανάκτηση δεδομένων	(SELECT)
Εισαγωγή δεδομένων	(INSERT)
Τροποποίηση δεδομένων	(UPDATE)
Διαγραφή δεδομένων	(DELETE).

SELECT

Στο παραπάνω παράδειγμα κώδικα, ο τύπος ερωτήματος προς την βάση είχε σκοπό την ανάκτηση εγγραφών.

Η εντολή **SQL** που χρησιμοποιήθηκε ήταν η **SELECT**, η οποία μπορεί να επιστρέψει εγγραφές από έναν ή περισσότερους πίνακες.

Οι εγγραφές αυτές μπορεί να είναι ολόκληρες (με όλα τα πεδία τους) η να επιστραφούν εγγραφές με επιλεγμένα πεδία μόνο.

Για παράδειγμα:

SELECT * FROM stores

Ο κώδικας αυτός επιστρέφει όλες τις εγγραφές του πίνακα stores.

SELECT Firm FROM stores

Ο κώδικας αυτός επιστρέφει από όλες τις εγγραφές του πίνακα stores μόνο τα πεδία Firm.

Όλοι οι τύποι ερωτημάτων **SQL** μπορεί να συνοδεύονται με εντολές φιλτραρίσματος και ταξινόμησης.

Η εντολή **WHERE** μας δίνει την δυνατότητα εκτέλεσης συνθήκης ή φιλτραρίσματος.

Η εντολή **ORDER BY** μας δίνει την δυνατότητα εκτέλεσης ταξινόμησης δεδομένων είτε με αύξουσα (**ASC**) είτε με φθίνουσα (**DES**) σειρά.

Για παράδειγμα:

SELECT * FROM stores WHERE StoreId =1

Ο κώδικας αυτός επιστρέφει όλες τις εγγραφές του πίνακα stores όπου το πεδίο StoreId έχει την τιμή 1.

SELECT * FROM stores ORDER BY StoreId

Ο κώδικας αυτός επιστρέφει από όλες τις εγγραφές του πίνακα stores ταξινομημένες με βάση το πεδίο StoreId κατά αύξουσα σειρά.

INSERT

Με την εντολή **INSERT** μπορούμε να εισάγουμε δεδομένα σε έναν πίνακα στην βάση μία εγγραφή κάθε φορά.

INSERT INTO stores VALUES(1,'ΚΑΤΑΣΤΗΜΑ1')

Με την εντολή αυτή εισάγουμε στον πίνακα stores μία εγγραφή με τιμή στο πρώτο της πεδίο 1 (StoreId) και στο δεύτερο την τιμή 'ΚΑΤΑΣΤΗΜΑ1' (Firm).

Ο τύπος δεδομένων που εισάγουμε στον πίνακα θα πρέπει να συμφωνεί με τον τύπο δεδομένων των πεδίων του πίνακα που είναι ορισμένα στην βάση.

Αν τα δεδομένα είναι κείμενο θα πρέπει να περικλείονται σε εισαγωγικά ('').

DELETE

Με την εντολή **DELETE** μπορούμε να διαγράψουμε εγγραφές από έναν πίνακα στην βάση.

DELETE FROM stores

Με την εντολή αυτή διαγράφουμε όλες τις εγγραφές από τον πίνακα stores.

DELETE FROM stores WHERE StoreId=1

Με την εντολή αυτή διαγράφουμε όλες τις εγγραφές από τον πίνακα stores που το πεδίο StoreId είναι ίσο με 1.

UPDATE

Με την εντολή **UPDATE** μπορούμε να τροποποιήσουμε τιμές πεδίων σε εγγραφές ενός πίνακα στην βάση.

UPDATE stores SET Firm='ΚΑΤΑΣΤΗΜΑ10' WHERE StoreId=1

Με την εντολή αυτή αλλάζουμε την τιμή του πεδίου Firm των εγγραφών του πίνακα stores σε 'ΚΑΤΑΣΤΗΜΑ10' όπου το πεδίο StoreId είναι ίσο με 1.

ΣΥΝΔΕΣΗ JAVASCRIPT ΜΕ PHP

Η επικοινωνία του χρήστη με την **PHP** γίνεται μέσω της **javascript** και του αντικειμένου **XMLHttpRequest** για όλους τους περιηγητές διαδικτύου (browsers

πλην αυτών της **Microsoft**) ή ενός **ActiveXObject** για τον **Microsoft IE**, ενώ τα αποτελέσματα αποστέλλονται στον χρήστη με την εντολή **echo** της **PHP** ως τιμή με μορφή κειμένου που επιστρέφεται από την συνάρτηση **XMLHttpRequest** και την μέθοδο της **XMLHttpRequest.responseText**.

Στην παρούσα εφαρμογή ο κώδικας που δημιουργεί το αντικείμενο **XMLHttpRequest** βρίσκεται στο αρχείο **common.js** και έχει παρθεί αυτούσιος από το Internet.

```
function GetXmlHttp()
{
    xmlHttp=null;
    try
    {
        // Firefox, Opera 8.0+, Safari
        xmlHttp=new XMLHttpRequest();
    }
    catch (e)
    {
        // Internet Explorer
        try
        {
            xmlHttp=new ActiveXObject("MSXML2.XMLHTTP.3.0");
        }
        catch (e)
        {
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttp;
}
```

Το αντικείμενο **XMLHttpRequest** παίρνει τρεις παραμέτρους.

XMLHttpRequest('μέθοδος', 'αρχείο', [true][false])

Η πρώτη παράμετρος είναι η μέθοδος αποστολής των δεδομένων.

Δύο μέθοδοι είναι οι συνηθέστεροι, η μέθοδος GET και η μέθοδος PUT.

Στην παρούσα εφαρμογή χρησιμοποιείτε η GET.

Σε αυτήν την μέθοδο τα δεδομένα που αποστέλλονται προσαρτούνται στο όνομα του αρχείου που καλείτε με τον τύπο μεταβλητή=δεδομένο.

π.χ. **validate.php?p='1234'&sid=1234567890.**

Στην πρώτη μεταβλητή προηγείτε το σύμβολο ? ενώ για τις υπόλοιπες το σύμβολο &.

Η δεύτερη μεταβλητή του αντικειμένου **XMLHttpRequest** είναι το όνομα του αρχείου που καλείτε με τα τυχόν δεδομένα προσαρτημένα σε αυτό.

Η Τρίτη μεταβλητή είναι μία τιμή μεταξύ **true** και **false** η οποία σημαίνει αν ο κώδικας θα περιμένει την απάντηση του **SERVER** πριν συνεχίσει την εκτέλεση του ή όχι.

Στην παρούσα εφαρμογή χρησιμοποιείτε πάντα η τιμή **false**.

Τα δεδομένα που αποστέλλονται με την μέθοδο **GET** στα αρχεία PHP αποθηκεύονται σε έναν πίνακα όπου είναι προσβάσιμα με την μεταβλητή τύπου πίνακα **\$_GET[]**.

Στο παραπάνω παράδειγμα η τιμή **\$_GET['p']** είναι **'1234'**

Σε όλες τις κλήσεις αρχείων PHP ή ιστοσελίδων είτε υπάρχουν προσαρτημένα δεδομένα είτε όχι καλή πρακτική αποτελεί η αποστολή ενός μεγάλου τυχαίου αριθμού προς αποφυγή του φαινομένου **caching** (ανάκτησης παλαιότερων αποθηκευμένων δεδομένων).

Αυτό επιτυγχάνεται με την εντολή της **javascript Math.random()**.

Το φαινόμενο αυτό είναι συνηθισμένο στους **WEB SERVERS** όπου για την ταχύτερη εξυπηρέτηση των αιτημάτων που δέχονται και την λιγότερη απασχόληση των πόρων τους για την παραγωγή ομοίων αποτελεσμάτων, οι **WEB SERVERS** αποθηκεύουν έναν αριθμό αποτελεσμάτων και αιτημάτων χρηστών σε μια προσωρινή μνήμη (**cache**), ώστε αν ζητηθεί η ίδια ιστοσελίδα ή τα ίδια αποτελέσματα αιτήματος να στέλνουν τα υπάρχοντα.

Στην παρούσα εφαρμογή τα δεδομένα που προσαρτούνται στα αρχεία **PHP** κωδικοποιούνται μέσω της συνάρτησης **urlencode**.

Η συνάρτηση παίρνει ως παράμετρο τα δεδομένα προς προσάρτηση με μορφή κειμένου και τα επιστρέφει κωδικοποιημένα.

Ο κώδικας της βρίσκεται στο αρχείο **common.js** και είναι παρμένος αυτούσιος από το internet.

```

function urlencode(str)
{
    var hexStr = function (dec) {
        return '%' + dec.toString(16).toUpperCase();
    };

    var ret = "",
        unreserved = /[\w.-]/; // A-Za-z0-9_.- // Tilde is not here for historical reasons;
to preserve it, use rawurlencode instead
    str = (str+"").toString();

    for (var i = 0, dl = str.length; i < dl; i++) {
        var ch = str.charAt(i);
        if (unreserved.test(ch)) {
            ret += ch;
        }
        else {
            var code = str.charCodeAt(i);
            // Reserved assumed to be in UTF-8, as in PHP
            if (code === 32) {
                ret += '+'; // %20 in rawurlencode
            }
            else if (code < 128) { // 1 byte
                ret += hexStr(code);
            }
            else if (code >= 128 && code < 2048) { // 2 bytes
                ret += hexStr((code >> 6) | 0xC0);
                ret += hexStr((code & 0x3F) | 0x80);
            }
            else if (code >= 2048 && code < 65536) { // 3 bytes
                ret += hexStr((code >> 12) | 0xE0);
                ret += hexStr(((code >> 6) & 0x3F) | 0x80);
                ret += hexStr((code & 0x3F) | 0x80);
            }
        }
    }
}

```



```

    }
    else if (code >= 65536) { // 4 bytes
        ret += hexStr((code >> 18) | 0xF0);
        ret += hexStr(((code >> 12) & 0x3F) | 0x80);
        ret += hexStr(((code >> 6) & 0x3F) | 0x80);
        ret += hexStr((code & 0x3F) | 0x80);
    }
}
}
return ret;
}

```

Η κωδικοποίηση των δεδομένων είναι απαραίτητη μόνο για τους περιηγητές (browsers) της Microsoft, οι υπόλοιποι κωδικοποιούν μόνοι τους τα δεδομένα στην κατάλληλη μορφή.

Τα αποτελέσματα που παράγονται από την **PHP** είναι προσβάσιμα μέσω του αντικειμένου **XMLHttpRequest.responseText**.

Τα αποτελέσματα αυτά είναι σε μορφή κειμένου.

Στην παρούσα εφαρμογή τα δεδομένα που επιστρέφονται είναι είτε **0** και **1** (true/false), είτε δεδομένα πίνακα είτε κώδικας **HTML**.

Στην περίπτωση που είναι δεδομένα πίνακα αυτά είναι οι τιμές των πεδίων των εγγραφών σειριακά διαχωριζόμενες από ένα διακριτικό σύμβολο που στην παρούσα εφαρμογή είναι το '@'.

Χρησιμοποιώντας την ιδιότητα αντικειμένων κειμένου της **javascript** **split('σύμβολο')** μπορούμε να μεταφέρουμε τα σειριακά αποτελέσματα σε μια μεταβλητή πίνακα.

Για παράδειγμα:

Έστω το αποτέλεσμα είναι πίνακας με δύο πεδία δύο εγγραφές.

Η τιμή του **XMLHttpRequest.responseText** είναι:

'πεδίο1'@'πεδίο2'@'πεδίο1'@πεδίο2

Αν θέσουμε την μεταβλητή **A = XMLHttpRequest.responseText.split("@")**

Τότε έχουμε:

A[0]='πεδίο1' πρώτη εγγραφή
A[1]='πεδίο2' πρώτη εγγραφή
A[2]='πεδίο1' δεύτερη εγγραφή
A[3]='πεδίο2' δεύτερη εγγραφή

ΕΚΚΙΝΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEX.PHP

Η αρχική σελίδα που φορτώνεται ονομάζεται «index.php».
Ο κώδικας HTML της σελίδας δημιουργεί τέσσερα πλαίσια.

```
<frameset rows="110,*" cols="*" frameborder="no" border="1"
framespacing="0">
  <frame src="indextop.php" name="indextop" scrolling="No"
noresize="noresize" id="indextop"/>
  <frameset cols="140*,560" frameborder="no" border="1" framespacing="0">
    <frame src="indexleft.php" name="indexleft" scrolling="No"
noresize="noresize" id="indexleft"/>
    <frame src=" empty.php " name="indexmain" id="indexmain" scrolling="no"/>
    <frame src="empty.php" name="indexright" id="indexright" scrolling="no"/>
  </frameset>
</frameset>
```

Το πάνω οριζόντιο πλαίσιο φορτώνει την σελίδα «indextop.php».

Το αριστερό πλαίσιο φορτώνει την σελίδα «indexleft.php».

Το κεντρικό (μεσαίο) πλαίσιο φορτώνει την σελίδα «empty.php».

Το δεξίο πλαίσιο φορτώνει την σελίδα «empty.php».

Η σελίδα «indextop.php» περιέχει τον κώδικα που παράγει τον τίτλο του ιστοτόπου.

```
<h1 style="font-size:40px; text-align:center"> ΔΙΑΔΙΚΤΥΑΚΗ ΣΥΓΚΡΙΤΙΚΗ
ΑΓΟΡΑ
```

Η σελίδα «empty.php» είναι κενή και δίνει μόνο το χρώμα φόντου στο πλαίσιο.

```
<body style="background-color:#FF9">
```

Η σελίδα «indexleft.php» περιέχει τον κώδικα που δημιουργεί το αντικείμενο εισαγωγής κειμένου για την είσοδο του κωδικού χρήστη, καθώς και τα τρία πλαίσια πτυσσόμενης λίστας που απεικονίζουν τα δεδομένα: κατηγορία, προϊόν και κατάσταση.

Υπάρχει επίσης και ο κώδικας που δημιουργεί το κουμπί με την ένδειξη «**ΕΜΦΑΝΙΣΗ**» ή «**ΣΥΓΚΡΙΣΗ**».

