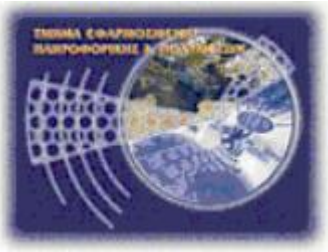
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή εργασία

Δημιουργία ιστοσελίδας για την αξιολόγηση προσφορών από διαδικτυακά καταστήματα

Ιωάννης Καλοχριστιανάκης (Α.Μ. 2336)

Τριανταφύλλου Άγγελος (Α.Μ. 2156)

Επιβλέπων Καθηγητής: Νίκος Παπαδάκης

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ 5
ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ 7
ΠΕΡΙΛΗΨΗ 8
ABSTRACT 8
1 ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ 9
2 ΕΙΣΑΓΩΓΗ 9
2.1 Η αλληλεπίδραση του χρήστη με το Διαδίκτυο 9
2.2 Οι γλώσσες προγραμματισμού σήμερα 10
2.3 Οι κύριες τεχνολογίες ανάπτυξης ιστοσελίδων 11
2.4 Το υπάρχον μοντέλο ανάπτυξης 12
3 ΥΠΑΡΧΟΥΣΕΣ ΤΕΧΝΟΛΟΓΙΕΣ 13
3.1 Server side interpreters: Php 13
3.1.1 Σχεδιαστική Φιλοσοφία 13
3.2 Java & Java Server Pages 14
3.2.1 Σχεδιαστική Φιλοσοφία 14
3.2.2 Java Server Pages 15
3.3 Asp & Asp.net 16
3.3.1 Σχεδιαστική Φιλοσοφία και Συντακτικό 17
3.3.2 Το .Net framework και η καινοτομική αρχιτεκτονική της asp.net 17
3.4 Javascript 18
3.4.1 Σχεδιαστική φιλοσοφία 18
3.4.2 Σύνταξη και χαρακτηριστικά 19
3.5 Flash & Flex 20
3.5.1 Γραφικές και πολυμεσικές δυνατότητες 21
3.5.2 Ο προγραμματισμός στο flash 22
4 ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΩΝ ΔΕΔΟΜΕΝΩΝ 23
4.1 XAMPP 23
4.1.1 Εγκατάσταση XAMPP 23
5 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ ΣΧΕΣΕΩΝ 27
6 ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ 28
7 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ 29
8 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΥΣΤΗΜΑΤΟΣ 35
8.1 Βασικά Χαρακτηριστικά Σελίδων Συστήματος 35

8.2 Login - Logout 37
8.3 Χρήστες Διαχειριστή 39
8.3.1 Αρχική Οθόνη Χρηστών 39
8.3.2 Δημιουργία Χρήστη 39
8.3.3 Διαγραφή Χρήστη 40
8.4 Κατηγορίες Διαχειριστή 42
8.4.1 Αρχική Οθόνη Κατηγοριών 42
8.4.2 Δημιουργία Κατηγορίας 42
8.4.3 Διαγραφή Κατηγορίας 44
8.5 Υποκατηγορίες Διαχειριστή 45
8.5.1 Αρχική Οθόνη Υποκατηγοριών 45
8.5.2 Δημιουργία Υποκατηγορίας 45
8.5.3 Διαγραφή Υποκατηγορίας 47
8.6 Καταστήματα Διαχειριστή 48
8.6.1 Αρχική Οθόνη Καταστημάτων 48
8.6.2 Δημιουργία Καταστήματος 48
8.6.3 Διαγραφή Καταστήματος 50
8.7 Προϊόντα Διαχειριστή 51
8.7.1 Αρχική Οθόνη Προϊόντων 51
8.7.2 Δημιουργία Προϊόντος 51
8.7.3 Διαγραφή Προϊόντος 53
8.8 Παραγγελίες Διαχειριστή 54
8.9 Εγγραφή χρήστη 55
8.10 Σύνδεση στην πλατφόρμα 56
8.10.1 Σύνδεση απλού χρήστη 56
8.10.2 Σύνδεση επαγγελματία χρήστη 56
8.11 Προφίλ χρήστη 57
8.12 Αναζήτηση 58
8.12.1 Κριτήρια Αναζήτησης 58
8.13 Όλα τα Καταστήματα 62
8.13.1 Αρχική Οθόνη Όλα τα Καταστήματα 62
8.13.2 Παρουσίαση προϊόντων καταστήματος 63
8.13.3 Παρουσίαση όλων των αξιολογήσεων για συγκεκριμένο κατάστημα 64
8.14 Όλα τα προϊόντα 65

8.15 Προϊόντα ανα κατηγορία 66
8.16 Καλάθι 68
8.16.1 Αρχική Οθόνη Καλαθιού 68
8.16.2 Προσθήκη στο καλάθι 68
8.16.3 Διαγραφή από το Καλάθι 71
8.17 Παραγγελίες 72
8.17.1 Αρχική Οθόνη Παραγγελίας 72
8.17.2 Οι παραγγελίες μου 74
8.17.3 Ακύρωση Παραγγελίας 74
8.17.4 Αξιολόγηση Παραγγελίας 75
8.18 Λογαριασμός επαγγελματία χρήστη 79
8.18.1 Οι παραγγελίες των καταστημάτων μου 79
8.18.2 Τα συνολικά στατιστικά των καταστημάτων μου 80
8.18.3 Προσθήκη καταστήματων 80
8.18.4 Προσθήκη Προιόντων για τα καταστήματα του 80
ΒΙΒΛΙΟΓΡΑΦΙΑ 81

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1:Εγκατάσταση xampp 23
Εικόνα 2: Ρυθμίσεις xampp 24
Εικόνα 3:Ολοκλήρωση Ρυθμίσεων xampp 24
Εικόνα 4:Εκκίνηση xampp Control Panel 25
Εικόνα 5:Εκκίνηση Apache και Mysql 25
Εικόνα 6:Επιλογή admin λειτουργίας για τον Apache 26
Εικόνα 7: Ολοκλήρωση εγκατάστασης xampp 26
Εικόνα 8: Διάγραμμα Οντοτήτων Σχέσεων 27
Εικόνα 9:Έλεγχος κωδικού-ονόματος χρήστη 37
Εικόνα 10: Εισαγωγή στο σύστημα - Μήνυμα καλωσορίσματος 37
Εικόνα 11: Αρχική Οθόνη Χρηστών 39
Εικόνα 12: Φόρμα Προσθήκης Νέου Χρήστη 40
Εικόνα 13: Ολοκλήρωση προσθήκης Νέου Χρήστη 40
Εικόνα 14: Διαγραφή Χρήστη 1 41
Εικόνα 15: Διαγραφή Χρήστη 2 41
Εικόνα 16: Αρχική Οθόνη Κατηγοριών 42
Εικόνα 17: Φόρμα Προσθήκης Νέας Κατηγορίας 43
Εικόνα 18: Ολοκλήρωση προσθήκης Νέας Κατηγορίας 43
Εικόνα 19: Διαγραφή Κατηγορίας 1 44
Εικόνα 20: Διαγραφή Κατηγορίας 2 44
Εικόνα 21: Αρχική Οθόνη Υποκατηγοριών 45
Εικόνα 22: Φόρμα Προσθήκης Νέας Υποκατηγορίας 46
Εικόνα 23: Ολοκλήρωση προσθήκης Νέας Υποκατηγορίας 46
Εικόνα 24: Διαγραφή ΥποΚατηγορίας 1 47
Εικόνα 25: Διαγραφή Υποκατηγορίας 2 47
Εικόνα 26: Αρχική Οθόνη Καταστημάτων 48
Εικόνα 27: Φόρμα Προσθήκης Καταστήματος 49
Εικόνα 28:Ολοκλήρωση προσθήκης Νέου Καταστήματος 49
Εικόνα 29: Διαγραφή Καταστήματος1 50
Εικόνα 30:Διαγραφή Καταστήματος 2 50
Εικόνα 31: Αρχική Οθόνη Προϊόντων 51
Εικόνα 32: Φόρμα Προσθήκης Νέου Προϊόντος 52
Εικόνα 33: Ολοκλήρωση προσθήκης Νέου Προϊόντος 52
Εικόνα 34: Διαγραφή Προϊόντος 1 53
Εικόνα 35: Διαγραφή Προϊόντος 2 53
Εικόνα 36: Παραγγελίες - Περιβάλλον Διαχειριστή 54
Εικόνα 37: Εγγραφή χρήστη 55
Εικόνα 38: Μενού απλού χρήστη 56
Εικόνα 39: Μενού επαγγελματία χρήστη 56
Εικόνα 40: Προφίλ χρήστη 57
Εικόνα 41: Αρχική Οθόνη Αναζήτησης (μη συνδεδεμένος) 58
Εικόνα 42: Αναζήτηση του όρου tech 59

Εικόνα 43: Επιστροφή αποτελεσμάτων που περιέχουν είτε στο όνομα, είτε στην περιγραφή, είτε στην κατηγορία, είτε στην υποκατηγορίατον όρο tech (μη συνδεδεμένος) 59
Εικόνα 44: Επιστροφή αποτελεσμάτων που περιέχουν είτε στο όνομα, είτε στην περιγραφή, είτε στην κατηγορία, είτε στην υποκατηγορίατον όρο tech (συνδεδεμένος) 60
Εικόνα 45: Προσθήκη στο καλάθι από τα αποτελέσματα αναζήτησης και διατήρηση αυτών 60
Εικόνα 46: Αρχική Οθόνη όλων των καταστημάτων 62
Εικόνα 47: Παρουσίαση Προιόντων ενός συγκεκριμένου καταστήματος 63
Εικόνα 48: Προσθήκη ενός προιόντος στο καλάθι και διατήρηση των προιόντων του καταστήματος που επιλέξαμε 63
Εικόνα 49: Παρουσίαση όλων των αξιολογήσεων για ένα κατάστημα 64
Εικόνα 50: Αρχική Οθόνη Προϊόντων 65
Εικόνα 51: Αρχική Οθόνη Προϊόντων ανά κατηγορία 66
Εικόνα 52: Επιλογή κατηγορίας Προϊόντων – Εμφάνιση Υποκατηγοριών 66
Εικόνα 53: Επιλογή υποκατηγορίας Προϊόντων – Εμφάνιση Προϊόντων Υποκατηγοριας 67
Εικόνα 54: Προσθήκη ενός προιόντος στο καλάθι και διατήρηση των προιόντων της υποκατηγορίας που επιλέξαμε 67
Εικόνα 55:Αρχική Οθόνη Καλαθιού 68
Εικόνα 56: Καλάθι 1 69
Εικόνα 57: Καλάθι 2 69
Εικόνα 58: Διαγραφή από το καλάθι 1 71
Εικόνα 59: Διαγραφή από το Καλάθι 2 71
Εικόνα 60: Αρχική Οθόνη Παραγγελίας – Ολοκλήρωση Παραγγελίας 72
Εικόνα 61: Οι παραγγελίες μου 74
Εικόνα 62: Ακύρωση Παραγγελίας 75
Εικόνα 63: Αξιολόγηση Παραγγελίας1 76
Εικόνα 64: Αξιολόγηση Παραγγελίας 2 76
Εικόνα 65: Αξιολόγηση Παραγγελίας 3 77
Εικόνα 66: Οι Παραγγελίες των καταστημάτων μου 79
Εικόνα 67: Τα συνολικά στατιστικά των καταστημάτων μου 80

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Διαθέσιμες λειτουργίες συστήματος ανά ρόλο 34
Πίνακας 2: Περιεχόμενα φακέλων-αρχείων 36
Πίνακας 3: Ενέργειες κώδικα κάθε αρχείου στον φάκελο admin 36

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία πραγματοποιήσαμε την ανάπτυξη ενός συστήματος για μια πλατφόρμα διαχείρισης ηλεκτρονικών καταστημάτων. Αρχικά, πραγματοποιήσαμε την εγκατάσταση των αναγκαίων προγραμμάτων με τα οποία μπορούμε να υποστηρίξουμε ένα διαδικτυακό σύστημα διαχείρισης βάσεων δεδομένων. Στη συνέχεια, η καταγραφή, η μοντελοποίηση και η αποτύπωση των αναγκαίων χαρακτηριστικών του συστήματος στις διάφορες φόρμες λειτουργίας ήταν μια ακολουθία από ενέργειες με τις οποίες ασχολείται η παρούσα εργασία. Επίσης, υλοποιήσαμε τη χρήση βάσει ρόλων διαχειριστή – χρηστών (απλοί χρήστες, επαγγελματίες χρήστες) στο σύστημα όπου ο κάθε χρήστης ανάλογα με το ρόλο του έχει συγκεκριμένες λειτουργίες στη διάθεση του. Τέλος, ακολουθήσαμε διαφορετικά σενάρια χρήσης και λειτουργίας με σκοπό να διασφαλίσουμε τη εύρυθμη λειτουργία του συστήματος μας.

ABSTRACT

In this thesis, we implemented a database management system for a platform for eshops. First of all, we proceeded to the installation of the necessary programs which assisted us in creating an online database system. Moreover, we noted down the requirement analysis of our system in order to clarify the whole functionality. Furthemore, we enabled the role-based use of administrators and simple users of the system, giving to each role different functionalities for use. Finally, we run a large number of case studies to ensure the proper functionality of our system.

1 ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Ο σκοπός της παρούσας εργασίας είναι ανάπτυξη ενός συστήματος για μια πλατφόρμα διαχείρισης ηλεκτρονικών καταστημάτων. Αρχικά, πραγματοποιήσαμε την εγκατάσταση των αναγκαίων προγραμμάτων με τα οποία μπορούμε να υποστηρίξουμε ένα διαδικτυακό σύστημα διαχείρισης βάσεων δεδομένων. Στη συνέχεια, η καταγραφή, η μοντελοποίηση και η αποτύπωση των αναγκαίων χαρακτηριστικών του συστήματος στις διάφορες φόρμες λειτουργίας ήταν μια ακολουθία από ενέργειες με τις οποίες ασχολείται η παρούσα εργασία. Επίσης, υλοποιήσαμε τη χρήση βάσει ρόλων διαχειριστή – χρηστών (απλοί χρήστες, επαγγελματίες χρήστες) στο σύστημα όπου ο κάθε χρήστης ανάλογα με το ρόλο του έχει συγκεκριμένες λειτουργίες στη διάθεση του. Τέλος, ακολουθήσαμε διαφορετικά σενάρια χρήσης και λειτουργίας με σκοπό να διασφαλίσουμε τη εύρυθμη λειτουργία του συστήματος μας.

2 ΕΙΣΑΓΩΓΗ

2.1 Η αλληλεπίδραση του χρήστη με το Διαδίκτυο

Ο παγκόσμιος ιστός (world wide web) είναι ίσως ο τομέας του διαδικτύου που παρουσιάζει τις εντυπωσιακότερες αλλαγές. Πράγματι, ξεκινώντας από απλές σελίδες υπερ-κειμένου, είναι πλέον σε θέση να υποστηρίζει πολύπλοκες πολυμεσικές, πολυχρηστικές εφαρμογές όπως το facebook και το youtube. Συγκρίνοντας παρόν με παρελθόν, παρατηρούμε ότι οι υπολογιστικές αρχιτεκτονικές έχουν αλλάξει εντυπωσιακά, τα μοντέλα αλληλεπίδρασης με το χρήστη έχουν επαναπροσδιοριστεί, εντούτοις η βάση του παγκόσμιου ιστού παραμένει σταθερή: Για να υποστηρίξει την εκρηκτική εξάπλωση της τεχνολογίας, η βάση αυτή επεκτάθηκε προς πάσα κατεύθυνση με ετερόκλητα, άναρχα στοιχεία και ανταγωνιστικές τεχνολογίες.