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEXLEFT.PHP

Η σελίδα ενσωματώνει ακόμα και δύο αρχεία κώδικα javascript το «**common.js**» και το «**indexleft.js**».

```
<SCRIPT LANGUAGE='JavaScript' SRC='common.js'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript' SRC='indexleft.js'></SCRIPT>

<body style="background-color:#FC0">
<p align="center">ΚΩΔ. ΔΙΑΧΕΙΡΗΣΤΗ
<input type='password' id='pass' size=14 onkeypress='PassKeyPress(event)'/>
<p align="center">ΚΑΤΗΓΟΡΙΕΣ<br/>
<select name=ComboName id="cat" size=1 style="width:90%"
onChange="FillProducts()"></select>
<p align="center">ΠΡΟΪΟΝΤΑ<br/>
<select name=ComboName id="prod" size=1 style="width:90%"
onChange="Check()"></select>
<p align="center">ΚΑΤΑΣΤΗΜΑΤΑ<br/>
<select name=ComboName id="stores" size=1 style="width:90%"
onChange="FillProducts()"></select>
<p align="center"><input type='button' id='comp' value="ΕΜΦΑΝΙΣΗ"
onclick='Compare()'/>
```

ΚΩΔΙΚΑΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΕΙΣΑΓΩΓΗΣ ΚΩΔΙΚΟΥ

Όπως φαίνεται στον παραπάνω κώδικα το αντικείμενο εισαγωγής κειμένου έχει το διακριτικό αναγνώρισης για τον κώδικα της javascript, **id='pass'** και την μέθοδο **onkeypress='PassKeyPress(event)'** η οποία εκτελείτε κάθε φορά που πατιέται ένα πλήκτρο μέσα στο αντικείμενο κωδικού.

Η συνάρτηση **PassKeyPress(event)** παίρνει ως παράμετρο το αντικείμενο **event** του συστήματος, το οποίο περιέχει μεταξύ άλλων και τον ASCII CODE του πλήκτρου που πατήθηκε (**event.KeyCode**).

Ο κώδικας της συνάρτησης βρίσκεται στο αρχείο **«indexleft.js»**.

```
function PassKeyPress(e)
{
    if(e.keyCode == 27)
    {
        document.getElementById("pass").value=""
    }

    if(e.keyCode == 13)
    {
        var url = "validate.php"
        xmlHttp = GetXmlHttp()
        if(xmlHttp != null)
        {
            url += "?p=" +
urlencode(document.getElementById("pass").value)
            url += "&sid=" + Math.random();
            try{
                xmlHttp.open("GET",url,false)
                xmlHttp.send(null)}
            catch(e) {history.back()}

            if (xmlHttp.readyState==4 && xmlHttp.status==200)
            {
                if(xmlHttp.responseText == 0)
                {
                    document.getElementById("pass").value = ""
                    alert("ΛΑΘΟΣ ΚΩΔΙΚΟΣ")
                }
            }
            else
```

```

        {
            parent.location.href = "admin.php"
        }
    }
}
else
    alert("No Connection")
}
}
}

```

Η συνάρτηση αποτελείται από δύο τμήματα ελέγχου (if {}).

Στο πρώτο ελέγχει αν η τιμή του e.keyCode είναι 27 δηλαδή **ESC** και στο δεύτερο αν είναι 13 δηλαδή **ENTER**.

Στην πρώτη περίπτωση καθαρίζει τα δεδομένα του αντικειμένου κωδικού ενώ στην δεύτερη καλεί το PHP αρχείο «**validate.php**» με παράμετρο τον κωδικό που εισήγαγε ο χρήστης.

Ο κώδικας του αρχείου «**validate.php**» ρωτάει τη βάση αν υπάρχει ο κωδικός και επιστρέφει ανάλογα μία τιμή. Αν δεν υπάρχει η τιμή αυτή είναι **0** αλλιώς **1**.

Αν δεν υπάρχει ο κωδικός τότε εμφανίζεται μήνυμα με την ένδειξη «**ΛΑΘΟΣ ΚΩΔΙΚΟΣ**», αλλιώς φορτώνεται η σελίδα «**admin.php**».

Ο κώδικας του αρχείου «**validate.php**» είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "SELECT * FROM passlist WHERE Pass='" . $_GET["p"] . "'";
$result = mysqli_query($Connection,$query);
if($result && mysqli_num_rows($result))
{
    $arr = mysqli_fetch_row($result);
}
}
}
}

```

```

$value = mt_rand();
$session = "INSERT INTO sessions VALUES(" . $value . ", " . time() . ")";
$res = mysqli_query($Connection,$session);
if($res)
{
    setcookie("session",$value);
    setcookie($value,$arr[2]);
    echo 1;
}
else
    echo 0;
}
else
    echo 0;
if($result)
    mysqli_free_result($result);
mysqli_close($Connection);
?>

```

Ο παραπάνω κώδικας, εφ' όσον ο κωδικός χρήστη είναι υπαρκτός, καταχωρεί στον πίνακα της βάσης **sessions**, έναν μεγάλο τυχαίο αριθμό ο οποίος παράγεται από την συνάρτηση της **PHP** [mt_rand\(\)](#).

Ο ίδιος αριθμός αποθηκεύεται στον υπολογιστή του χρήστη με ένα [Cookie](#) που έχει το όνομα **session** ενώ αποθηκεύεται και ένα δεύτερο [Cookie](#) το όνομα του οποίου είναι ο αριθμός αυτός και τα δεδομένα του είναι ο κωδικός καταστήματος του χρήστη.

Αυτή η τεχνική χρησιμοποιείται για να μπορεί η **PHP**, διαβάζοντας αυτά τα **Cookie**, να γνωρίζει ποιός χρήστης είναι συνδεδεμένος με την σελίδα διαχείρισης καθώς και ποια δικαιώματα πρόσβασης έχει.

ΚΩΔΙΚΑΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΠΤΥΣΣΟΜΕΝΗΣ ΛΙΣΤΑΣ

Ο κώδικας των αντικειμένων πτυσσόμενης λίστας είναι ο ίδιος, διαφέρουν μόνο οι παράμετροι που αποστέλλονται στην βάση.

Στα αντικείμενα αυτά χρησιμοποιούμε την μέθοδο τους **onchange()** με την οποία καλούμε τα αντίστοιχα κομμάτια κώδικα κάθε φορά που αλλάζει η τιμή τους. Η πτυσσόμενη λίστα που απεικονίζει τις κατηγορίες, καθώς και αυτή που απεικονίζει τα καταστήματα καλούν την συνάρτηση **FillProducts**, ενώ το αντικείμενο που απεικονίζει τα υπάρχοντα προϊόντα καλεί την συνάρτηση **check**. Κάθε φορά που αλλάζει κάποια επιλογή από τις κατηγορίες ή τα καταστήματα ο κώδικας αποστέλλει τις νέες τιμές αυτών των αντικειμένων και γεμίζει την λίστα των προϊόντων με τα ανάλογα αποτελέσματα.

Ο κώδικας της συνάρτησης **FillProducts** βρίσκεται στο αρχείο **indexleft.js**.

```
function FillProducts()
{
    var ProdIndex = 0;
    Check()
    ComboProd = document.getElementById("prod")
    ComboCat = document.getElementById("cat")
    ComboStores = document.getElementById("stores")
    if(ComboProd)
        ProdIndex = ComboProd.selectedIndex
    var url="getprod.php"
    if(ComboCat.options[ComboCat.selectedIndex].value)
        url += "?cid=" +
urlencode(ComboCat.options[ComboCat.selectedIndex].value)
    else
        url += "?cid=" + urlencode("ΟΛΑ")
    if(ComboStores.options[ComboStores.selectedIndex].value)
        url += "&stid=" +
urlencode(ComboStores.options[ComboStores.selectedIndex].value)
    else
        url += "&stid=" + urlencode("ΟΛΑ")
    url += "&sid=" +Math.random();
    if(xmlHttp != null)
    {
        try{
```

```

xmlHttp.open("GET",url,false)
xmlHttp.send(null)
catch(e) {}
if (xmlHttp.readyState==4 && xmlHttp.status==200)
{
    for (m=ComboProd.options.length-1;m>=0;m--)
    {
        ComboProd.options[m]=null
    }
    if(xmlHttp.responseText == 0)
        ComboProd.options[0] = new Option("ΔΕΝ
ΥΠΑΡΧΟΥΝ", 0)
    else
    {
        var array = xmlHttp.responseText.split("@")
        ComboProd.options[ComboProd.length] = new
Option("ΟΛΑ","ΟΛΑ")
        for(i=0;i<array.length - 1;i++)
        {
            ComboProd.options[ComboProd.length] = new
Option(array[i],array[Number(i)+1])
            i++
        }
        if(ProdIndex != -1)
            ComboProd.selectedIndex = ProdIndex
    }
}
}
}
}

```

Η συνάρτηση **check** που εκτελείτε όταν αλλάζει η τιμή της λίστας των προϊόντων, διαβάζει τις τιμές των τριών πτυσσόμενων λιστών και αν αυτός είναι: συγκεκριμένο προϊόν συγκεκριμένη κατηγορία και η τιμή της λίστας καταστημάτων είναι **ΟΛΑ** τότε αλλάζει την ένδειξη του κουμπιού, που βρίσκεται κάτω από τα αντικείμενα λίστας, σε **ΣΥΓΚΡΙΣΗ** ενώ για κάθε άλλη περίπτωση είναι **ΕΜΦΑΝΙΣΗ**. Με το πάτημα του κουμπιού αυτού καλείτε η συνάρτηση **compare** μέσω της μεθόδου του **onclick**.

Το αποτέλεσμα της συνάρτησης **compare** είναι η φόρτωση νέας σελίδας, που εμφανίζει τα αποτελέσματα των επιλογών του χρήστη στο κεντρικό πλαίσιο (FRAME), το οποίο αρχικά είναι κενό, με την μορφή πίνακα. Λεπτομερέστερη ανάλυση ακολουθεί στην επόμενη ενότητα.

Ο κώδικας της συνάρτησης **compare** βρίσκεται στο αρχείο **indexleft.js**

```
function Compare()
{
    var url = "compare.php?pid=" +
urlencode(document.getElementById("prod").options[document.getElementById("pr
od").selectedIndex].value)
    url += "&cid=" +
urlencode(document.getElementById("cat").options[document.getElementById("cat")
.selectedIndex].value)
    url += "&stid=" +
urlencode(document.getElementById("stores").options[document.getElementById("st
ores").selectedIndex].value)
    url += "&sid=" + Math.random()
    var url2 = "compare.php?pid=" +
document.getElementById("prod").options[document.getElementById("prod").select
edIndex].value
    url2 += "&cid=" +
document.getElementById("cat").options[document.getElementById("cat").selectedI
ndex].value
    url2 += "&stid=" +
document.getElementById("stores").options[document.getElementById("stores").sele
ctedIndex].value
    url2 += "&sid=" + Math.random()
    parent.indexmain.location.href = url
}
```

Ο κώδικας **'parent.indexmain.location.href = url'** φορτώνει στο μεσαίο πλαίσιο (FRAME) με id «indexmain» την νέα σελίδα με τις τυχόν παραμέτρους της.

Τέλος μετά την φόρτωση της σελίδας **indexleft.php**, εκτελείτε ο κώδικας **javascript:**

```
<script
type="text/javascript">FillStores();FillCat();FillProducts();document.getElementById
("pass").focus(</script>
```

Ο κώδικας αυτός καλεί διαδοχικά τις συναρτήσεις **FillStores**, **FillCat**, **FillProducts** και δίνει την εστίαση της ιστοσελίδας στο αντικείμενο εισαγωγής κωδικού.

Οι συναρτήσεις αυτές γεμίζουν τα πλαίσια λίστας με τα υπάρχοντα δεδομένα της βάσης και θέτουν τις αρχικές τιμές τους στην ένδειξη **ΟΛΑ**.

Αν η βάση ή κάποιος πίνακας είναι κενός τότε έχουν όλα ή αντίστοιχα κατά περίπτωση, την ένδειξη **ΔΕΝ ΥΠΑΡΧΟΥΝ**.

Ο κώδικας των συναρτήσεων αυτών βρίσκεται επίσης στο αρχείο **indexleft.js**.

```
function FillStores()
{
    var url="getstor.php"
    url += "?sid=" +Math.random();
    xmlhttp = GetXmlHttp()
    if(xmlHttp != null)
    {
        try{
            xmlhttp.open("GET",url,false)
            xmlhttp.send(null)}
        catch(e) {}
        if (xmlHttp.readyState===4 && xmlhttp.status===200)
        {
            var combo = document.getElementById("stores")
            if(xmlHttp.responseText == 0)
                combo.options[0] = new Option("ΔΕΝ ΥΠΑΡΧΟΥΝ")
            else
            {
                var array = xmlhttp.responseText.split("@")
                combo.options[combo.length] = new
Option("ΟΛΑ","ΟΛΑ")
                for(i=0;i<array.length - 1;i++)
```

```

        {
            combo.options[combo.length] = new
Option(array[Number(i)+1],array[i])
            i++;
        }
    }
}
}

function FillCat()
{
    var url="getcat.php"
    url += "?stid=" +
urlencode(document.getElementById("stores").options[document.getElementById("st
ores").selectedIndex].value)
    url += "&sid=" + Math.random();
    xmlHttp = GetXmlHttp()
    if(xmlHttp != null)
    {
        try{
            xmlHttp.open("GET",url,false)
            xmlHttp.send(null)}
        catch(e) {}
        if (xmlHttp.readyState==4 && xmlHttp.status==200)
        {
            var combo = document.getElementById("cat")
            if(xmlHttp.responseText == 0)
                combo.options[0] = new Option("ΔΕΝ ΥΠΙΑΡΧΟΥΝ")
            else
            {
                var array = xmlHttp.responseText.split("@")
                combo.options[combo.length] = new

```