Με κριτήριο το μοντέλο αλληλεπίδρασης με το χρήστη και τις υποστηριζόμενες τεχνολογίες, μπορούμε να εντοπίσουμε τρία διακριτά στάδια στην ανάπτυξη του Παγκόσμιου Ιστού (1):

1. Αρχικά ο παγκόσμιος ιστός αποτελείται από στατικές σελίδες, οι οποίες κατά κύριο λόγο περιέχουν υπερκείμενο και διανθίζονται σταδιακά από απλά γραφικά, κινούμενες εικόνες και ήχους. Με την έλευση του Macromedia flash, οι σελίδες αποκτούν και κινούμενα γραφικά.

2. Ακολούθως ένα μεγάλο μέρος του παγκόσμιου ιστού γίνεται δυναμικό, δηλαδή το περιεχόμενο που προβάλλεται στο χρήστη παράγεται από κάποιο πρόγραμμα στον εξυπηρετητή ανάλογα με τις δράσεις του χρήστη. Γλώσσες όπως η php επιτρέπουν στους προγραμματιστές να αναπτύσσουν ολόκληρα προγράμματα που παράγουν δυναμικό περιεχόμενο, χωρίς να γνωρίζουν την υπολογιστική αρχιτεκτονική του εξυπηρετητή. Νέες εφαρμογές όπως τα blogs και τα forums κάνουν την εμφάνισή τους.

3. Η εξέλιξη αυτή φέρνει τα web applications και web services. Η αλληλεπίδραση της ιστοσελίδας με το χρήστη αλλάζει σημαντικά ώστε να προσομοιώνει παραθυρική εφαρμογή. Αυτό γίνεται εφικτό με τη χρήση κώδικα που τρέχει στον υπολογιστή του χρήστη, που επιτρέπει τη

δυναμική ανταλλαγή πληροφοριών από τον εξυπηρετητή. Ιστορικά λοιπόν, ο Παγκόσμιος Ιστός ξεκίνησε ως σύνολο στατικών σελίδων υπερκειμένου. Επεκτάθηκε με την εισαγωγή ενός επιπέδου προγραμματισμού στον εξυπηρετητή ώστε οι σελίδες να παράγονται δυναμικά, ενώ πρόσφατα εισήχθη (ή ακριβέστερα, συστηματοποιήθηκε) ένα ακόμα επίπεδο προγραμματισμού στον υπολογιστή-πελάτη, ώστε πλέον να μιλάμε για διαδικτυακές εφαρμογές αντί για ιστοσελίδες. Κάθε μια από τις παραπάνω φάσεις συνιστά και μια επανάσταση στο τρόπο που οι ιστοσελίδες αλληλεπιδρούν με τους χρήστες τους. Επίσης,, κάθε μια από τις παραπάνω φάσεις ανάπτυξης πραγματώθηκε με την ωρίμανση και συστηματοποίηση ενός συνόλου τεχνολογιών και γλωσσών προγραμματισμού. Έτσι από την HTML περάσαμε στις server-side γλώσσες και την XHTML ενώ πρόσφατα με το AJAX συστηματοποιείται και η χρήση javascript(2)(3) ενώ προτείνονται ειδικές γλώσσες (πχ OWL) για την ανταλλαγή δεδομένων μεταξύ ιστοτόπων. Αφού η εξέλιξη του διαδικτύου είναι συνδεδεμένη με τις γλώσσες προγραμματισμού, σκόπιμο κρίνεται να εξετάσουμε τις σχετικές τεχνολογίες.

2.2 Οι γλώσσες προγραμματισμού σήμερα

Οι γλώσσες προγραμματισμού θα μπορούσαν να χωριστούν σε τρείς κύριες κατηγορίες:

1. Γλώσσες με μεταφραστή (compilers). Κύριο χαρακτηριστικό τους είναι η off-line μετάφραση του πηγαίου κώδικα σε κώδικα άμεσα εκτελέσιμο από μια υπολογιστική μηχανή. Παράγεται δηλαδή κατά τη μετάφραση ένα εκτελέσιμο αρχείο από τη μηχανή, που αποτελεί και το πρόγραμμα. Αντιπρόσωποι αυτών των γλωσσών είναι η C, η Pascal αλλά και η Java (4).

2. Γλώσσες σήμανσης (markup languages), όπως η XML και η HTML, οι οποίες περιγράφουν σε μια καλώς ορισμένη δομή χαρακτηριστικά ή ιδιότητες ενός αντικειμένου.

3. Γλώσσες με διερμηνευτή (interpreter) που έχουν επικρατήσει να καλούνται και scripting-γλώσσες. Σε αυτές τις γλώσσες ο πηγαίος κώδικας διερμηνεύεται σε πραγματικό χρόνο από τον διερμηνευτή, ή μεταφράζεται σε αντικειμενικό κώδικα διαφανώς προς το χρήστη, δηλαδή δεν παράγεται κάποιο εκτελέσιμο αρχείο από τη διαδικασία. Ας σημειωθεί εδώ ότι σε πολλές περιπτώσεις παρέχεται η δυνατότητα αποθήκευσης του αντικειμενικού κώδικα σε ένα εκτελέσιμο αρχείο (για παράδειγμα στην Lightwave Scripting Language) (5), γεγονός που κάνει δυσδιάκριτα τα όρια μεταφραστή και διερμηνευτή. Στο εξής θα αναφερόμαστε στις γλώσσες με διερμηνευτή ως scripting languages και στα προγράμματα μιας τέτοιας γλώσσας ως scripts.

Ο κόσμος των διερμηνευτών θα μπορούσε να χαρακτηριστεί στάσιμος μέχρι την έλευση του Διαδικτύου και των σχετικών εφαρμογών(1). Πράγματι, παραλλαγές γλωσσών με μεταγλωττιστή όπως οι C και Pascal χρησιμοποιούνταν κατά κόρον για την ανάπτυξη εφαρμογών σε εταιρικά και παραθυρικά περιβάλλοντα. Διερμηνευτές (interpreters) χρησιμοποιούνταν κυρίως για βοηθητικές εργασίες (shel scripts, automation scripts). Η βελτιωμένη απόδοση του μεταγλωττισμένου κώδικα αλλά και η σχετική αντοχή του σε τεχνικές reverse engineering είναι πιθανότατα οι κύριοι λόγοι για τη κυριαρχία των μεταγλωττιστών.

Η επέκταση του διαδικτύου ωστόσο έφερε αλλαγές στα μοντέλα ανάπτυξης. Ένα σημαντικό μερίδιο εφαρμογών πλέον εκτελείται σε ετερογενή περιβάλλοντα αγνώστου αρχιτεκτονικής, ενώ η πλειοψηφία των (δυναμικών) δικτυακών ιστοσελίδων φιλοξενούνται σε κάποιον εξυπηρετητή μιας

εταιρίας παροχής υπηρεσιών διαδικτύου. Συνεπώς τόσο οι δυναμικές ιστοσελίδες όσο και οι διαδικτυακές εφαρμογές, είναι απαραίτητο να είναι ανεξάρτητες αρχιτεκτονικής και ευέλικτες στην ανάπτυξη. Νέες γλώσσες προγραμματισμού, βασισμένες στο μοντέλο του διερμηνευτή σχεδιάζονται με στόχο να υποστηρίξουν την ανάπτυξη τέτοιων εφαρμογών. Δεδομένου ότι η υπολογιστική ισχύς επιτρέπει πλέον τη δημιουργία διερμηνευτών υψηλής πολυπλοκότητας, γλώσσες όπως η Php και η Asp γίνονται δημοφιλείς.