```

Option("ΟΛΑ","ΟΛΑ")
        for(i=0;i<array.length - 1;i++)
        {
                combo.options[combo.length] = new
Option(array[Number(i)+1],array[i])
                i++;
        }
    }
}
}
}
}

```

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ COMPARE.PHP

Όπως αναφέρθηκε στην προηγούμενη ενότητα η σελίδα αυτή απεικονίζει τα αποτελέσματα του συνδυασμού των επιλογών του χρήστη στα πλαίσια πτυσσόμενης λίστας που βρίσκονται στο αριστερό πλαίσιο (FRAME).

Δύο κατηγορίες εμφάνισης υπάρχουν και είναι οι εξής:

α) *Εμφάνιση περισσότερων του ενός προϊόντων από μία ή περισσότερες κατηγορίες και ένα ή περισσότερα καταστήματα.*

β) *Σύγκριση τιμών ενός προϊόντος συγκεκριμένης κατηγορίας σε όλα τα καταστήματα.*

Τα αποτελέσματα και στις δύο περιπτώσεις προβάλλονται με την μορφή πίνακα τεσσάρων στηλών.

Στην πρώτη στήλη απεικονίζονται τα **καταστήματα** στην δεύτερη οι **κατηγορίες των προϊόντων** στην τρίτη τα **προϊόντα** και στην τέταρτη οι **τιμές των προϊόντων**.

Η διαφορά των δύο κατηγοριών είναι ότι μεν στην πρώτη η παρουσίαση των αποτελεσμάτων γίνεται με αύξουσα σειρά ταξινόμησης με βάση τη σειρά των πεδίων **ΚΑΤΑΣΤΗΜΑΤΑ, ΚΑΤΗΓΟΡΙΕΣ, ΠΡΟΪΟΝΤΑ, ΤΙΜΗ**.

Ενώ στην δεύτερη με βάση την μικρότερη τιμή.

Ο κώδικας που δημιουργεί και προβάλλει τα αποτελέσματα είναι στο αρχείο **compare.php**.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head>
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<META HTTP-EQUIV="EXPIRES" CONTENT="Mon, 22 Jul 2000 11:12:01
GMT">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ΣΥΓΚΡΙΣΗ ΤΙΜΩΝ</title>
<SCRIPT LANGUAGE='JavaScript' SRC='common.js'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript' SRC='compare.js'></SCRIPT>
<style type="text/css">
table {
    background-color:#FFF;
    border-collapse:collapse;
}
div.MyDiv {
    overflow:auto;
    height:370px;
    width:560px;
}
</style>
</head>
<body style="background-color:#FF9">
<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$compare = false;

if($_GET["pid"]=="ΟΛΑ")
    $pid = " Like'%'";
else

```

```

        $pid = "=" . $_GET["pid"];
if($_GET["cid"]=="OAA")
    $cid = " Like'%'";
else
    $cid = "=" . $_GET["cid"];
if($_GET["stid"]=="OAA")
    $stid = " Like'%'";
else
    $stid = "=" . $_GET["stid"];

if($res)
    mysqli_free_result($res);

if($_GET["cid"]!="OAA" && $_GET["pid"]!="OAA" && $_GET["stid"]=="OAA")
{
    $query = "SELECT p.`StoreId`, p.`ProductId`, p.`CategoryId`, s.`Firm`,
c.`Name`, p.`Product`, p.`Price` FROM products p, stores s, categories c WHERE
p.StoreId=s.StoreId AND p.`CategoryId`=c.`CategoryId` AND p.`ProductId` " . $pid .
" AND p.`CategoryId` " . $cid . " AND p.`StoreId` " . $stid . " ORDER BY p.`Price`
ASC, s.`StoreId` ASC";
    $compare = true;
}
else
    $query = "SELECT p.`StoreId`, p.`ProductId`, p.`CategoryId`, s.`Firm`,
c.`Name`, p.`Product`, p.`Price` FROM products p, stores s, categories c WHERE
p.StoreId=s.StoreId AND p.`CategoryId`=c.`CategoryId` AND p.`ProductId` " . $pid .
" AND p.`CategoryId` " . $cid . " AND p.`StoreId` " . $stid . " ORDER BY p.`StoreId`
ASC, p.`CategoryId` ASC, p.`ProductId` ASC";

$result = mysqli_query($Connection,$query);
if($result && mysqli_num_rows($result))
{
    if($compare)

```

```

{
    echo "<table border='1'>";
    echo "<caption style='font-weight:bold'>ΣΥΓΚΡΙΣΗ ΤΙΜΩΝ
ΠΙΠΟΪΟΝΤΟΣ</caption>";
    echo "<tr>";
    echo "<td style='width:150px;text-align:center'>" . "ΚΑΤΑΣΤΗΜΑ" .
"</td>";
    echo "<td style='width:150px;text-align:center'>" . "ΚΑΤΗΓΟΡΙΑ" .
"</td>";
    echo "<td style='width:150px;text-align:center'>" . "ΠΙΠΟΪΟΝ" .
"</td>";
    echo "<td style='width:80px;text-align:center'>" . "ΤΙΜΗ" . "</td>";
    echo "</tr>";
    echo "</table>";
    echo "<div id='scroll' class='MyDiv'>";
    echo "<table border='1'>";
    for($i=0;$i<mysql_num_rows($result);$i++)
    {
        echo "<tr>";
        $arr = mysql_fetch_row($result);
        if($i==0)
            $prod = $arr[5];
        for($k=3;$k<mysql_num_fields($result);$k++)
        {
            if($i==0)
            {
                if($k==3)
                    echo "<td onclick='AddItem(" . $arr[1] .
", " . $arr[0] . ", " . $arr[$k+3] . ", " . $arr[2] . ")'
style='width:150px;cursor:pointer;text-decoration:underline;background-color:red'>" .
$arr[$k];
                else if($k<6)
                    echo "<td style='width:150px;text-

```

```

align:center;background-color:red">" . $arr[$k];
        else
            echo "<td style='width:80px;text-
align:center;background-color:red">" . $arr[$k];
            echo "</td>";
        }
        else
        {
            if($k==3)
                echo "<td onclick='AddItem(" . $arr[1] .
", " . $arr[0] . " , " . $arr[$k+3] . " , " . $arr[2] . ")'
style='width:150px;cursor:pointer;text-decoration:underline'" . $arr[$k];
            else if($k<6)
                echo "<td style='width:150px;text-
align:center'" . $arr[$k];
            else
                echo "<td style='width:80px;text-
align:center'" . $arr[$k];
            echo "</td>";
        }
    }
    echo "</tr>";
}
}
else
{
    echo "<table border='1'">";
    echo "<caption style='font-weight:bold'>ΕΜΦΑΝΙΣΗ ΠΡΟΪΟΝΤΩΝ
ΚΑΙ ΤΙΜΩΝ</caption>";
    echo "<tr>";
    echo "<td style='width:150px;text-align:center'" . "ΚΑΤΑΣΤΗΜΑ" .
"</td>";
    echo "<td style='width:150px;text-align:center'" . "ΚΑΤΗΓΟΡΙΑ" .

```



```

"</td>";
        echo "<td style='width:150px;text-align:center'>" . "TIPOÏON" .
"</td>";
        echo "<td style='width:80px;text-align:center'>" . "TIMH" . "</td>";
        echo "</tr>";
        echo "</table>";
        echo "<div id='scroll' class='MyDiv'>";
        echo "<table border='1'>";
        for($i=0;$i<mysql_num_rows($result);$i++)
        {
            echo "<tr>";
            $sarr = mysql_fetch_row($result);
            for($k=3;$k<mysql_num_fields($result);$k++)
            {
                if($i==0)
                {
                    if($k==3)
                        echo "<td id='" . $sarr[0] . "@" . $sarr[1]
."'" onclick='AddItem(" . $sarr[0] . "," . $sarr[1] . "," . $sarr[$k+3] . "," . $sarr[2] . "'
style='width:150px;cursor:pointer;text-decoration:underline;'>" . $sarr[$k];
                    else if($k<6)
                        echo "<td style='width:150px;text-
align:center;'>" . $sarr[$k];
                    else
                        echo "<td style='width:80px;text-
align:center;'>" . $sarr[$k];
                    echo "</td>";
                }
                else
                {
                    if($k==3)
                        echo "<td id='" . $sarr[0] . "@" . $sarr[1]
."'" onclick='AddItem(" . $sarr[0] . "," . $sarr[1] . "," . $sarr[$k+3] . "," . $sarr[2] . "'

```

```

style='width:150px;cursor:pointer;text-decoration:underline'" . $arr[$k];
        else if($k<6)
            echo "<td style='width:150px;text-align:center'" . $arr[$k];
        else
            echo "<td style='width:80px;text-align:center'" . $arr[$k];
        echo "</td>";
    }
}
echo "</tr>";
}
}
echo "</table>";
echo "</div>";

if($compare)
{
    echo "<p> Ο παραπάνω πίνακας απεικονίζει την σύγκριση των τιμών του προϊόντος <B><U>" . $prod . "</B></U> σε όλα τα διαθέσιμα καταστήματα";
    echo "<br><U><B>Για να παραγγείλετε κάποιο προϊόν επιλέξτε το αντίστοιχο υπογραμμισμένο κατάστημα από τον πίνακα";
}
else
{
    echo "<p> Ο παραπάνω πίνακας εμφανίζει τα επιλεγμένα προϊόντα στα επιλεγμένα καταστήματα";
    echo "<br><U><B>Για να παραγγείλετε κάποιο προϊόν επιλέξτε το αντίστοιχο υπογραμμισμένο κατάστημα από τον πίνακα";
}

echo "<script>document.getElementById('scroll').scrollTop=0</script>";
}

```

```

if($result)
    mysqli_free_result($result);
mysqli_close($Connection);
?>
</body>
</html>

```

Σε όλες τις περιπτώσεις τα δεδομένα των πεδίων της πρώτης στήλης, στην προκειμένη περίπτωση τα καταστήματα, είναι υπογραμμισμένα για να υποδηλώσουν ότι ο χρήστης έχει την δυνατότητα να τα επιλέξει.

Επιλέγοντας το κατάστημα που βρίσκεται κάποιο συγκεκριμένο προϊόν εκτελείτε η μέθοδος κυψέλης πίνακα (<td>) **onclick**

```

echo "<td onclick='AddItem(" . $arr[1] . "," . $arr[0] . "," . $arr[$k+3] . "," . $arr[2] .
")' style='width:150px;cursor:pointer;text-decoration:underline'>" . $arr[$k];

```

η οποία εκτελεί την συνάρτηση **AddItem** που ο κώδικας της βρίσκεται στο αρχείο **compare.js**.

```

function AddItem(std, pid, pr, cat)
{
    var url = "additem.php"
    url += "?pid=" + urlencode(pid)
    url += "&std=" + urlencode(std)
    url += "&pr=" + urlencode(pr)
    url += "&cat=" + urlencode(cat)
    url += "&sid=" + Math.random();
    xmlhttp = GetXmlHttp()
    if(xmlhttp != null)
    {
        try{
            xmlhttp.open("GET",url,false)
            xmlhttp.send(null)}
        catch(e) {}
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {

```

```

        if(xmlHttp.responseText == 0)
            alert("Η ΚΑΤΑΧΩΡΗΣΗ ΔΕΝ ΕΓΙΝΕ")
        else
        {
            parent.indexright.location.href = "indexright.php"
        }
    }
}
}

```

Η παραπάνω συνάρτηση παίρνει τέσσερις παραμέτρους που είναι οι εξής:

1. Κωδικός καταστήματος.
2. Κωδικός προϊόντος.
3. Τιμή προϊόντος
4. Κατηγορία προϊόντος

Το αποτέλεσμα της εκτέλεσης αυτού του κώδικα είναι η καταχώριση του επιλεγμένου προϊόντος στην λίστα παραγγελίας του πελάτη, δηλαδή στον πίνακα **orders** (παραγγελίες) της βάσης.

Αυτό επιτυγχάνεται με την κλήση του αρχείου **additem.php** το οποίο καλείτε με τις παραπάνω τέσσερις παραμέτρους.

Ο κώδικας του αρχείου **additem.php** είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

if(!isset($_COOKIE["customer"]))
{
    $value = mt_rand();
    setcookie("customer",$value);
}

```

```

$query = "INSERT INTO orders VALUES(" . $value . "," . $_GET["pid"] .
"," . $_GET["stdid"] . "," . $_GET["cat"] . "," . $_GET["pr"] . ", 1, " . time() . ")";
$result = mysqli_query($Connection,$query);
if($result)
    echo 1;
else
    echo 0;
}
else
{
    $s = "SELECT Stamp FROM orders WHERE OrderSid=" .
$_COOKIE["customer"] . """;
    $r = mysqli_query($Connection,$s);
    if($r && mysqli_num_rows($r))
    {
        $stamp = mysqli_fetch_row($r);

        $ask = "SELECT ProductId FROM orders WHERE OrderSid=" .
$_COOKIE["customer"] . " AND ProductId=" . $_GET["pid"] . " AND StoreId=" .
$_GET["stdid"] . " AND CategoryId=" . $_GET["cat"] . """;
        $res = mysqli_query($Connection,$ask);
        if($res && mysqli_num_rows($res))
            $query = "UPDATE orders SET Quantity = Quantity + 1
WHERE OrderSid=" . $_COOKIE["customer"] . " AND ProductId=" . $_GET["pid"]
. " AND StoreId=" . $_GET["stdid"] . " AND CategoryId=" . $_GET["cat"] . """;
        else
            $query = "INSERT INTO orders VALUES(" .
$_COOKIE["customer"] . "," . $_GET["pid"] . "," . $_GET["stdid"] . "," . $_GET["cat"]
. "," . $_GET["pr"] . ", 1, " . $stamp[0] . ")";

        $result = mysqli_query($Connection,$query);
        if($result)
            echo 1;
    }
}
}

```