Στους περισσότερους διερμηνευτές σήμερα, ο πηγαίος κώδικας μετατρέπεται (μεταγλωττίζεται) αυτόματα σε ενδιάμεσο κώδικα (byte-code), ο οποίος εκτελείται από μια ιδεατή μηχανή. Η Ιδεατή μηχανή μεταφέρεται σε κάθε υποστηριζόμενη αρχιτεκτονική, επιτρέποντας στον ενδιάμεσο κώδικα να είναι ανεξάρτητος αρχιτεκτονικής. Γλώσσες όπως η Php, η Java αλλά και η Visual Basic παλαιότερα, ακολουθούν αυτό το παράδειγμα(4). Λόγω της ανάγκης για ανεξαρτησία από την υπολογιστική αρχιτεκτονική αλλά και την αυξημένη απόδοση, προτιμάται η χρήση byte-code από τη δημιουργία εγγενούς κώδικα.

Ας σημειωθεί εδώ ότι εκτός από το διαδίκτυο οι διερμηνευτές βρίσκουν εφαρμογή και σε άλλες περιπτώσεις: Οι περισσότερες σύγχρονες εφαρμογές διαθέτουν ένα σύστημα αυτοματοποίησης (scripting system) το οποίο επιτρέπει σε προχωρημένους χρήστες να επεκτείνουν τις λειτουργίες τους γράφοντας μικρά προγράμματα (scripts). Η φιλοσοφία αυτή βρίσκει εφαρμογή και στη ψηφιακή διασκέδαση, αφού τα περισσότερα βιντεοπαιχνίδια χρησιμοποιούν διερμηνευτές (πχ unreal script και Lua) για να περιγράψουν δυναμικά τόσο τα σενάρια παιχνιδιού όσο και στοιχεία λογικής (6)(7).

2.3 Οι κύριες τεχνολογίες ανάπτυξης ιστοσελίδων

Η ανάπτυξη του παγκόσμιου ιστού έφερε και ένα σύνολο νέων τεχνολογιών ανάπτυξης εφαρμογών, το οποίο χαρακτηρίζεται από έντονη ετερογένεια. Για παράδειγμα, μια μέση διαδικτυακή εφαρμογή απαιτεί τη γνώση τουλάχιστον τριών γλωσσών για τη συγγραφή της (8)(9)(10):

• Php, Asp, Jsp ή παρόμοια server-side γλώσσα για τη δημιουργία δυναμικού περιεχομένου από τη μεριά του εξυπηρετητή,

• HTML (XML, CSS) για τη περιγραφή του περιεχομένου που θα δείξει ο φυλλομετρητής (web browser)

• Javascript για τη δυναμική παρουσίαση και διαχείριση του περιεχομένου από τη μεριά του πελάτη. Ας σημειωθεί εδώ ότι κάθε browser φέρει τι δική του ιδιαίτερη υλοποίηση της γλώσσας, με αποτέλεσμα να απαιτείται σημαντική αφαίρεση (abstraction) κατά τον προγραμματισμό, ώστε να υποστηριχθούν όλοι οι γνωστοί web browsers.

Σε πραγματικές διαδικτυακές εφαρμογές χρησιμοποιούνται περισσότερες τεχνολογίες και γλώσσες, όπως για παράδειγμα οι Flash/actionscript για δυναμικά γραφικά και video και η SQL για διασύνδεση με βάσεις δεδομένων. Εκτενής αναφορά στις ιδιαιτερότητες κάθε γλώσσας γίνεται στα επόμενα κεφάλαια.

2.4 Το υπάρχον μοντέλο ανάπτυξης

Παρατηρούμε ότι η συντριπτική πλειοψηφία των παραπάνω τεχνολογιών χρησιμοποιείται για την υλοποίηση client-server εφαρμογών και τη μεταφορά δεδομένων μέσω δικτύου. Ειδικά στη περίπτωση των διαδικτυακών εφαρμογών, παρατηρείται μια πολύπλοκη και ετερογενής συμμετοχή διαφορετικών τεχνολογιών κατά τη συγγραφή της εφαρμογής (8)(1).

Πράγματι, στη συνήθη περίπτωση, ο προγραμματιστής καλείται να χρησιμοποιήσει μια server-side scripting language (php, jsp, asp), η οποία με χρήση αλφαριθμητικών θα παράγει σε μορφή κειμένου τον πηγαίο κώδικα μιας άλλης γλώσσας (Html, Javascript). Ουσιαστικά, ο πηγαίος κώδικας της εφαρμογής στον υπολογιστή-πελάτη βρίσκεται εμφωλευμένος μέσα στις διάφορες ρουτίνες του κώδικα της εφαρμογής στον server. Το μοντέλο αυτό, παρόλο που χαρακτηρίζεται από ευελιξία, όπως θα δειχθεί παρακάτω, πάσχει από σοβαρά προβλήματα αναγνωσιμότητας, απόδοσης, συντήρησης του κώδικα αλλά και επεκτασιμότητας.

Ας σημειωθεί εδώ ότι οι περισσότερες server-side γλώσσες δίνουν την δυνατότητα να ενσωματωθούν μέσα σε αρχεία .html ή xml, τα οποία «φιλτράρονται» αυτόματα από το διερμηνευτή πριν προβληθούν στο χρήστη (8). Παρ ότι βολική, αυτή η δυνατότητα επιτρέπει στους προγραμματιστές να “κρύβουν” κομμάτια κώδικα σε περισσότερα αρχεία, δυσχεραίνοντας περισσότερο τη διαχείρισή του.

Το κύριο ζήτημα που ανακύπτει στο υπάρχον μοντέλο ανάπτυξης διαδικτυακών εφαρμογών είναι ότι χρησιμοποιείται ένα ετερογενές σύνολο από γενικές, δίκτυο-αγνωστικές γλώσσες για να υλοποιηθούν καθαρά δικτυακές εφαρμογές (client-server). Μάλιστα, για να διατηρηθεί η προς τα πίσω συμβατότητα με το πρωτόκολλο http, οι περισσότερες server-side γλώσσες επικεντρώνονται στη διαχείριση αλφαριθμητικών ώστε να παράγουν τον κώδικα (html, javascript κλπ) που θα παρουσιαστεί στο πελάτη.

Ένα παράδειγμα των προβλημάτων που αυτή η αντιμετώπιση φέρει, είναι η υλοποίηση των συνόδων (sessions) στις server-side γλώσσες (11)(8). Ως γνωστόν, το πρωτόκολλο http είναι stateless, δηλαδή δεν ενδιαφέρεται για το ιστορικό των συνδέσεων ενός πελάτη με τον εξυπηρετητή. Εντούτοις οι περισσότερες εφαρμογές απαιτούν ο χρήστης να περιηγηθεί από πολλές html σελίδες και ο εξυπηρετητής να διατηρεί στοιχεία τόσο για το χρήστη όσο καιγια τις σελίδες που εκείνος επισκέφτηκε. Για να αντιμετωπιστεί το πρόβλημα και να διατηρηθεί η προς τα πίσω συμβατότητα, αναπτύχθηκε η έννοια της «συνόδου» (browsing session) ως το σύνολο των σελίδων που διαβάζει ένας χρήστης κατά την επίσκεψη του στον ιστοτόπο. Οι browsers εισήγαγαν τα cookies ως μονάδες τοπικής αποθήκευσης δεδομένων που αντιστοιχούν σε μια σύνοδο.

Στην έκδοση 2, η γλώσσα PHP δεν διέθετε κάποιο μηχανισμό για υποστήριξη συνόδου πέρα από κάποιες ρουτίνες για διαχείριση cookies (11),(1). Αυτό έκανε δυσχερή και άκομψη τη συγγραφή διαδικτυακών εφαρμογών αφού επαφιόταν στο προγραμματιστή να διαχειριστεί τα cookies μιας συνόδου. Με την υλοποίηση ρουτινών για την υποστήριξη συνόδων, η ανάπτυξη τέτοιον εφαρμογών έγινε ευκολότερη, καθώς ένα επίπεδο αφαίρεσης αναλάμβανε την αυτόματη διαχείριση (θέση και ανάγνωση) όλων των σχετικών cookies ώστε να υπάρχει συνοχή από σελίδα σε σελίδα σε μια σύνοδο. Παρ’ όλα αυτά, το γεγονός ότι κάθε σελίδα θεωρείται αυτόνομη και όχι μέρος μιας συνόδου, τόσο από το http όσο και από τη php, αναγκάζει το προγραμματιστή να συμπεριλάβει ειδικό κώδικα στις σελίδες που αναπτύσσει. Άλλες γλώσσες επιτρέπουν γρηγορότερη

ανάπτυξη διαδικτυακών εφαρμογών ακριβώς λόγω του καλύτερου συστήματος αφαίρεσης που διαθέτουν.

Αν λάβουμε υπ' όψη την εκτενή χρήση javascript και τα ασύγχρονα μοντέλα που χρησιμοποιούν οι τρέχουσες διαδικτυακές εφαρμογές γίνεται προφανής η ανάγκη για ένα απλούστερο, περισσότερο φιλικό στο προγραμματιστή περιβάλλον ανάπτυξης εφαρμογών.

Συνοψίζοντας, τα κύρια μειονεκτήματα που εντοπίζονται στο υπάρχον μοντέλο ανάπτυξης είναι τα ακόλουθα

• Έντονα ετερογενές περιβάλλον: Εμφώλευση κώδικα μιας γλώσσας στο πηγαίο κώδικα μιας άλλης. Αυτή η τεχνική δυσχεραίνει την αναγνωσιμότητα και διαχείριση του κώδικα αλλά και τον εντοπισμό λαθών.

• Δυσκολία προτυποποίησης και συστηματοποίησης της συγγραφής κώδικα: Χάριν της προς τα πίσω συμβατότητας και λόγω ασυμβατοτήτων μεταξύ τεχνολογιών, ο προγραμματιστής αφιερώνει το σημαντικότερο ποσοστό του χρόνου του για να διαχειριστεί ασυμβατότητες μεταξύ λογισμικού ή να υποστηρίξει παλαιές τεχνολογίες (πχ ανάγκη χρήσης διαφορετικού javascript κώδικα για κάθε τύπο browser)

• Μη-ειδικευμένα εργαλεία: Η χρήση γλωσσών γενικής χρήσεως που δεν έχουν σχεδιαστεί για το client-server μοντέλο, δυσχεραίνει την ανάπτυξη σχετικών εφαρμογών.

• Κατακερματισμός τεχνολογιών: Το μεγαλύτερο μέρος της προσπάθειας του προγραμματιστή δεν είναι η ανάπτυξη αυτής καθαυτής της εφαρμογής, αλλά η υποστήριξη όλων των ετερογενών προτύπων και τεχνολογιών στο κώδικά του. Αναγκάζεται να εντοπίζει και να επιλύει προβλήματα που δημιουργεί όχι ο κώδικας του, αλλά οι ασυμβατότητες των ετερογενών τεχνολογιών και η ελλιπής υποστήριξη τους από τους φυλλομετρητές (πχ διαφορετική σημειογραφία μεταξύ εκδόσεων της javascript ή διαφορετική συμπεριφορά συγκεκριμένων stylesheet tags).

3 ΥΠΑΡΧΟΥΣΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

Σε αυτή την ενότητα παρουσιάζονται οι κυριότερες τεχνολογίες και μέθοδοι για την ανάπτυξη διαδικτυακών εφαρμογών. Αναλύεται η φιλοσοφία κάθε

3.1 Server side interpreters: Php

Η Php είναι μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού δυναμικών ιστοσελίδων. Πρόκειται για διερμηνευτές υψηλής απόδοσης που συνήθως διασυνδέονται με το λογισμικό web server ως αρθρώματα, αλλά μπορούν να εκτελεστούν και ανεξάρτητα.

3.1.1 Σχεδιαστική Φιλοσοφία

Η Php ξεκίνησε ως ένα σετ εργαλείων CGI (common gateway interface) με στόχο τη συγγραφή προσωπικών ιστοσελίδων (αρχικά το όνομα php σήμαινε Personal Home Page Tool). (1)(11)Την πρώτη της έκδοση ανέπτυξε ως βοηθητικό εργαλείο ο Rasmus Ledorf. Αργότερα την διένειμε ως ανοιχτό κώδικα, ενώ παράλληλα άλλαξε το όνομα σε “Php Hypertext Preprocessor”.

Ο γλωσσικός αναλυτής (parser) υπέστη σημαντικές αλλαγές από τους Zeev Suraski και Andi Gutmans, οι οποίοι, στην έκδοση 3, άλλαξαν την αρχιτεκτονική ώστε ο κώδικας να μετατρέπεται on-the-fly σε byte-code και να εκτελείται από μια ιδεατή μηχανή (virtual machine), τη Zend Engine. Αργότερα οι Suraski και Gutmans ίδρυσαν την Zend Technologies, η οποία εκμεταλλεύεται και εμπορικά τη Zend engine (www.zend.com). Παράλληλα, η Php αναπτύσσεται στα πλαίσια λογισμικού ανοιχτού κώδικα από το Php group.

Η Php μπορεί να συνδεθεί ως άρθρωμα με τους περισσότερους web server, να διασυνδεθεί Μέσω Cgi (Common Gateway Interface) ή και να κληθεί από τη γραμμή εντολών για να εκτελέσει αυτόνομα και τοπικά ένα Php πρόγραμμα. Για τη χρήση σε περιβάλλον web server, η διασύνδεση μέσω cgi θεωρείται ασφαλέστερη.

Όπως στις περισσότερες server side γλώσσες, η php μπορεί να εντεθεί σε κώδικα html. Ο λεκτικός αναλυτής αναγνωρίζει ως πηγαίο κώδικα μόνο το κείμενο που βρίσκεται εντός των αναγνωριστικών <?php και ?> (ή συντομογραφικά <? και ?>). Εναλλακτικά μπορούν να χρησιμοποιηθούν και τα html tags <script language="php"> και </script>. Το κείμενο εκτός των php tags δεν αναγνωρίζεται από τον αναλυτή και αντιγράφεται αυτούσιο στην έξοδο (standard output). Αυτό επιτρέπει την εμφώλευση κώδικα php σε αρχεία html ή οποιαδήποτε αρχείο κειμένου.

3.2 Java & Java Server Pages

Η Java είναι μια πλήρως αντικειμενοστραφής γλώσσα προγραμματισμού με μεταφραστή, που αναπτύχθηκε από τη Sun Microsystems Inc και χρησιμοποιείται ευρέως χάρις στα cross-platform χαρακτηριστικά που διαθέτει. Η τεχνολογία Java Server Pages επιτρέπει τη χρήση της Java για τη συγγραφή δυναμικών σελίδων αλλά και ολοκληρωμένων εφαρμογών εξυπηρετητή, ομογενοποιημένων με τον παγκόσμιο ιστό (4)(12).

3.2.1 Σχεδιαστική Φιλοσοφία

Ο κώδικας java μεταφράζεται όχι για κάποια συγκεκριμένη αρχιτεκτονική υλικού, αλλά για την αρχιτεκτονική μιας ιδεατής μηχανής (Java Virtual Machine), η οποία έχει ένα σαφώς ορισμένο σύνολο χαρακτηριστικών(4). Η ιδεατή αυτή μηχανή υλοποιείται σε κάθε υποστηριζόμενη πλατφόρμα υλικού, ακολουθώντας αυστηρές συμβάσεις ως προς τους τύπους δεδομένων και τις δυνατότητες που υποστηρίζονται. Αυτό επιτρέπει στα προγράμματα που γράφονται σε java, να εκτελούνται όμοια και να βγάζουν ίδια αποτελέσματα ανεξάρτητα από την υφιστάμενη αρχιτεκτονική υλικού.