```

else
    echo 0;
}
else
{
    $query = "INSERT INTO orders VALUES(" .
$_COOKIE["customer"] . "," . $_GET["pid"] . "," . $_GET["stid"] . "," .
$_GET["cat"] . "," . $_GET["pr"] . ", 1, " . time() . ")";
    $result = mysqli_query($Connection,$query);
    if($result)
        echo 1;
    else
        echo 0;
}
if($r)
    mysqli_free_result($r);
if($res)
    mysqli_free_result($res);
}
mysqli_close($Connection);
?>

```

Στον παραπάνω κώδικα επειδή τα στοιχεία του υποψήφιου πελάτη είναι άγνωστα μέχρι τώρα, χρησιμοποιείτε πάλι η τεχνική των **Cookies** για να μπορεί η εφαρμογή να διαχωρίζει τις παραγγελίες του κάθε άγνωστου υποψήφιου πελάτη από τους υπόλοιπους που μπορεί να χρησιμοποιούν ταυτόχρονα το Site.

Δημιουργείτε ένα **Cookie** με όνομα **customer** και δεδομένα έναν μεγάλο τυχαίο αριθμό που αποθηκεύεται στον υπολογιστή του χρήστη ενώ καταχωρείτε και στη βάση στον πίνακα **orders** μαζί με τα δεδομένα της παραγγελίας.

Αν επιλέξουμε το ίδιο προϊόν περισσότερο από μία φορά, αυτό δεν καταχωρείτε ως νέα εγγραφή αλλά αυξάνεται η ποσότητα της παραγγελίας του συγκεκριμένου προϊόντος στον πίνακα παραγγελία πελάτη που απεικονίζεται στο δεξιό πλαίσιο (FRAME) και περιγράφεται στην επόμενη ενότητα.

Τελειώνοντας ο κώδικας της συνάρτησης **AddItem** φορτώνει την σελίδα **indexright.php** στο δεξιό πλαίσιο (FRAME) το οποίο απεικονίζει τα επιλεγμένα προϊόντα προ αγορά του χρήστη με την μορφή πίνακα.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ INDEXRIGHT.PHP

Ο κώδικας αυτού του αρχείου παράγει μία ιστοσελίδα η οποία απεικονίζει στο επάνω μέρος της δύο κουμπιά.

Το αριστερό έχει την ένδειξη **ΠΡΟΩΘΗΣΗ** και το δεξιό την ένδειξη **ΑΚΥΡΩΣΗ**. Μετά ακολουθεί ένας πίνακας έξι στηλών που απεικονίζει τα επιλεγμένα προϊόντα του χρήστη.

Η στήλες περιλαμβάνουν τα εξής πεδία του πίνακα των παραγγελιών.

ΚΑΤΑΣΤΗΜΑ, ΚΑΤΗΓΟΡΙΑ, ΠΡΟΪΟΝ, ΠΟΣΟΤΗΤΑ (ΤΕΜ), ΤΙΜΗ, ΑΞΙΑ

Στο πεδίο ποσότητα απεικονίζεται το πόσες φορές έχει επιλεγεί το συγκεκριμένο προϊόν. Κάθε φορά που επιλέγεται αυξάνεται η ποσότητα κατά μία μονάδα.

Στο πεδίο αξία απεικονίζεται το γινόμενο της ποσότητας επί της τιμής του συγκεκριμένου προϊόντος.

Τέλος κάτω από τον πίνακα απεικονίζεται η συνολική αξία της παραγγελίας.

Ο κώδικας του αρχείου **indexright.php** είναι:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<META HTTP-EQUIV="EXPIRES" CONTENT="Mon, 22 Jul 2000 11:12:01
GMT">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>right</title>
<SCRIPT LANGUAGE='JavaScript' SRC='common.js'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript' SRC='indexright.js'></SCRIPT>
<style type="text/css">
table {
    background-color:#FFF;
    border-collapse:collapse;
```

```

    }
div.MyDiv {
    overflow:auto;
    height:322px;
    width:540px;
}
</style>
</head>
<body style="background-color:#FC0">

<input type="button" id="forward" value="ΠΡΟΩΘΗΣΗ"
onclick='ForwardOrder()'/>
<input type="button" id="cancel" value="ΑΚΥΡΩΣΗ" onclick='CancelOrder()'/>

<br>
<br>
<br>
<br>
<div align="left">
<table border="1">
<caption style="font-weight:bold">ΠΑΡΑΓΓΕΛΙΑ</caption>
<thead><th style="width:125px">ΚΑΤΑΣΤΗΜΑ</th><th
style="width:115px">ΚΑΤΗΓΟΡΙΑ</th><th style="width:114px">ΠΡΟΪΟΝ</th><th
style="width:50px">ΤΕΜ</th>
<th style="width:50px">ΤΙΜΗ</th><th style="width:50px">ΑΞΙΑ</th></thead>
</table>
</div>
<div id="scroll" class="MyDiv" style="position:absolute;">
<table id="order" border="1">
<?php
$host="localhost";
$user="admin";
$password="1234";

```



```

$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$total = 0;
$query = "SELECT DISTINCTROW o.ProductId, o.StoreId, o.CategoryId, s.Firm,
c.Name, p.`Product`, o.`Quantity`, o.`Price` FROM stores s, orders o LEFT OUTER
JOIN categories c ON c.CategoryId=o.CategoryId LEFT OUTER JOIN products p
ON p.CategoryId=c.CategoryId WHERE s.`StoreId`=o.`StoreId` AND
p.ProductId=o.ProductId AND o.OrderSid="" . $_COOKIE["customer"] . """;
$result = mysqli_query($Connection,$query);
if($result && mysqli_num_rows($result))
{
    for($i=0;$i<mysqli_num_rows($result);$i++)
    {
        $arr = mysqli_fetch_row($result);
        echo "<tr>";
        for($k=3;$k<mysqli_num_fields($result);$k++)
        {
            if($k==3)
                echo "<td onclick='RemoveItem(" . $arr[0] . ", " .
                $arr[1] . ", " . $_COOKIE["customer"] . ", " . $arr[2] . ")'
                style='width:125px;cursor:pointer;text-decoration:underline'>" . $arr[$k];
            else if($k<6)
                echo "<td style='width:115px'>" . $arr[$k];
            else
                echo "<td style='width:50px;text-align:center'>" .
                $arr[$k];
            echo "</td>";
        }
        $total += ($arr[$k-2] * $arr[$k-1]);
        echo "<td style='width:50px;text-align:center'>" . ($arr[$k-2] *
        $arr[$k-1]) . "</td>";
        echo "</tr>";
    }
}

```

```

    }
}
else
{
    if($result)
        mysqli_free_result($result);
    mysqli_close($Connection);
    echo "<script type='text/javascript'>parent.location.href =
'index.php'</script>";
}
if($result)
    mysqli_free_result($result);
if($Connection)
mysqli_close($Connection);
?>
</table>
</div>
<div style="position:absolute; bottom:5%">
<table style="empty-cells:hide; background-color:#FC0">
<tr><td style="font-weight:bold">ΣΥΝΟΛΙΚΗ ΑΞΙΑ</td><td style="width:340px;
background-color:#FC0"></td><td style="width:50px; text-align:center; font-
weight:bold"><?php echo $total; ?></td></tr>
</table>
</div>
<script>document.getElementById('scroll').scrollTop=80900</script>
</body>
</html>

```

Και σε αυτόν το πίνακα τα πεδία της πρώτης στήλης είναι υπογραμμισμένα, μπορεί δηλαδή ο χρήστης να τα επιλέξει.

Επιλέγοντας κάποια εγγραφή, αν η ποσότητα του προϊόντος της είναι μεγαλύτερη από 1, τότε αυτή μειώνεται κατά μία μονάδα, ενώ αν είναι 1 τότε διαγράφεται η συγκεκριμένη εγγραφή από την λίστα των παραγγελιών.

Αν διαγραφούν όλες οι εγγραφές τότε διαγράφεται όλος ο πίνακας και επανέρχεται το πλαίσιο (FRAME) στην αρχική του κατάσταση.

Η επιλογή προς μείωση ποσότητας ή διαγραφής κάποιας εγγραφής γίνεται με την ανάγνωση του τυχαίου αριθμού που είναι αποθηκευμένος με την τεχνική των **Cookies** και η χρησιμοποίηση του ως φίλτρο στην διαγραφή ή τροποποίηση της συγκεκριμένης εγγραφής από τον πίνακα **orders** της βάσης.

Επιλογή εγγραφών με βάση τον αριθμό που είναι αποθηκευμένος στον υπολογιστή του πελάτη με το **\$_COOKIE["customer"]**.

```
$query = "SELECT DISTINCTROW o.ProductId, o.StoreId, o.CategoryId, s.Firm, c.Name, p.`Product`, o.`Quantity`, o.`Price` FROM stores s, orders o LEFT OUTER JOIN categories c ON c.CategoryId=o.CategoryId LEFT OUTER JOIN products p ON p.CategoryId=c.CategoryId WHERE s.`StoreId`=o.`StoreId` AND p.ProductId=o.ProductId AND o.OrderSid=" . $_COOKIE["customer"] . """;
```

Η μείωση της ποσότητας ή η διαγραφή μιας εγγραφής γίνεται με την συνάρτηση **RemoveItem** η οποία καλείτε από την μέθοδο **onclick** του αντικειμένου κυψέλης πίνακα (<td onclick=...>).

Ο κώδικας της συνάρτησης **RemoveItem** βρίσκεται στο αρχείο **indexright.js**.

```
function RemoveItem(pid, stid, sid, cat)
{
    var url = "removeitem.php"
    url += "?pid=" + pid
    url += "&stid=" + stid
    url += "&sid=" + sid
    url += "&cat=" + cat
    url += "&s=" + Math.random();
    xmlHttp = GetXmlHttp()
    if(xmlHttp != null)
    {
        try{
            xmlHttp.open("GET", url, false)
            xmlHttp.send(null)}
        catch(e) {}
        if (xmlHttp.readyState==4 && xmlHttp.status==200)
```

```

        {
            if(xmlHttp.responseText == 0)
                alert("Η ΚΑΤΑΧΩΡΗΣΗ ΔΕΝ ΕΓΙΝΕ")
            else
            {
                location.reload()
            }
        }
    }
}

```

Η περιγραφή του κώδικα είναι ίδια με αυτή της συνάρτησης **AddItem** και παίρνει ακριβώς τις ίδιες παραμέτρους.

Η διαφορά είναι ότι καλεί το αρχείο **removeitem.php** ο κώδικας του οποίου είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$s = "SELECT Quantity FROM orders WHERE OrderSid=" . $_GET["sid"] . " AND
ProductId=" . $_GET["pid"] . " AND StoreId=" . $_GET["stid"] . " AND
CategoryId=" . $_GET["cat"];
$r = mysqli_query($Connection,$s);
if($r && mysqli_num_rows($r))
{
    $quant = mysqli_fetch_row($r);

    if($quant[0] > 1)
        $query = "UPDATE orders SET Quantity = Quantity - 1 WHERE
OrderSid=" . $_GET["sid"] . " AND ProductId=" . $_GET["pid"] . " AND
StoreId=" . $_GET["stid"] . " AND CategoryId=" . $_GET["cat"];

```

```

else
    $query = "DELETE FROM orders WHERE OrderSid=" .
$_GET["sid"] . " AND ProductId=" . $_GET["pid"] . " AND StoreId=" .
$_GET["stid"] . " AND CategoryId=" . $_GET["cat"];

    $result = mysqli_query($Connection,$query);
    if($result)
        echo 1;
    else
        echo 0;
}
else
    echo 0;
if($r)
    mysqli_free_result($r);

mysqli_close($Connection);
?>

```

Όπως έχει αναφερθεί, πάνω από τον πίνακα παραγγελιών υπάρχουν δύο κουμπιά. Το κουμπί με την ένδειξη **ΑΚΥΡΩΣΗ** εκτελεί όταν πατηθεί με την μέθοδο του **onclick** την συνάρτηση **CancelOrder** η οποία διαγράφει τον πίνακα παραγγελιών του χρήστη και επαναφέρει το δεξιό πλαίσιο (FRAME) στην αρχική του κατάσταση. Η διαγραφή της παραγγελίας επιτυγχάνεται με την κλήση του αρχείου **delorder.php**. Ο κώδικας του **delorder.php** είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$del = "DELETE FROM orders WHERE OrderSid=" . $_COOKIE["customer"] . """;

```

```
$result = mysqli_query($Connection,$del);
if($result)
{
    setcookie("customer","",time()-3600);
    echo 1;
}
else
    echo 0;
mysqli_close($Connection);
?>
```

Ο παραπάνω κώδικας διαγράφει τις εγγραφές του συγκεκριμένου χρήστη από τον πίνακα **orders** της βάσης χρησιμοποιώντας τον αριθμό που είναι αποθηκευμένος στον υπολογιστή του χρήστη με την τεχνική **Cookies** και εν συνεχεία διαγράφει και το ίδιο το **Cookie**.

Αν ο ίδιος χρήστης κάνει νέα παραγγελία θα δημιουργηθεί νέο **Cookie** και νέα αντίστοιχη καταχώριση στην βάση.

Το κουμπί με την ένδειξη **ΠΡΟΩΘΗΣΗ** εκτελεί όταν πατηθεί με την μέθοδο του **onclick** την συνάρτηση **ForwardOrder** η οποία χωρίς να διαγράψει τον πίνακα παραγγελιών επαναφέρει το δεξιό πλαίσιο (FRAME) στην αρχική του κατάσταση, και φορτώνει στο κεντρικό πλαίσιο (FRAME) την σελίδα **custform.php**.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ CUSTFORM.PHP

Η σελίδα **custform.php** προβάλλει μία φόρμα καταχώρισης στοιχείων πελάτη η οποία απαρτίζεται από οκτώ αντικείμενα εισαγωγής κειμένου και δύο κουμπιά.

Τα οκτώ αντικείμενα εισαγωγής κειμένου αντιστοιχούν στα εξής πεδία:

ΟΝΟΜΑ

ΕΠΩΝΥΜΟ

ΔΙΕΥΘΥΝΣΗ

ΠΟΛΗ

T.K.

ΧΩΡΑ

ΑΡ.ΚΑΡΤΑΣ

ΚΩΔ. ΚΑΡΤΑΣ

Τα δύο κουμπιά έχουν τις εξής ενδείξεις:

Το αριστερό την ένδειξη **ΑΠΟΣΤΟΛΗ ΠΑΡΑΓΓΕΛΙΑΣ**

Το δεξιό την ένδειξη **ΑΚΥΡΩΣΗ ΠΑΡΑΓΓΕΛΙΑΣ**

Το δεξιό με την μέθοδο **onclick** καλεί την συνάρτηση **CancelOrder** η λειτουργία της οποίας έχει περιγραφεί.

Το κουμπί με την ένδειξη **ΑΠΟΣΤΟΛΗ ΠΑΡΑΓΓΕΛΙΑΣ** με την μέθοδο **onclick** καλεί την συνάρτηση **SendOrder** ο κώδικας της οποίας βρίσκεται στο αρχείο **custform.js** και είναι:

```
function SendOrder()
{
    if(ValidateForm())
    {
        var url = "regcustomer.php"
        url += "?fn=" + urlencode(document.getElementById("fname").value)
        url += "&sn=" +
urlencode(document.getElementById("sname").value)
        url += "&add=" + urlencode(document.getElementById("add").value)
        url += "&city=" + urlencode(document.getElementById("city").value)
        url += "&zip=" + urlencode(document.getElementById("zip").value)
        url += "&cnt=" +
urlencode(document.getElementById("country").value)
        url += "&card=" +
urlencode(document.getElementById("card").value)
        url += "&pin=" + urlencode(document.getElementById("pin").value)
        url += "&sid=" + Math.random();
        xmlHttp = GetXmlHttp()
        if(xmlHttp != null)
        {
            try{
                xmlHttp.open("GET",url,false)
                xmlHttp.send(null)}
            catch(e) {}
        }
    }
}
```

```

        if (xmlHttp.readyState==4 && xmlHttp.status==200)
        {
            if(xmlHttp.responseText == 0)
                alert("Η ΚΑΤΑΧΩΡΗΣΗ ΔΕΝ ΕΓΙΝΕ")
            else
            {
                location.href = "empty.php"
            }
        }
    }
}

```

Ο παραπάνω κώδικας ξεκινάει καλώντας την συνάρτηση [Validate Form](#) ο κώδικας της οποίας βρίσκεται επίσης στο αρχείο **custform.js** και η λειτουργία του είναι η εξής:

Με τμήματα κώδικα **if{}** ελέγχει αν κάποιο από τα αντικείμενα κειμένου της φόρμας δεν έχει δεδομένα (κενό) ή αν κάποιο από τα αντικείμενα κειμένου που αντιπροσωπεύουν πεδία αριθμητικά περιέχουν κείμενο στα δεδομένα τους.