Όπως αναφέρθηκε παραπάνω, υπάρχουν και άλλες γλώσσες που αξιοποιούν τη φιλοσοφία της ιδεατής μηχανής. Η java είναι η πρώτη γλώσσα ισχυρών τύπων με καθαρά αντικειμενοστραφή

φιλοσοφία, στην οποία οι βασικοί τύποι (ακέραιοι, δεκαδικοί, πίνακες κλπ) αποθηκεύονται με σαφώς ορισμένο τρόπο ανεξαρτήτως της αρχιτεκτονικής υλικού. Για παράδειγμα, ενώ κάποιες αρχιτεκτονικές υλικού θεωρούν τον τύπο char ως 8-bit signed integer, κάποιες άλλες υλοποιούν τον ίδιο τύπο ως 8-bit unsigned integer. Η java Virtual Machine υλοποιεί τον τύπο char με ενιαίο και ομογενή τρόπο σε όλες τις πλατφόρμες. Έτσι, ο μεταγλωττισμένος κώδικας java μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα υποστηρίζεται από τη java virtual machine.

Δυστυχώς η ενιαία αυτή αντιμετώπιση φέρει προβλήματα αποδοτικότητας, καθώς σε περίπτωση που κάποια αρχιτεκτονική δεν υποστηρίζει εγγενώς ένα τύπο δεδομένων, όλες οι προσβάσεις στον τύπο αυτό γίνονται μέσω λογισμικού. Η σημαντικά μειωμένη ταχύτητα του κώδικα java αποτέλεσε και τη κυριότερη τροχοπέδη στη διάδοσή της. Παρ όλα αυτά, ένα σετ τεχνολογιών όπως οι Just In Time Compilers επιτρέπουν βελτιστοποίηση της απόδοσης ενός προγράμματος java.

Σε αντίθεση με τη φιλοσοφία της Java, άλλες γλώσσες υλοποιούν Virtual Machines που χρησιμοποιούν τους εκάστοτε εγγενείς τύπους της αρχιτεκτονικής. Δηλαδή ένας τύπος char μπορεί να σημαίνει είτε ένα 8 bit unsigned integer, είτε ένα 8 bit signed integer στο μεταγλωττισμένο κώδικα, ανάλογα με την υφιστάμενη αρχιτεκτονική. Η φιλοσοφία αυτή

συνεπάγεται ότι ο ίδιος πηγαίος κώδικας μπορεί να μεταφραστεί σε διαφορετικές πλατφόρμες, όμως δεν υπάρχουν εγγυήσεις ότι θα εκτελεστεί με τον ίδιο τρόπο.(8) Πάντως ένα μεγάλο μέρος του κώδικα θα έχει κοινή συμπεριφορά σε ανεξάρτητες πλατφόρμες. Προφανώς, επειδή γίνεται χρήση εγγενών τύπων, η απόδοση της ιδεατής μηχανής είναι βελτιωμένη.

3.2.2 Java Server Pages

Λόγω της ανεξαρτησίας της από το υλικό, η Java αποτελεί ιδανική επιλογή για ανάπτυξη επεκτάσιμων εφαρμογών. Εντούτοις σε περιβάλλον παγκόσμιου ιστού η Java αυτή καθ’ αυτή πάσχει από δύο προβλήματα:

• Η σύνταξή της δεν είναι ευέλικτη, συνεπώς δε συνίσταται για απλές εργασίες. Προφανώς δεν είναι εφικτή η εμφώλευση κώδικα HTML σε καθαρό κώδικα java στα πρότυπα της php

• Για τη εκτέλεση του κώδικα απαιτείται πρώτα η μετάφραση του και αργότερα η εκτέλεση του μέσω της ιδεατής μηχανής. Δεδομένου ότι η ανάπτυξη κώδικα για τον Παγκόσμιο ιστό γίνεται συνήθως σε κάποιο απομακρυσμένο σταθμό εργασίας και ο κώδικας πρέπει να μεταφερθεί και εκτελεστεί στον εξυπηρετητή, μια κομψότερη λύση όπως αυτή της php είναι απαραίτητη.

Για την επίλυση των παραπάνω προβλημάτων, η Sun επέκτεινε το σύνολο δυνατοτήτων της Java και ανέπτυξε μια βοηθητική γλώσσα, την JSP (13)(1).

Η JSP επιτρέπει στο προγραμματιστή να εμφωλεύσει κώδικα java σε κώδικα HTML με τη χρήση μια σύνταξης ανάλογης με αυτών της php και asp. Επίσης χρησιμοποιεί ένα σύνολο XML tags για να ορίσει συγκεκριμένες λειτουργίες. Τα tags αυτά μπορούν να διασυνδεθούν με τάξεις java οι οποίες εκτελούνται σε χαμηλότερα στρώματα του εξυπηρετητή (bacend process, δηλαδή πίσω από τον web server). Παράλληλα, η JSP δίνει ένα σύνολο μηχανισμών για τη διασύνδεση του java τμήματος ενός server με το web τμήμα του, κάποιοι από τους οποίους αναλύονται παρακάτω.

Σε αντίθεση με τη php, η jsp χρησιμοποιεί μια πληθώρα από διαφορετικά jsp tags, ενώ επιτρέπει στο χρήστη να δηλώσει και νέες τάξεις tags αν αυτό είναι αναγκαίο. Πρόσθετα tags (<jsp: tags) υλοποιούν άλλες λειτουργίες web server, όπως η ενσωμάτωση αντικειμένων και η εκτέλεση ειδικού κώδικα. Τέλος, ενδιαφέρον είναι το plugin tag, το οποίο χρησιμοποιείται για να ενσωματώσει αντικείμενα (applets) σε μια σελίδα, υλοποιώντας κατ ουσία τη λειτουργικότητα του <object> tag της html, αλλά με σύνταξη κατάλληλη για το φυλλομετρητή του τρέχοντος χρήστη (τον οποίο γνωρίζει ο εξυπηρετητής):

Τέλος, μέσω των JSP tag libraries, οι προγραμματιστές μπορούν να ορίσουν εξειδικευμένα tags και να τα ενθέσουν σε HTML κώδικα. Με τον τρόπο αυτό μπορούν εύκολα να διασυνδέσουν το τμήμα αλληλεπίδρασης με το χρήστη της εφαρμογής με το εσωτερικό σύστημα που παρέχει πρόσβαση και επεξεργασία στα εσωτερικά δεδομένα του εξυπηρετητή. Τα tags αυτά μπορούν να υλοποιηθούν σε καθαρή java με κώδικα.

Ο κώδικας java που περιέχεται σε μια jsp σελίδα μεταγλωττίζεται αυτόματα από ένα ειδικό άρθρωμα του εξυπηρετητή (το οποίο στην ουσία επεκτείνει τις δυνατότητες του java compiler) και αυτόματα αποθηκεύεται προς άμεση εκτέλεση. Σε περίπτωση που ο πηγαίος jsp κώδικας υποστεί αλλαγές, θα μεταγλωττιστεί αυτόματα από τον εξυπηρετητή. Με αυτό το τρόπο επιτρέπεται η εύκολη συγγραφή δυναμικών ιστοσελίδων, οι οποίες έχουν τη δυνατότητα να συνεργαστούν με το υπόλοιπο σύνολο λειτουργικότητας που παρέχει ένας εξυπηρετητής.

Η προαναφερθείσα δυνατότητα δίνει ένα σημαντικό πλεονέκτημα στο πρότυπο ανάπτυξης της JSP, καθώς κάνει εφικτό το διαχωρισμού της εφαρμογής σε δύο μεγάλα τμήματα: αυτό της παρουσίασης στο χρήστη (web – front end) και σε αυτό της λογικής επεξεργασίας των δεδομένων (back end). Το backend μπορεί να διαχωριστεί περεταίρω σε μοντέλο δεδομένων (model) και λογικής (control er). Έτσι οδηγούμαστε στο μοντέλο ανάπτυξης model-view-controler, το οποίο διαχωρίζει την εφαρμογή σε τρία τμήματα:

1. Model: Το μοντέλο δεδομένων της εφαρμογής. Η Τα μη-εφήμερα δεδομένα αποθηκεύονται σύμφωνα με τις απαιτήσεις και τις δομές που ορίζει το μοντέλο δεδομένων

2. Controler: Το τμήμα διαχείρισης και επεξεργασίας δεδομένων της εφαρμογής. Η επεξεργασία γίνεται με java servlets ή με ένα σύνολο από JSP σελίδες.

3. View: Το τμήμα αλληλεπίδρασης με το χρήστη. Συγκεκριμένες σελίδες (JSP) χρησιμοποιούν τα JSP tags είτε για να τυπώσουν δεδομένα στην οθόνη είτε για να καλέσουν μεθόδους που θα αλλάξουν ή επεξεργαστούν τα δεδομένα αυτά. Με το διαχωρισμό αυτό ο κώδικας γίνεται ευκολότερα διαχειρίσιμος και επεκτάσιμος. Επίσης, ισχυροποιείται η ασφάλεια του κώδικα αφού είναι δυνατόν να ελεγχθεί η πρόσβαση στα εσωτερικά δεδομένα.

3.3 Asp & Asp.net

Η γλώσσα Asp είναι μια μηχανή συγγραφής δυναμικών ιστοσελίδων και εφαρμογών ιστού. ASP σημαίνει Active Server Pages (8). Αναπτύσσεται από τη Microsoft, ενώ η πρώτη έκδοση της ονομαζόταν dbWeb και είχε αναπτυχθεί από την Aspect Software Engineering(1). Το κύριο χαρακτηριστικό της Asp είναι η στενή του συνεργασία με το λογισμικό web server της Microsoft, IIS (Internet Information Services). Η Asp μπορεί να αναγνωρίσει κώδικα σε οποιαδήποτε γλώσσα είναι

συμβατή με το «Microsoft Active Scripting» στάνταρ. Συνηθέστερα χρησιμοποιούμενη είναι η VBScript. Η επιλογή της γλώσσας που χρησιμοποιείται για την ερμηνεία του κώδικα γίνεται μέσω της εντολής @language ή μέσω του tag <script language=”language” runat=”server”>.

Άλλες συμβατές γλώσσες περιλαμβάνουν τη Jscript (μια υλοποίηση ECMAscript από τη Microsoft) και την PerlScript. Ας τονιστεί εδώ ότι εκτός της Asp, η Microsoft αναπτύσσει και τη γλώσσα asp.net, η οποία έχει ουσιαστικές διαφορές με την asp: βασίζεται στο .Net framework της εταιρίας και χρησιμοποιεί just-in-time compilation τεχνικές. Αρχεία κώδικα της απλής Asp έχουν κατάληξη “.asp”, ενώ αρχεία asp.net έχουν συνήθως κατάληξη ”aspx” .

3.3.1 Σχεδιαστική Φιλοσοφία και Συντακτικό

Η δυναμική δημιουργία περιεχομένου στη ASP γίνεται με τη χρήση αντικειμένων τα οποία υλοποιούν συγκεκριμένες λειτουργίες. Στην έκδοση 2, η ASP διαθέτει 6 τέτοια αντικείμενα: Application, ASPError, Request, Response, Server και Session. Κάθε αντικείμενο υλοποιεί κάποια βασική web-server λειτουργικότητα. Για παράδειγμα, το αντικείμενο Session αποθηκεύει δεδομένα από σελίδα σε σελίδα με τη χρήση των Browser Cookies.

Όμοια με τη php, στη κλασική asp επιτρέπεται η εμφώλευση html κώδικα με τη χρήση των αναγνωριστικών <% και %>. Το αναγνωριστικό <%= χρησιμοποιείται αντί του <% για να καταδείξει μια «έκφραση» (expression). Επίσης, όμοια με την php, μέθοδοι μπορούν να κληθούν και χωρίς τη χρήση παρενθέσεων.

Συναρτήσεις ορίζονται στα πρότυπα της εκάστοτε ενεργής γλώσσας, στη περίπτωση της VBScript με τα αναγνωριστικά function και end function, ενώ για σχολιασμό χρησιμοποιείται ο χαρακτήρας “ ' ”:

Ένα σημαντικό μέρος της Asp λειτουργικότητας περιλαμβάνεται σε βιβλιοθήκες αντικειμένων, οι οποίες απλοποιούν επαναληπτικές διαδικασίες, όπως για παράδειγμα η παρουσίαση λίστας στοιχείων από μια βάση δεδομένων.

3.3.2 Το .Net framework και η καινοτομική αρχιτεκτονική της asp.net

Με την έλευση του .Net framework, η Microsoft προσπάθησε να υποστηρίξει την ανάπτυξη πολύπλοκων web-εφαρμογών μέσα από μια ολοκληρωμένη πλατφόρμα. Έτσι ξανασχεδίασε τις τεχνολογίες τις για τη δημιουργία δυναμικών ιστοσελίδων, με στόχο τη καλύτερη διασύνδεση με το .Net framework, τη καλύτερη κλιμάκωση και την ευκολότερη συντήρηση. Έτσι παρουσιάστηκε η asp.net (14). Όπως και με τη απλή asp, η asp.net μπορεί να διασυνδέσει κώδικα οποιασδήποτε γλώσσας με HTML στοιχεία, όμως συνίσταται η χρήση της γλώσσας C# για ανάπτυξη web-εφαρμογών.

Το πέρασμα στη νέα πλατφόρμα .net, σηματοδοτεί και μια καινοτομική φιλοσοφία στην ανάπτυξη σελίδων. Σε αντίθεση με παλαιότερες πρακτικές, η asp.net προσπαθεί να εξομοιώσει τον τρόπο λειτουργίας και προγραμματισμό του γραφικού περιβάλλοντος των windows στη συγγραφή σελίδων. Παρέχει μια πλήρη σειρά από γραφικά αντικείμενα (κουμπιά, λίστες, πίνακες, data views κλπ) με τα οποία ο προγραμματιστής μπορεί να αναπτύξει σελίδες, με τρόπο που ομοιάζει περισσότερο με ανάπτυξη εφαρμογής παρά σελίδας. Ενώ όμως σε μια παραθυρική εφαρμογή τα

διάφορα γραφικά στοιχεία σχεδιάζονται χρησιμοποιώντας κλήσεις γραφικών, στην asp.net τα εκάστοτε στοιχεία σχεδιάζονται παράγοντας HTML και javascript κώδικα. Ένα σύστημα Μηνυμάτων και Γεγονότων (messages & events) χρησιμοποιείται για τη διασύνδεση των στοιχείων αυτών με κώδικα asp.net. Ουσιαστικά γίνεται προσπάθεια να δημιουργηθεί ένα περιβάλλον ανταλλαγής μηνυμάτων μέσα από το (stateless) html πρωτόκολλο. Με τον τρόπο αυτό απλοποιείται σημαντικά η δημιουργία πολύπλοκων web-εφαρμογών (1).

Μια σημαντική διαφορά της Asp.net σε σχέση με τη απλή asp είναι η χρήση μεταγλωττισμένου κώδικα. Για να διατηρηθεί η ευχρηστία ανάπτυξης κώδικα, ο εξυπηρετητής (Microsoft IIS) μεταφράζει αυτόματα (just in time) τον κώδικα ενός συνόλου σελίδων asp.net (οι οποίες έχουν συνήθως κατάληξη .aspx) και παράγει ένα .Net module το οποίο **τρέχει εγγενώς** στην αρχιτεκτονική (windows server) (8)(14). Αυτό συνεπάγεται μια σημαντική καθυστέρηση κατά τη μεταγλώττιση της σελίδας, όταν υπάρχουν αλλαγές. Καθώς όμως ο κώδικας που παράγεται είναι εγγενής για την αρχιτεκτονική, οι μετέπειτα επισκέψεις στην ίδια σελίδα επιταχύνονται εντυπωσιακά. Ο εξυπηρετητής γνωρίζει αν ο χρήστης έκανε αλλαγές σε κάποιο αρχείο κώδικα και το μεταγλωττίζει αυτόματα. Έτσι η επεξεργασία κώδικα πάνω στο server απλοποιείται σημαντικά. Ένα πρόβλημα που δημιουργείται λόγω της δημιουργίας εγγενούς κώδικα είναι η ασυμβατότητα με non-microsoft αρχιτεκτονικές εξυπηρετητή.