Αν υπάρχει κάποιο λάθος η συνάρτηση επιστρέφει **false** αλλιώς επιστρέφει **true** και εκτελείτε ο κώδικας της συνάρτησης [Validate Form](#).

Ο κώδικας αυτός καλεί το αρχείο **regcustomer.php** με παραμέτρους τα δεδομένα των αντικειμένων κειμένου της φόρμας.

Ο κώδικας του αρχείου **regcustomer.php** είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

if(!isset($_COOKIE["customer"]))
    echo 0;

```



```

else
{
    $query = "INSERT INTO customers VALUES('" . $_COOKIE["customer"] .
    ", '" . $_GET["fn"] . "', '" . $_GET["sn"] . "', '" . $_GET["add"] . "', '" . $_GET["city"]
    . "', '" . $_GET["zip"] . "', '" . $_GET["cnt"] . "', '" . $_GET["card"] . "', '" .
    $_GET["pin"] . "', " . time() . ")";
    $result = mysqli_query($Connection,$query);
    if($result)
    {
        setcookie("customer","",time()-3600);
        echo 1;
    }
    else
        echo 0;
}
mysqli_close($Connection);
?>

```

Ο κώδικας αυτός ελέγχει αν ο χρήστης έχει αποθηκευμένο στον υπολογιστή του το απαραίτητο **Cookie** και εν συνεχεία καταχωρεί τα δεδομένα στον πίνακα **customers** της βάσης.

Αφού γίνει η καταχώριση διαγράφεται το **Cookie** από τον υπολογιστή του.

Αν ο ίδιος πελάτης κάνει νέα παραγγελία θα δημιουργηθεί νέο **Cookie**.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ ADMIN.PHP

Ο κώδικας HTML της σελίδας **admin.php** δημιουργεί τρία πλαίσια.

```

<frameset rows="110,*" cols="*" frameborder="no" border="1" framespacing="0">
  <frame src="top.php" name="top" scrolling="No" noresize="noresize" id="top"
  title="top" />
  <frameset cols="200,*" frameborder="no" border="1" framespacing="0">
    <frame src="left.php" name="left" scrolling="No" noresize="noresize" id="left"
    title="left" />
    <frame src="main.php" name="main" id="main" title="main" scrolling="no"/>

```

```
</frameset>
</frameset>
```

Το πάνω οριζόντιο πλαίσιο φορτώνει την σελίδα «top.php».

Το αριστερό πλαίσιο φορτώνει την σελίδα «left.php».

Το κεντρικό πλαίσιο φορτώνει την σελίδα «main.php».

Η σελίδα «top.php» περιέχει τον κώδικα που παράγει τον τίτλο της ιστοσελίδας.

```
<h1 style="font-size:40px; text-align:center">ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΗΣ
```

Η σελίδα «main.php» είναι κενή και δίνει μόνο το χρώμα φόντου στο πλαίσιο.

```
<body style="background-color:#CCC">
```

Η σελίδα «left.php» περιέχει τον κώδικα που δημιουργεί μία μη ταξινομημένη λίστα **HTML** και ένα κουμπί κάτω από αυτή.

Το κουμπί αυτό έχει την ένδειξη **ΑΡΧΙΚΗ ΣΕΛΙΔΑ** και με την μέθοδο **onclick** εκτελεί τον κώδικα:

```
onclick="javascript:parent.location.href='index.php'"
```

Ο οποίος επαναφορτώνει και επιστρέφει τον χρήστη στην αρχική σελίδα του Site.

Ο κώδικας που δημιουργεί την λίστα είναι:

```
<h3 style="text-align:center">ΠΙΝΑΚΕΣ ΒΑΣΗΣ
<ul style="font-size:14px; text-align:left; line-height:24px">
<li onclick="stores()"><a>ΚΑΤΑΣΤΗΜΑΤΑ</a></li>
<li onclick="categories()"><a>ΚΑΤΗΓ. ΠΡΟΪΟΝΤΩΝ</a></li>
<li onclick="products()"><a>ΠΡΟΪΟΝΤΑ</a></li>
<li onclick="orders()"><a>ΠΑΡΑΓΓΕΛΙΕΣ</a></li>
<li onclick="password()"><a>ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ</a></li>
<li onclick="EmptyData()"><a>ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ</a></li>
<li onclick="CreateRandomData()"><a>ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ
ΔΕΔΟΜΕΝΩΝ</a></li>
</ul>
```

Τα στοιχεία της λίστας εκτός των δύο τελευταίων και του στοιχείου

ΠΑΡΑΓΓΕΛΙΕΣ φορτώνουν στο κεντρικό πλαίσιο (FRAME) αρχεία **PHP** τα οποία

δημιουργούν φόρμες καταχώρισης δεδομένων και κουμπιά ελέγχου, για τους αντίστοιχους πίνακες της βάσης.

Η κλήση των αρχείων γίνεται από κώδικα **javascript** ο οποίος καλείτε με τις μεθόδους των στοιχείων λίστας **onclick**.

```
<li onclick=...>
```

Οι συναρτήσεις **javascript** που καλούνται για κάθε στοιχείο λίστας είναι:

Για το στοιχείο λίστας με την ένδειξη ΚΑΤΑΣΤΗΜΑΤΑ είναι **stores**.

Για το στοιχείο λίστας με την ένδειξη ΚΑΤΗΓ. ΠΡΟΪΟΝΤΩΝ είναι **categories**.

Για το στοιχείο λίστας με την ένδειξη ΠΡΟΪΟΝΤΑ είναι **products**.

Για το στοιχείο λίστας με την ένδειξη ΠΑΡΑΓΓΕΛΙΕΣ είναι **orders**.

Για το στοιχείο λίστας με την ένδειξη ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ είναι **password**.

Για το στοιχείο λίστας με την ένδειξη ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ είναι **EmptyData**.

Για το στοιχείο λίστας με την ένδειξη ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ είναι **CreateRandomData**.

Ο κώδικας των συναρτήσεων αυτών βρίσκεται στο αρχείο **left.js** και είναι:

```
function stores()
{
    parent.main.location.href = "stores.php"
}

function categories()
{
    parent.main.location.href = "categories.php"
}

function products()
{
    parent.main.location.href = "products.php"
}

function password()
{
    parent.main.location.href = "password.php"
```

```

}

function orders()
{
    parent.main.location.href = "orders.php"
}

function EmptyData()
{
    parent.location.href = "emptydatabase.php"
}

function CreateRandomData()
{
    parent.location.href = "createdata.php"
}

```

Ο παραπάνω κώδικας απλά φορτώνει το κατάλληλο αρχείο **PHP** για τον αντίστοιχο κώδικα που θα προβληθεί στο κεντρικό πλαίσιο (FRAME).

Ο κώδικας και η τεχνική διαχείρισης των πινάκων της βάσης από την σελίδα διαχείρισης είναι σχεδόν όμοιος γι αυτό θα αναλυθεί μόνο το πρώτο κομμάτι κώδικα που δημιουργείτε από το αρχείο **stores.php**.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ STORES.PHP

Ο κώδικας του αρχείου αυτού δημιουργεί έναν πίνακα δύο στηλών και αντιπροσωπεύει τον πίνακα καταστημάτων (stores) της βάσης..

Τα πεδία των στηλών είναι ο κωδικός αριθμός του καταστήματος και το όνομα του καταστήματος.

Στον πίνακα αυτόν απεικονίζονται οι υπάρχουσες καταχωρίσεις καταστημάτων εάν υπάρχουν.

Κάτω από τον πίνακα υπάρχουν δύο αντικείμενα εισαγωγής κειμένου που είναι κρυφά και χρησιμεύουν για την εισαγωγή ή αλλαγή των δεδομένων του πίνακα.

Αυτά εμφανίζονται όταν ο χρήστης επιλέξει την δημιουργία νέας καταχώρισης ή την επεξεργασία μιας υπάρχουσας.

Στη συνέχεια υπάρχουν τέσσερα κουμπιά, το τελευταίο με την ένδειξη **ΔΙΑΓΡΑΦΗ** εμφανίζεται μόνο αν έχει επιλεγεί η επεξεργασία κάποιας υπάρχουσας εγγραφής.

Τα άλλα έχουν τις εξής ενδείξεις από αριστερά προς τα δεξιά.

ΔΗΜΙΟΥΡΓΙΑ

ΑΠΟΘΗΚΕΥΣΗ

ΑΚΥΡΩΣΗ

Ο κώδικας που αρχείου **stores.php** είναι:

```
<?php
require("check.php");
if(!$ok || !isset($_COOKIE[$_COOKIE["session"]]) ||
$_COOKIE[$_COOKIE["session"]] > 0)
{
    echo "<script type='text/javascript'>parent.location.href =
'index.php'</script>";
    exit();
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>main</title>
<link rel=stylesheet href="table.css" type="text/css">
<script type="text/javascript">var choice="stores"</script>
<SCRIPT LANGUAGE='JavaScript' SRC='common.js'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript' SRC='stores.js'></SCRIPT>
</head>
<body style="background-color:#CCC">
<?php
$host="localhost";
$user="admin";
$password="1234";
```

```

$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "SELECT * FROM stores ORDER BY StoreId";
$result = mysqli_query($Connection,$query);
echo "<div style='width:50%'>";
echo "<table border='1'>";
echo "<caption>ΚΑΤΑΣΤΗΜΑΤΑ</caption>";
echo "<tr>";
echo "<td class='aa'>" . "ΚΩΔ.ΚΑΤ." . "</td>";
echo "<td style='text-align:center'>" . "ΕΠΩΝΥΜΙΑ" . "</td>";
echo "</tr>";
echo "</table>";
echo "</div>";
if($result && mysqli_num_rows($result))
{
    echo "<div id='scroll' class='MyDiv'>";
    echo "<table border='1'>";
    for($i=0;$i<mysqli_num_rows($result);$i++)
    {
        echo "<tr>";
        $arr = mysqli_fetch_row($result);
        for($k=0;$k<mysqli_num_fields($result);$k++)
        {
            if($k==0)
                echo "<td class='aa' id='' . $arr[$k] . '' onclick='Edit(" .
                $arr[$k] . ")' style='cursor:pointer;text-decoration:underline'>";
            else
                echo "<td id='f' . $arr[0] . ''>";
            echo $arr[$k];
            echo "</td>";
        }
        echo "</tr>";
    }
}

```

```

    }
    echo "</table>";
    echo "</div>";
    echo "<script>document.getElementById('scroll').scrollTop=10900</script>";
}

mysqli_free_result($result);
mysqli_close($Connection);
?>
<br>
<div id="entry" style="visibility:hidden">
<input type="text" id="storeid" onkeydown="StoreKey(event)" size="10"/>
<input type="text" id="firm" onkeydown="FirmKey(event)" size="50"/>
<br>
</div>
<input type="button" id="new" size="10" value="ΔΗΜΙΟΥΡΓΙΑ "
onclick="New()"/>
<input type="button" id="save" size="10" value="ΑΠΟΘΗΚΕΥΣΗ"
onclick="Save()"/>
<input type="button" id="cancel" size="10" value="ΑΚΥΡΩΣΗ"
onclick="Cancel()"/>
<input type="button" id="delete" size="10" value="ΔΙΑΓΡΑΦΗ" onclick="Delete()"
style="visibility:hidden"/>
</body>
</html>

```

Στην αρχή του κώδικα απαιτείτε η ύπαρξη και η εκτέλεση του αρχείου **check.php** καθώς και η ύπαρξη του **COOKIE["session"]** που έχει δημιουργηθεί στον υπολογιστή του χρήστη και έχει γίνει και η αντίστοιχη καταχώριση στην βάση στον πίνακα **sessions**, όταν ο χρήστης εισήγαγε τον κωδικό πρόσβασης του στη σελίδα διαχείρισης.

Ο κώδικας του αρχείου **check.php** ελέγχει αν υπάρχει στον υπολογιστή του χρήστη το απαραίτητο **Cookie** και αν υπάρχει ότι αυτό δεν έχει λήξει. Αν υπάρχει και έχει λήξη τότε το διαγράφει από τον υπολογιστή του χρήστη.

Αν δεν πληρούνται οι προϋποθέσεις πρόσβασης ο χρήστης ανακατευθύνεται στην αρχική σελίδα του Site.

Στον πίνακα που δημιουργείτε, τα πεδία της πρώτης στήλης είναι υπογραμμισμένα ώστε να καταλάβει ο χρήστης ότι παρέχεται κάποια δυνατότητα επεξεργασίας.

Όπως και στους πίνακες των προηγούμενων ενοτήτων έτσι και εδώ επιλέγοντας το πρώτο πεδίο κάποιας εγγραφής και εμφανίζονται τα κρυφά πλαίσια εισαγωγής κειμένου με φορτωμένες τις τιμές των πεδίων της εγγραφής που επιλέχθηκε.

Ο συνάρτηση που εκτελεί αυτήν την ενέργεια είναι η **Edit** που καλείτε με την μέθοδο **onclick** του αντικειμένου κυψέλης πίνακα (<td onclick=...>).

Η συνάρτηση αυτή παίρνει μία παράμετρο που είναι ο κωδικός καταστήματος.

Ο κώδικας της συνάρτησης βρίσκεται στο αρχείο **stores.js** και είναι:

```
function Edit(i)
{
    edit = true
    document.getElementById("entry").style.visibility="visible"
    document.getElementById("delete").style.visibility="visible"
    document.getElementById("storeid").value =
document.getElementById(i).innerHTML
    document.getElementById("firm").value = document.getElementById("f" +
i).innerHTML
    if(choice=="prod")
    {
        document.getElementById("category").value =
document.getElementById("c" + i).innerHTML
        document.getElementById("stores").value =
document.getElementById("s" + i).innerHTML
        document.getElementById("price").value =
document.getElementById("p" + i).innerHTML
    }
    document.getElementById("storeid").select()
    oldvalue = document.getElementById(i).innerHTML
}
```

Στην συνέχεια μπορούμε να επιλέξουμε κάποια ενέργεια από τα κουμπιά.

Μπορούμε να επιλέξουμε το κουμπί **ΑΚΥΡΩΣΗ** το οποίο με την μέθοδο του **onclick** εκτελεί την συνάρτηση **Cancel**.

Ο κώδικας της συνάρτησης αυτής απλά αδειάζει τις τιμές των πεδίων εισαγωγής κειμένου και τα ξανά κάνει κρυφά .

Αν επιλέξουμε το κουμπί **ΔΙΑΓΡΑΦΗ**, με την μέθοδο του **onclick** εκτελείτε η συνάρτηση **Delete** ο κώδικας της οποίας βρίσκεται στο αρχείο **stores.js** και είναι:

```
function Delete()
{
    switch(choice)
    {
        case "stores":
            var url="delstor.php"
            break;
        case "cat":
            var url="delcat.php"
            break;
        case "prod":
            var url="delprod.php"
            break;
    }
    url += "?pid=" + urlencode(oldvalue)
    url += "&sid=" +Math.random();
    if(choice == "prod")
    {
        url += "&ost=" + urlencode(oldstores)
    }
    xmlHttp = GetXmlHttp()
    if(xmlHttp != null)
    {
        try{
            xmlHttp.open("GET",url,false)
            xmlHttp.send(null)}
    }
```

```

        catch(e) {}
        if (xmlHttp.readyState==4 && xmlHttp.status==200)
        {
//            alert(xmlHttp.responseText)
            temp = xmlHttp.responseText.split("@")
            if(temp[0] == "0")
                alert(temp[1])
            else if(temp[0] == "1451")
                alert("Η ΔΙΑΓΡΑΦΗ ΑΠΕΤΥΧΕ\nΥΠΑΡΧΕΙ
ΣΧΕΤΙΚΗ ΚΑΤΑΧΩΡΙΣΗ ΣΤΑ ΠΡΟΪΟΝΤΑ\nΔΙΑΓΡΑΨΤΕ ΠΡΩΤΑ ΤΟ
ΠΡΟΪΟΝ")
                else
                {
                    window.location.reload()
                    Cancel()
                }
            }
        }
    }
}

```

Ο κώδικας είναι κοινός για όλους τους πίνακες που διαχειρίζεται ο χρήστης. Ανάλογα με το ποιόν πίνακα επεξεργάζεται καλείτε και το αντίστοιχο αρχείο διαγραφής.

Ο κώδικας διαγραφής των αρχείων αυτών είναι σχεδόν ίδιος οπότε παραθέεται ως παράδειγμα ο κώδικας που περιέχεται στο αρχείο διαγραφής καταστημάτων

delstor.php και είναι:

```

<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

```

```

$query = "DELETE FROM stores WHERE StoreId=" . $_GET["pid"] . """;
$result = mysqli_query($Connection,$query);
if(!$result)
    if(mysqli_errno($Connection)==1451)
        echo 1451 . "@";
    else
        echo 0 . "@" . mysqli_error($Connection);
else
    echo 1 . "@";
mysqli_close($Connection);
?>

```

Ο κώδικας αυτός διαγράφει συγκεκριμένη εγγραφή από συγκεκριμένο πίνακα της βάσης.

Οι συναρτήσεις **mysqli_errno(\$Connection)** και **mysqli_error(\$Connection)** μας δίνουν τον κωδικό λάθους και την περιγραφή του αντίστοιχα, που παράγει η **MySQL** σε περίπτωση αποτυχίας εκτέλεσης κάποιου ερωτήματος.

Αν επιλέξουμε το κουμπί **ΔΗΜΙΟΥΡΓΙΑ**, με την μέθοδο του **onclick** εκτελείτε η συνάρτηση **New** ο κώδικας της οποίας απλά εμφανίζει τα κρυφά αντικείμενα εισαγωγής κειμένου χωρίς δεδομένα (κενά), προς καταχώριση νέων δεδομένων.

Αν επιλέξουμε το κουμπί **ΑΠΟΘΗΚΕΥΣΗ**, με την μέθοδο του **onclick** εκτελείτε η συνάρτηση **Save** ο κώδικας της οποίας είναι:

```

function Save()
{
    if(document.getElementById("entry").style.visibility=="visible")
    {
        if(choice=="prod")
        {
            if(isNaN(document.getElementById("storeid").value) ||
Number(document.getElementById("storeid").value) < 1)
                document.getElementById("storeid").select()
            else if(isNaN(document.getElementById("category").value) ||
Number(document.getElementById("category").value) < 1)

```

```

        document.getElementById("category").select()
        else if(isNaN(document.getElementById("stores").value) ||
Number(document.getElementById("stores").value) < 1)
            document.getElementById("stores").select()
        else
        {
            if(!edit)
                SendData()
            else
                UpdateData()
            edit = false
        }
    }
else
{
    if(isNaN(document.getElementById("storeid").value) ||
Number(document.getElementById("storeid").value) < 1)
        document.getElementById("storeid").select()
    else
    {
        if(!edit)
            SendData()
        else
            UpdateData()
        edit = false
    }
}
}
}
}
}

```

Ο παραπάνω κώδικας ανάλογα με τον αν η εγγραφή προς αποθήκευση είναι κάποια υπάρχουσα που τροποποιήθηκε ή νέα, καλεί το αντίστοιχο αρχείο **PHP**.

Στην πρώτη περίπτωση καλείτε το αρχείο **UpdateData** ενώ στην δεύτερη το αρχείο **SendData**.

Ο κώδικας του αρχείου **UpdateData** είναι:

```
function UpdateData()
{
    exit = false
    switch(choice)
    {
        case "stores":
            if(document.getElementById("storeid").value &&
document.getElementById("firm").value)
                var url="upstor.php"
            else
                exit = true
            break;
        case "cat":
            if(document.getElementById("storeid").value &&
document.getElementById("firm").value)
                var url="upcat.php"
            else
                exit = true
            break;
        case "prod":
            if(document.getElementById("storeid").value &&
document.getElementById("firm").value &&
document.getElementById("price").value)
                var url="upprod.php"
            else
                exit = true
            break;
    }
    if(!exit)
    {
        url += "?pid=" +
urlencode(document.getElementById("storeid").value)
```

```

url += "&f=" + encodeURIComponent(document.getElementById("firm").value)
url += "&oid=" + encodeURIComponent(oldvalue)
if(choice=="prod")
{
    url += "&c=" +
encodeURIComponent(document.getElementById("category").value)
    url += "&s=" +
encodeURIComponent(document.getElementById("stores").value)
    url += "&p=" +
encodeURIComponent(document.getElementById("price").value)
    url += "&ost=" + encodeURIComponent(oldstores)
    url += "&ocat=" + encodeURIComponent(oldcat)
}
url += "&sid=" + Math.random();
xmlHttp = GetXmlHttp()
if(xmlHttp != null)
{
    try{
        xmlHttp.open("GET",url,false)
        xmlHttp.send(null)}
    catch(e) {}
    if (xmlHttp.readyState==4 && xmlHttp.status==200)
    {
        temp = xmlHttp.responseText.split("@")
        if(temp[0] == 0)
            alert(temp[1])
        else
            window.location.reload()
        Cancel()
    }
}
}
}

```

Επειδή ο κώδικας είναι κοινός για όλους τους πίνακες της βάσης, ανάλογα με τον ποιόν πίνακα δουλεύουμε καλείτε και το αντίστοιχο αρχείο **PHP** που τροποποιεί εγγραφές στον αντίστοιχο πίνακα.

Για κάθε πίνακα το αντίστοιχο αρχείο είναι:

ΚΑΤΑΣΤΗΜΑΤΑ	upstor.php
ΚΑΤΗΓΟΡΙΕΣ	upcat.php
ΠΡΟΪΟΝΤΑ	upprod.php
ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ	updatepass.php

Ενδεικτικά παραθέεται ο κώδικας του αρχείου **upstor.php**

```
<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "UPDATE stores SET StoreId=" . $_GET["pid"] . ", Firm=LEFT(" .
$_GET["f"] . ",45) WHERE StoreId=" . $_GET["oid"];
$result = mysqli_query($Connection,$query);
if(!$result)
    echo 0 . "@" . mysqli_error($Connection);
else
    echo 1 . "@";
mysqli_close($Connection);
?>
```

Στην δεύτερη περίπτωση καλείτε το αρχείο **SendData**.

Ο κώδικας του αρχείου **SendData** είναι:

```
function SendData()
{
    exit = false
    switch(choice)
```

```

{
    case "stores":
        if(document.getElementById("storeid").value &&
document.getElementById("firm").value)
            var url="regstor.php"
        else
            exit = true
    break;
    case "cat":
        if(document.getElementById("storeid").value &&
document.getElementById("firm").value)
            var url="regcat.php"
        else
            exit = true
    break;
    case "prod":
        if(document.getElementById("storeid").value &&
document.getElementById("firm").value &&
document.getElementById("price").value)
            var url="regprod.php"
        else
            exit = true
    break;
}
if(!exit)
{
    url += "?pid=" +
urlencode(document.getElementById("storeid").value)
    url += "&f=" + urlencode(document.getElementById("firm").value)
    if(choice=="prod")
    {
        url += "&c=" +
urlencode(document.getElementById("category").value)

```



```

        url += "&s=" +
urlencode(document.getElementById("stores").value)
        url += "&p=" +
urlencode(document.getElementById("price").value)
    }
    url += "&sid=" +Math.random();
    xmlhttp = GetXmlHttp()
    if(xmlHttp != null)
    {
        try{
            xmlhttp.open("GET",url,false)
            xmlhttp.send(null)}
        catch(e) {}
        if (xmlHttp.readyState==4 && xmlhttp.status==200)
        {
            temp = xmlhttp.responseText.split("@")
            if(temp[0] == 0)
                alert(temp[1])
            else
                window.location.reload()
            Cancel()
        }
    }
}
}
}

```

Και εδώ πάλι ανάλογα με τον ποιόν πίνακα επεξεργαζόμαστε καλείτε το αντίστοιχο αρχείο **PHP** που καταχωρεί νέες εγγραφές στον αντίστοιχο πίνακα στην βάση.

Για κάθε πίνακα το αντίστοιχο αρχείο είναι:

ΚΑΤΑΣΤΗΜΑΤΑ	regstor.php
ΚΑΤΗΓΟΡΙΕΣ	regcat.php
ΠΡΟΪΟΝΤΑ	regprod.php
ΚΩΔ. ΔΙΑΧΕΙΡΙΣΗΣ	newpass.php

Ενδεικτικά παραθέτεται ο κώδικας του αρχείου **regstor.php**

```
<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "INSERT INTO stores VALUES('" . $_GET["pid"] . "',LEFT('" .
$_GET["f"] . "',45))";
$result = mysqli_query($Connection,$query);
if(!$result)
    echo 0 . "@" . mysqli_error($Connection);
else
    echo 1 . "@";
mysqli_close($Connection);
?>
```

Οι παραπάνω κώδικες είναι κοινοί όπως έχει λεχθεί, υπάρχουν μόνο κάποιες μικρές διαφοροποιήσεις στον πίνακα **ΠΡΟΪΟΝΤΑ** λόγω ότι έχει περισσότερες στήλες, οπότε και περισσότερα αντικείμενα εισαγωγής κειμένου για την τροποποίηση και την δημιουργία νέων εγγραφών.

Συγκεκριμένα έχει τρεις στήλες περισσότερες που είναι η κατηγορία που ανήκει το προϊόν, το κατάστημα στο οποίο βρίσκεται καθώς και η τιμή του.

Στα κρυφά αντικείμενα εισαγωγής κειμένου που εμφανίζονται τα δύο από αυτά είναι για την ακρίβεια πτυσσόμενες λίστες.

Αυτά είναι ένα για την επιλογή της κατηγορίας που ανήκει το προϊόν και το άλλο για το κατάστημα που βρίσκεται.

Απαραίτητη προϋπόθεση λοιπόν για την καταχώριση νέου προϊόντος είναι να υπάρχει τουλάχιστον μία κατηγορία καταχωρημένη και ένα κατάστημα.

Λίγο διαφορετική είναι και η συνάρτηση που δίνει την δυνατότητα τροποποίησης υπάρχουσας εγγραφής. Στους άλλους πίνακες είναι η συνάρτηση **Edit** ενώ εδώ η συνάρτηση **EditProd**.

Αρχική διαφορά είναι ότι παίρνει δύο παραμέτρους περισσότερες οι οποίες είναι ο κωδικός του προϊόντος, ο κωδικός του καταστήματος και ο κωδικός της κατηγορίας που ανήκει το προϊόν.

Ο κώδικας της συνάρτησης **EditProd** βρίσκεται στο αρχείο **stores.js** και είναι:

```
function EditProd(i,k,l)
{
    edit = true
    var array = GetRecord(i,k,l)
    document.getElementById("entry").style.visibility="visible"
    document.getElementById("delete").style.visibility="visible"
    document.getElementById("storeid").value = array[0]
    document.getElementById("firm").value = array[1]
    document.getElementById("category").value = array[2]
    document.getElementById("stores").value = array[3]
    document.getElementById("price").value = array[4]
    document.getElementById("storeid").select()
    oldvalue = array[0]
    oldstores = array[3]
    oldcat = array[2]
    document.getElementById("new").value = "ΠΡΟΣΘΗΚΗ"
}
```

Ο παραπάνω κώδικας καλεί στην αρχή την συνάρτηση **GetRecord** η οποία επιστρέφει ως αποτέλεσμα έναν πίνακα με τις τιμές του οποίου γεμίζουν τα αντίστοιχα αντικείμενα δεδομένων της φόρμας.

Η συνάρτηση **GetRecord** καλεί το αρχείο **getrec.php** με τις παραμέτρους τις και το αποτέλεσμα είναι η επιστροφή συγκεκριμένης εγγραφής από τον πίνακα προϊόντα με βάση τον κωδικό προϊόντος, τον κωδικό καταστήματος και τον κωδικό κατηγορίας του προϊόντος.

Ο κώδικας του αρχείου **getrec.php** είναι:

```
<?php
$host="localhost";
$user="admin";
```

```

$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$query = "SELECT * FROM products WHERE ProductId=" . $_GET["pid"] . " AND
StoreId=" . $_GET["stid"] . " AND CategoryId=" . $_GET["cat"];
$result = mysqli_query($Connection,$query);
if($result && mysqli_num_rows($result))
{
    for($i=0;$i<mysqli_num_rows($result);$i++)
    {
        $array = mysqli_fetch_row($result);
        for($k=0; $k < mysqli_num_fields($result); $k++)
        {
            echo $array[$k];
            if(($k < mysqli_num_fields($result)-1) || $i <
mysqli_num_rows($result)-1)
                echo "@";
        }
    }
}
else
    echo 0;
if($result)
    mysqli_free_result($result);
mysqli_close($Connection);
?>

```

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ ORDERS.PHP

Το αρχείο αυτό δημιουργεί δύο πίνακες και τέσσερα κουμπιά.

Ο πρώτος πίνακας απεικονίζει τα στοιχεία του πελάτη της τρέχουσας παραγγελίας ενώ ο δεύτερος τις εγγραφές των προϊόντων της παραγγελίας του πελάτη.

Ανάλογα με τα δικαιώματα πρόσβασης μπορεί κάποιος να δει και τις αντίστοιχες εγγραφές παραγγελιών.

Για παράδειγμα ο διαχειριστής του Site μπορεί να δει όλες τις εγγραφές ενώ ο διαχειριστής συγκεκριμένου καταστήματος (χρήστης) μόνο τις εγγραφές που αφορούν προϊόντα του καταστήματος του εφ' όσων υπάρχουν.

Ο κώδικας του αρχείου **orders.php** είναι:

```
<?php
require("check.php");
if(!$ok || !isset($_COOKIE[$_COOKIE["session"]]))
{
    echo "<script type='text/javascript'>parent.location.href =
'index.php'</script>";
    exit();
}
else
    $stid = $_COOKIE[$_COOKIE["session"]];
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>main</title>
<SCRIPT LANGUAGE='JavaScript' SRC='common.js'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript' SRC='orders.js'></SCRIPT>
<style type="text/css">
table {
    background-color:#FFF;
    border-collapse:collapse;
}
div.MyDiv {
    overflow:auto;
    height:277px;
```

```

        width:880px;
    }
</style>
</head>
<body style="background-color:#CCC">
<?php
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$total = 0;
if(!$_GET["start"])
    $start = 0;
else
    $start = $_GET["start"] - 1;

if($stid == 0)
{
    $query = "SELECT * FROM customers ORDER BY Stamp ASC LIMIT " .
    $start . ",1";
    $ask = "SELECT Count(*) FROM customers";
}
else
{
    $query = "SELECT DISTINCTROW c.* FROM customers c, orders o where
c.ordersid=o.ordersid AND o.`StoreId`='" . $stid . "' ORDER BY Stamp LIMIT " .
    $start . ",1";
    $ask = "SELECT DISTINCTROW c.OrderSid FROM customers c, orders o
where c.ordersid=o.ordersid AND o.`StoreId`='" . $stid . "' ORDER BY Stamp";
}
}

```

```

$result = mysqli_query($Connection,$query);
if($result && mysqli_num_rows($result))
{
    $arr = mysqli_fetch_row($result);
    $sid = $arr[0];
    if($stid==0)
    {
        echo "<input type='button' id='execute' style='float:right' size='10'
value='ΕΚΤΕΛΕΣΗ' onclick='Execute()'/>";
        echo "<input type='button' id='delete' style='float:right' size='10'
value='ΔΙΑΓΡΑΦΗ' onclick='Delete()'/>";
    }

    echo "<table border='1'>";
    echo "<caption style='text-align:left;font-weight:bold'>ΣΤΟΙΧΕΙΑ
ΠΕΛΑΤΗ</caption>";
    echo "<tr>";
    echo "<td style='width:100px;background-
color:#CCC'>ΟΝΟΜΑ:&nbsp;  </td>";
    echo "<td style='width:200px;'>" . $arr[1] . "</td>";
    echo "<td style='width:100px;background-
color:#CCC'>ΕΠΩΝΥΜΟ:&nbsp;  </td>";
    echo "<td style='width:200px;'>" . $arr[2] . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td style='width:100px;background-
color:#CCC'>ΔΙΕΥΘΥΝΣΗ:&nbsp;  </td>";
    echo "<td style='width:200px;'>" . $arr[3] . "</td>";
    echo "<td style='width:100px;background-
color:#CCC'>ΠΟΛΗ:&nbsp;  </td>";
    echo "<td style='width:200px;'>" . $arr[4] . "</td>";
    echo "</tr>";
    echo "<tr>";

```

```

echo "<td style='width:100px;background-color:#CCC'>T.K.:&nbsp;</td>";
echo "<td style='width:200px;'>" . $arr[5] . "</td>";
echo "<td style='width:100px;background-color:#CCC'>XΩPA:&nbsp;</td>";
echo "<td style='width:200px;'>" . $arr[6] . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td style='width:100px;background-
color:#CCC'>ΑΡ.ΚΑΡΤΑΣ:&nbsp;</td>";
echo "<td style='width:200px;'>" . $arr[7] . "</td>";
echo "<td style='width:100px;background-
color:#CCC'>ΚΩΔ.ΚΑΡΤΑΣ:&nbsp;</td>";
echo "<td style='width:200px;'>" . $arr[8] . "</td>";
echo "</tr>";
echo "</table>";

if($stid == 0)
{
    $query2 = "SELECT DISTINCTROW s.Firm, c.Name, p.`Product`,
o.`Quantity`, o.`Price` FROM stores s, orders o LEFT OUTER JOIN categories c ON
c.CategoryId=o.CategoryId LEFT OUTER JOIN products p ON
p.CategoryId=c.CategoryId WHERE s.`StoreId`=o.`StoreId` AND
p.ProductId=o.ProductId AND o.OrderSid=" . $sid . " ORDER BY o.Stamp ASC";
}
else
{
    $query2 = "SELECT DISTINCTROW s.Firm, c.Name, p.`Product`,
o.`Quantity`, o.`Price` FROM stores s, orders o LEFT OUTER JOIN categories c ON
c.CategoryId=o.CategoryId LEFT OUTER JOIN products p ON
p.CategoryId=c.CategoryId WHERE s.`StoreId`=o.`StoreId` AND
p.ProductId=o.ProductId AND o.OrderSid=" . $sid . " AND o.`StoreId`=" . $stid . ""
ORDER BY o.Stamp ASC";
}
$res = mysqli_query($Connection,$query2);

```



```

if($res && mysqli_num_rows($res))
{
    echo "<table border='1'>";
    echo "<caption style='text-align:left;font-weight:bold'>ΣΤΟΙΧΕΙΑ
ΠΑΡΑΓΓΕΛΙΑΣ</caption>";
    echo "<tr>";
    echo "<td style='width:200px;text-align:center'>ΚΑΤΑΣΤΗΜΑ</td>";
    echo "<td style='width:200px;text-align:center'>ΚΑΤΗΓΟΡΙΑ</td>";
    echo "<td style='width:200px;text-align:center'>ΠΡΟΪΟΝ</td>";
    echo "<td style='width:80px;text-align:center'>ΤΕΜ.</td>";
    echo "<td style='width:80px;text-align:center'>ΤΙΜΗ</td>";
    echo "<td style='width:80px;text-align:center'>ΑΞΙΑ</td>";
    echo "</tr>";
    echo "</table>";
    echo "<div id='scroll' class='MyDiv'>";
    echo "<table border='1'>";
    for($i=0;$i<mysqli_num_rows($res);$i++)
    {
        $arr = mysqli_fetch_row($res);
        echo "<tr>";
        for($k=0;$k<mysqli_num_fields($res);$k++)
        {
            if($k<3)
                echo "<td style='width:200px'>" . $arr[$k] .
"</td>";
            else
                echo "<td style='width:80px;text-align:center'>"
. $arr[$k] . "</td>";
            if($k==mysqli_num_fields($res)-1)
            {
                echo "<td style='width:80px;text-align:center'>"
. ($arr[$k-1] * $arr[$k]) . "</td>";
                $total += ($arr[$k-1] * $arr[$k]);
            }
        }
    }
}

```

```

    }
}
echo "</tr>";
}
if(mysql_num_rows($res)<12)
{
    echo "<tfoot><td colspan='5' style='font-
weight:bold>ΣΥΝΟΛΙΚΗ ΑΞΙΑ</td><td style='text-align:center;font-weight:bold>"
. $total . "</td></tfoot>";
    echo "</table>";
    echo "</div>";
}
else
{
    echo "</table>";
    echo "</div>";
    echo "<table>";
    echo "<tr>";
    echo "<td style='width:200px;font-weight:bold;border-
right:none;border-left:thin;border-left-style:solid;border-bottom:thin;border-bottom-
style:solid>ΣΥΝΟΛΙΚΗ ΑΞΙΑ</td>";
        echo "<td style='width:200px;text-align:center;border-
right:none;border-bottom:thin;border-bottom-style:solid></td>";
        echo "<td style='width:200px;text-align:center;border-
right:none;border-bottom:thin;border-bottom-style:solid></td>";
        echo "<td style='width:80px;text-align:center;border-
right:none;border-bottom:thin;border-bottom-style:solid></td>";
        echo "<td style='font-weight:bold;width:80px;text-
align:center;border-left:thin;border-left-style:solid;border-right:thin;border-right-
style:solid;border-bottom:thin;border-bottom-style:solid>" . $total . "</td>";
    echo "</tr>";
}
}
}

```

```

        echo "</table>";
    }
    echo
"<script>document.getElementById('scroll').scrollTop=10900</script>";
    }
}
else
    echo "ΔΕΝ ΥΠΑΡΧΟΥΝ ΠΑΡΑΓΓΕΛΙΕΣ";

if($res)
    mysqli_free_result($res);
if($result && mysqli_num_rows($result))
{
    $askres = mysqli_query($Connection,$ask);
    if($askres && mysqli_num_rows($askres))
    {
        if($stid==0)
        {
            $temp = mysqli_fetch_row($askres);
            $totalrecords = $temp[0];
        }
        else
            $totalrecords = mysqli_num_rows($askres);
    }
    if($askres)
        mysqli_free_result($askres);
    echo "<br>";
    echo "<input type='button' id='back' size='10' value='ΠΡΟΗΓΟΥΜΕΝΗ'
onclick='Back()'/>";
    echo "<input type='text' style='text-align:center' value='' . ($start + 1) . " ΑΠΟ
" . $totalrecords . "' readonly size='10'>";
    echo "<input type='button' id='forth' size='15' value='ΕΠΙΟΜΕΝΗ'
onclick='Forth()'/>";

```

```

        echo "<script type='text/javascript'>var Start=" . ($start+1) . ";var
Totalrecords=" . $totalrecords . ";var SessionId=" . $sid . "</script>";
    }
    if($result)
        mysqli_free_result($result);
    mysqli_close($Connection);
?>
</body>
</html>

```

Κάτω από τους πίνακες αυτούς υπάρχουν δύο κουμπιά το αριστερό έχει την ένδειξη **ΠΡΟΗΓΟΥΜΕΝΗ** και το δεξιό την ένδειξη **ΕΠΟΜΕΝΗ**.

Με την μέθοδο **onclick** τους καλούν τις αντίστοιχες συναρτήσεις που βοηθούν τον χρήστη να πλοηγηθεί στο αρχείο παραγγελιών προς τα μπροστά και προς τα πίσω.

Η συνάρτηση που καλείτε από το κουμπί **ΠΡΟΗΓΟΥΜΕΝΗ** είναι η **Back** και η συνάρτηση του κουμπιού **ΕΠΟΜΕΝΗ** είναι **Forth**.

Επειδή ο κώδικας τους είναι ίδιος, παραθέτω τον κώδικα της συνάρτησης **Back** ο οποίος βρίσκεται στο αρχείο **orders.js** και είναι:

```

function Back()
{
    if(Start > 1)
    {
        url = "orders.php?start=" + (Number(Start) - 1)
        url += "&sid=" + Math.random()
        location.href = url
    }
}

```

Ο παραπάνω κώδικας απλά ξαναφορτώνει την σελίδα **orders.php** με την παράμετρο **start** που παίρνει, να έχει για τιμή την θέση της συγκεκριμένης εγγραφής στον πίνακα παραγγελιών.

Ανάμεσα στα δύο αυτά κουμπιά υπάρχει ένα αντικείμενο εισαγωγής κειμένου το οποίο είναι μόνο για ανάγνωση και απεικονίζει τον εκάστοτε αριθμό παραγγελίας και το σύνολο των παραγγελιών.

Υπάρχουν επίσης και δύο κουμπιά στην πάνω δεξιά γωνία της ιστοσελίδας τα οποία έχουν τις εξής ενδείξεις:

Το αριστερό την ένδειξη **ΔΙΑΓΡΑΦΗ** και το δεξιό την ένδειξη **ΕΚΤΕΛΕΣΗ**.

Το κουμπί με την ένδειξη **ΔΙΑΓΡΑΦΗ** εκτελεί με την μέθοδο του **onclick** την συνάρτηση **Delete** ενώ το κουμπί **ΕΚΤΕΛΕΣΗ** την συνάρτηση **Execute**.

Στην πράξη και τα δύο κουμπιά εκτελούν τον ίδιο κώδικα αφού ο κώδικας της συνάρτησης **Execute** είναι η κλήση της συνάρτησης **Delete**.

Στην πράξη είτε εκτελεστεί η παραγγελία είτε διαγραφεί και στις δύο περιπτώσεις πρέπει να διαγραφεί από το αρχείο παραγγελιών.

Ο κώδικας που διαγράφει εγγραφές από κάποιον πίνακα της βάσης, όπως και στην παρούσα περίπτωση, είναι σχεδόν ίδιος με προηγούμενους που έχουν περιγραφεί.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ EMPTYDATABASE.PHP

Αν επιλεγεί η διαγραφή των δεδομένων της βάσης από την λίστα του αριστερού πλαισίου (FRAME) τότε εκτελείτε ο κώδικας του αρχείου **emptydatabase.php**.

Ο κώδικας του αρχείου είναι:

```
<?PHP
require("check.php");
if(!$ok || !isset($_COOKIE[$_COOKIE["session"]]) ||
$_COOKIE[$_COOKIE["session"]] > 0)
{
    echo "<script type='text/javascript'>parent.location.href =
'index.php'</script>";
    exit();
}
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");
```

```

$del1 = "DELETE FROM customers";
$del2 = "DELETE FROM orders";
$del3 = "DELETE FROM passlist";
$del4 = "DELETE FROM products";
$del5 = "DELETE FROM categories";
$del6 = "DELETE FROM stores";
$del7 = "DELETE FROM sessions";
$res1 = mysqli_query($Connection,$del1);
$res2 = mysqli_query($Connection,$del2);
$res3 = mysqli_query($Connection,$del3);
$res4 = mysqli_query($Connection,$del4);
$res5 = mysqli_query($Connection,$del5);
$res6 = mysqli_query($Connection,$del6);
$res7 = mysqli_query($Connection,$del7);

$insert = "INSERT INTO passlist VALUES('0', 'ΔΙΑΧΕΙΡΗΣΤΗΣ' , 0)";
$result = mysqli_query($Connection,$insert);

setcookie($_COOKIE[$_COOKIE["session"]], "", time()-3600);
setcookie($_COOKIE["session"], "", time()-3600);
setcookie($_COOKIE["customer"], "", time()-3600);

mysqli_close($Connection);
header("Location: index.php");
?>

```

Ο παραπάνω κώδικας απλά εκτελεί συνεχόμενα πολλά απλά ερωτήματα διαγραφής εγγραφών σε όλους του πίνακες της βάσης.

Το αποτέλεσμα είναι μια βάση εντελώς κενή.

ΚΩΔΙΚΑΣ ΣΕΛΙΔΑΣ CREATEDATA.PHP

Ο κώδικας του αρχείου αυτού δημιουργεί έναν τυχαίο αριθμό δεδομένων σε κάθε πίνακα της βάσης για λόγους ευκολίας στην επίδειξη της εφαρμογής.

Ο κώδικας του αρχείου **createdata.php** είναι:

```
<?PHP
require("check.php");
if(!$ok || !isset($_COOKIE[$_COOKIE["session"]]) ||
$_COOKIE[$_COOKIE["session"]] > 0)
{
    echo "<script type='text/javascript'>parent.location.href =
'index.php'</script>";
    exit();
}
$host="localhost";
$user="admin";
$password="1234";
$Connection = mysqli_connect($host,$user,$password,"stores");
mysqli_set_charset ($Connection, "utf8");

$check = "SELECT StoreId FROM stores";
$checkres = mysqli_query($Connection,$check);
if($checkres && mysqli_num_rows($checkres))
{
    if($checkres)
        mysqli_free_result($checkres);
    header("Location: index.php");
    exit();
}

$error = false;
$error_val = "";

$MaxStores = mt_rand(4, 8);
```

```

$MaxCategories = mt_rand(3, 6);
$MaxProducts = mt_rand(5, 10);
$MaxOrders = mt_rand(5, 10);
$MaxCustomers = $MaxOrders;
$MaxUsers = $MaxStores;

for($i=1;$i<$MaxStores + 1;$i++)
{
    $insert1 = "INSERT INTO stores VALUES(" . $i . ", 'KATAΣΤΗΜΑ " . $i .
    ")";
    $resins1 = mysqli_query($Connection,$insert1);
    if(!$resins1)
    {
        $error = true;
        $error_val = "insert1" . "<br>" . $i . "<br>" . $insert1;
    }
}
if(!$error)
{
    for($i=1;$i<$MaxCategories + 1;$i++)
    {
        $insert2 = "INSERT INTO categories VALUES(" . $i . ",
        'KATHΓΟΙΑ " . $i . "')";
        $resins2 = mysqli_query($Connection,$insert2);
        if(!$resins2)
        {
            $error = true;
            $error_val = "insert2" . "<br>" . $i . "<br>" . $insert2;
        }
    }
}
if(!$error)
{

```



```

for($a=1;$a<$MaxStores + 1;$a++)
{
    for($i=1;$i<$MaxCategories + 1;$i++)
    {
        for($k=1;$k<$MaxProducts + 1;$k++)
        {
            $insert3 = "INSERT INTO products VALUES(" . $k .
", 'ΤΙΠΟΪΟΝ " . $k . "', " . $i . "', " . $a . "', " . mt_rand(8, 29) . ")";
            $resins3 = mysqli_query($Connection,$insert3);
            if(!$resins3)
            {
                $error = true;
                $error_val = "insert3" . "<br>" . $k . "<br>" .
$insert3;
            }
        }
    }
}
if(!$error)
{
    $OrderSid = array();
    $exclude = array();
    $exclude[0]=-1;
    $stamp_arr = array();
    for($i=1;$i<$MaxOrders + 1;$i++)
    {
        $OrderSid[$i-1] = mt_rand();
        if(($MaxProducts * $MaxCategories)>20)
            $MaxOrderItems = mt_rand(1, 19);
        else
            $MaxOrderItems = mt_rand(1, ($MaxProducts *
$MaxCategories)-1);
    }
}

```

```

        for($k=0;$k<$MaxOrderItems + 1;$k++)
        {
            $stamp_arr[$i] = (time()-((3600 * 24)*$i));
            $exclude[$k] = rand_except(0, ($MaxProducts *
$MaxCategories)-1, $exclude);
            $insert4 = "insert into orders select " . $OrderSid[$i-1] . " ,
productid, storeid, categoryid, price, " . mt_rand(1,3) . " , " . $stamp_arr[$i] . " from
products limit " . $exclude[$k] . " ,1";
            $resins4 = mysqli_query($Connection,$insert4);
            if(!$resins4)
            {
                $error = true;
                $error_val = "insert4" . "<br>". $i . " + " . $k . "<br>" .
$insert4;
            }
        }
        $exclude = array();
        $exclude[0]=-1;
    }
}
if(!$error)
{
    for($i=1;$i<$MaxCustomers + 1;$i++)
    {
        $insert5 = "INSERT INTO customers VALUES(" . $OrderSid[$i-1] .
", 'ONOMA " . $i . "', 'ΕΠΩΝΥΜΟ " . $i . "', 'ΔΙΕΥΘΥΝΣΗ " . $i . "', 'ΠΟΛΗ " . $i .
"', " . $i . "', 'ΧΩΡΑ " . $i . "', " . $i . "', " . $i . "', " . $stamp_arr[$i] . ")";
        $resins5 = mysqli_query($Connection,$insert5);
        if(!$resins5)
        {
            $error = true;
            $error_val = "insert5" . "<br>". $i . "<br>" . $insert5;
        }
    }
}

```

```

    }
}
if(!$error)
{
    for($i=1;$i<$MaxUsers + 1;$i++)
    {
        $insert6 = "INSERT INTO passlist VALUES(" . $i . ", 'XPΗΣΤΗΣ " .
        $i . ", " . $i . ")";
        $resins6 = mysqli_query($Connection,$insert6);
        if(!$resins6)
        {
            $error = true;
            $error_val = "insert6" . "<br>" . $i . "<br>" . $insert6;
        }
    }
}
if(!$error)
{
    $del = "DELETE FROM sessions";
    $res = mysqli_query($Connection,$del);
    if(!$res)
    {
        $error = true;
        $error_val = "del" . "<br>" . $del;
    }
}
if($error)
{
    echo "<HTML>";
    echo "<HEAD>";
    echo "<TITLE>ΑΠΟΤΥΧΙΑ ΔΗΜΙΟΥΡΓΙΑΣ ΔΕΔΟΜΕΝΩΝ</TITLE>";
    echo "<style type='text/css'>";

```

```

echo "table {background-color:#FFF;      border-collapse:collapse;}";
echo "</style>";
echo "</HEAD>";
echo "<BODY style='background-color:#CCC'>";
echo "<div style='width:50%;height:40%;top:30%; left:30%; border:solid;
background-color:#CF0'>";
    echo "<table border='1'>";
    echo "<caption>ΔΗΜΙΟΥΡΓΗΘΗΚΑΝ</caption>";
    echo "<tr>";
    echo "<td>ΚΑΤΑΣΤΗΜΑΤΑ</td><td style='width:50px;text-align:center'>" .
$MaxStores . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td>ΚΑΤΗΓΟΡΙΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxCategories . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td>ΠΟΙΪΟΝΤΑ</td><td style='width:50px;text-align:center'>" .
$MaxProducts . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td>ΠΑΡΑΓΓΕΛΙΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxOrders . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td>ΠΕΛΑΤΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxCustomers . "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td>ΧΡΗΣΤΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxUsers . "</td>";
    echo "</tr>";
    echo "<tr>";

```

```

        echo "<td>ΜΕΓΙΣΤΟΣ ΑΡ. ΕΓΓΡΑΦΩΝ ΠΑΡΑΓΓΕΛΙΩΝ</td><td
style='width:50px;text-align:center>' . $MaxOrderItems . "</td>";
    echo "</tr>";
    echo "</table><br><br>";
    echo "<input type='button' value='ΕΞΟΔΟΣ' onclick='Exit()'/>";
    echo "INFO:&nbsp;&nbsp;&nbsp;" . $error_val . "<br>" . "ERROR
CODE:&nbsp;&nbsp;&nbsp;" . mysqli_errno($Connection) . "<br>" . "ERROR
DESCRIPTION:&nbsp;&nbsp;&nbsp;" . mysqli_error($Connection) . "<br>";
    echo "<input type='button' value='ΕΞΟΔΟΣ' onclick='Exit()'/>";
    echo "</div>";
    echo "<script type='text/javascript'>function Exit()
{parent.location.href='index.php'}</script>";
    mysqli_close($Connection);
    echo "</BODY></HTML>";
}
else
{
    mysqli_close($Connection);
    setcookie($_COOKIE[$_COOKIE["session"]], "", time()-3600);
    setcookie($_COOKIE["session"], "", time()-3600);
    setcookie($_COOKIE["customer"], "", time()-3600);
    echo "<HTML>";
    echo "<HEAD>";
    echo "<TITLE>ΔΗΜΙΟΥΡΓΙΑ ΔΕΔΟΜΕΝΩΝ</TITLE>";
    echo "<style type='text/css'>";
    echo "table {background-color:#FFF;            border-collapse:collapse;}";
    echo "</style>";
    echo "</HEAD>";
    echo "<BODY style='background-color:#CCC'>";
    echo "<div align = 'center' style='width:50%;height:50%;top:40%; left:30%;
border:solid; background-color:#CF0'>";
    echo "<table border='1' align='center'>";
    echo "<caption>ΔΗΜΙΟΥΡΓΗΘΗΚΑΝ</caption>";

```

```

echo "<tr>";
echo "<td>ΚΑΤΑΣΤΗΜΑΤΑ</td><td style='width:50px;text-align:center'>" .
$MaxStores . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΚΑΤΗΓΟΡΙΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxCategories . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΠΟΙΪΟΝΤΑ</td><td style='width:50px;text-align:center'>" .
$MaxProducts . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΠΑΡΑΓΓΕΛΙΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxOrders . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΠΕΛΑΤΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxCustomers . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΧΡΗΣΤΕΣ</td><td style='width:50px;text-align:center'>" .
$MaxUsers . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td>ΜΕΓΙΣΤΟΣ ΑΡ. ΕΓΓΡΑΦΩΝ ΠΑΡΑΓΓΕΛΙΩΝ</td><td
style='width:50px;text-align:center'>" . $MaxOrderItems . "</td>";
echo "</tr>";
echo "</table><br><br>";
echo "<input type='button' value='ΕΞΟΔΟΣ' onclick='Exit()'/>";
echo "</div>";
echo "<script type='text/javascript'>function Exit()
{parent.location.href='index.php'}</script>";

```

```

        echo "</BODY></HTML>";
//    header("Location: index.php");
}

function rand_except($min, $max, $except)
    //function returns a random integer between min and max, just like function rand()
    does.
    // Difference to rand is that the random generated number will not use any of the
    values
    // placed in $except. ($except must therefore be an array)
    // function returns false if $except holds all values between $min and $max.
    {
        //first sort array values
        sort($except, SORT_NUMERIC);
        //calculate average gap between except-values
        $except_count = count($except);
        $avg_gap = ($max - $min + 1 - $except_count) / ($except_count + 1);
        if ($avg_gap <= 0)
            return false;
        //now add min and max to $except, so all gaps between $except-values can be
        calculated
        array_unshift($except, $min - 1);
        array_push($except, $max + 1);
        $except_count += 2;
        //iterate through all values of except. If gap between 2 values is higher than average
        gap,
        // create random in this gap
        for ($i = 1; $i < $except_count; $i++)
            if ($except[$i] - $except[$i - 1] - 1 >= $avg_gap)
                return mt_rand($except[$i - 1] + 1, $except[$i] - 1);
        return false;
    }
?>

```

Ο παραπάνω κώδικας εκτελεί πολλές εντολές της **SQL** τύπου **INSERT**.

Εκτός από την εντολή της **PHP mt_rand** χρησιμοποιείτε και μία συνάρτηση [rand_except](#) ο κώδικας της οποίας είναι παρμένος αυτούσιος από το **internet**.

Η συνάρτηση αυτή παράγει μεγάλους τυχαίους αριθμούς αλλά η τρίτη της παράμετρος είναι μια μεταβλητή πίνακας που έχει όλα τα προηγούμενα αποτελέσματα που έχει παράξει η συνάρτηση αυτή, ώστε να μην δίνει σε καμία περίπτωση το ίδιο αποτέλεσμα δεύτερη φορά.

Η χρήση της είναι σημαντική στην τυχαία παραγωγή προϊόντων τα οποία εξαρτώνται και από την κατηγορία αλλά και το κατάστημα.

Για παράδειγμα μπορεί να υπάρχουν εγγραφές που να λένε ΠΡΟΪΟΝ1 ΚΑΤΗΓΟΡΙΑ1 ΚΑΤΑΣΤΗΜΑ1 και ΠΡΟΪΟΝ1 ΚΑΤΗΓΟΡΙΑ1 ΚΑΤΑΣΤΗΜΑ2.

Αν χρησιμοποιούσαμε την απλή συνάρτηση **mt_rand** υπήρχε ο κίνδυνος να παραχθούν διπλές εγγραφές κάτι που θα παρήγαγε λάθος στον κώδικα από την βάση **MySQL** και δεν θα ολοκληρωνόταν η δημιουργία των δεδομένων σε όλους τους πίνακες.

Μετά την δημιουργία των δεδομένων φορτώνεται μία σελίδα που απεικονίζει τον αριθμό των εγγραφών σε κάθε πίνακα, που δημιουργήθηκαν.