3.4 Javascript

Η javascript είναι η βασική γλώσσα προγραμματισμού της ιστοσελίδας από τη άκρη του χρήστη. Χρησιμοποιείται για να περιγράψει διαδικασίες που πρέπει να γίνουν στο φυλλομετρητή (browser) (8).

3.4.1 Σχεδιαστική φιλοσοφία

Στοιχεία javascript εμφωλεύονται σε html στοιχεία μέσω του tag <script>. Τα στοιχεία αυτά, τα οποία μπορούν να περιλαμβάνουν κώδικα in-line ή να αναφέρονται σε αρχεία κώδικα, ανιχνεύονται από τον φυλλομετρητή, ο οποίος και διερμηνεύει τον κώδικα σε πραγματικό χρόνο.

Η javascript συνήθως χρησιμοποιείται για τη δυναμική διαχείριση των διαφόρων στοιχείων μιας ιστοσελίδας σε πραγματικό χρόνο και χωρίς να απαιτείται αλληλεπίδραση πελάτη-εξυπηρετητή. Για παράδειγμα, μέσω javascript ο προγραμματιστής μπορεί να μετακινήσει ομάδες στοιχείων, να κρύψει στοιχεία ή να αλλάξει τα περιεχόμενα τους δυναμικά. Για να γίνει αυτό, πρέπει πρώτα να αποκτήσει πρόσβαση στα στοιχεία αυτά (δηλαδή στο Document Object Model), χρησιμοποιώντας τις μεθόδους που παρέχει ο φυλλομετρητής.

Δυστυχώς το πρότυπο της Javascript δεν επιβάλλει κάποια κοινή ονοματολογία για τη πρόσβαση στα στοιχεία μιας ιστοσελίδας. Έτσι κάθε φυλλομετρητής παρέχει τις δικές του μεθόδους για τη πρόσβαση στα στοιχεία αυτά. Αυτός είναι ο κύριος λόγος που η ανάπτυξη κώδικα javascript είναι δύσκολη και επίπονη διαδικασία: απαιτείται να αναπτυχθεί ειδικός κώδικας για την υποστήριξη των διαφορετικών φυλλομετρητών. Αυτός εξ άλλου είναι και ο λόγος που η χρήση της

javascript ήταν περιορισμένη μέχρι την έλευση κάποιων wrapper-βιβλιοθηκών και τη βελτίωση της υποστήριξης της από τους φυλλομετρητές.

Για την επίλυση του παραπάνω προβλήματος, η επιτροπή προτύπων του παγκόσμιου ιστού έχει προτείνει ένα κοινό πρότυπο, το HTML DOM (html Document Object Model), το οποίο φαίνεται να κερδίζει έδαφος και να υποστηρίζεται από όλο και περισσότερους φυλλομετρητές.

3.4.2 Σύνταξη και χαρακτηριστικά

Κάθε φυλλομετρητής υλοποιεί τον δικό του διερμηνευτή για javascript (συνήθως καλείται javascript engine), ο οποίος μάλιστα είναι και καθοριστικής σημασίας για την απόδοση του φυλλομετρητή όσον αφορά τη συμβατότητα, σταθερότητα, ταχύτητα και ασφάλεια.

Συνήθως ο κώδικας javascript διερμηνεύεται, δε μεταγλωττίζεται. Εντούτοις για να βελτιωθεί η απόδοση, υπάρχουν φυλλομετρητές που ακολουθούν μεθόδους just-in-time μεταγλώττισης, όπως ο Google Chrome με μια καινοτομική javascript engine (οι οποία ονομάζεται V8 και είναι ανοιχτού κώδικα)

Η σύνταξη javascript είναι κοινή μεταξύ φυλλομετρητών, παρόλο που υπάρχουν διαφορές στην ονοματολογία των βασικών μεθόδων. Η σύνταξη θυμίζει php, με τη διαφορά ότι δεν χρησιμοποιείται το χαρακτηριστικό ‘$’ στα ονόματα των μεταβλητών. Πρόκειται για μια γλώσσα χαλαρών τύπων όπου οι μεταβλητές μπορούν να αλλάξουν τύπο δυναμικά. Αλφαριθμητικά μεταβλητού μεγέθους υποστηρίζονται εγγενώς. Για τη δήλωση μιας μεταβλητής χρησιμοποιείται η λέξη-κλειδί “var” ενώ για τον ορισμό συνάρτησης χρησιμοποιείται η λέξη “function”, με ένα συντακτικό όμοιο της php (δεν απαιτείται να δηλωθεί ο τύπος του αποτελέσματος). Δεν υπάρχουν δείκτες, ενώ υποστηρίζονται τάξεις.

Τις περισσότερες φορές ζητείται η χρήση της javascript όταν ο χρήστης πράξει μια συγκεκριμένη ενέργεια (πχ πατήσει κάποιο πλήκτρο ή μετακινήσει κάποιο στοιχείο της σελίδας). Για το λόγο αυτό τα διάφορα HTML στοιχεία ορίζουν συγκεκριμένα συμβάντα, και δέχονται ως παράμετρο μια σειρά από javascript εντολές. Στη περίπτωση αυτή, ο κώδικας javascript παρέχεται in-line στον ορισμό του html-στοιχείου, και συγκεκριμένα ως παράμετρος κειμένου μέσα στο HTML tag του στοιχείου. Για παράδειγμα, ένα html button element ορίζει ένα “onClick” event, το οποίο δέχεται javascript κώδικα κατά τη σύνταξη:

<button onclick=”javascriptfunction();”> button text </button>

Στο πεδίο onclick μπορεί να εντεθεί πιο πολύπλοκος κώδικας, όπως στο παρακάτω παράδειγμα, όπου τα στοιχεία

ενός πεδίου αντιγράφονται σε ένα άλλο:

<html><body>

A: <input type="text" id="field1" value="Hello World!">

B: <input type="text" id="field2">

Click the button below to copy the content of field A to field B.

<button onclick="document.getElementById('field2').value=

document.getElementById('field1').value">Copy Text</button>

</body></html>

Η δυνατότητα αυτή απλοποιεί σημαντικά τον προγραμματισμό σε javascript, αλλά καθώς ενοποιεί τον html με τον javascript κώδικα, μπορεί να δημιουργήσει προβλήματα: αφενός είναι δυσκολότερο για το προγραμματιστή να διακρίνει τα διάφορα στοιχεία javascript κώδικα όταν κάνει αποσφαλμάτωση, αφετέρου εισάγει σοβαρά ζητήματα ασφαλείας: Για παράδειγμα, έστω ένα διαδικτυακό forum, στο οποίο οι χρήστες μπορούν να εισάγουν κείμενο σε μορφή html. Έστω λοιπόν ότι ο χρήστης ενός forum εισάγει το παρακάτω html-formatted μήνυμα:

Check this picture:

** picture **

Το μήνυμα του θα αποθηκευτεί σε μια βάση δεδομένων ως απλό κείμενο. Όταν όμως κάποιος χρήστης προσπαθήσει να το διαβάσει, αυτό το κείμενο **θα εντεθεί σε html κώδικα** και ο φυλλομετρητής του θα θεωρήσει το <a> πεδίο ως μέρος του κώδικα αυτού. Όταν ανυποψίαστοι χρήστες ακολουθήσουν το σύνδεσμο στην υποτιθέμενη εικόνα, θα εκτελεστεί στους υπολογιστές τους ο javascript κώδικας του κακόβουλου χρήστη! Μάλιστα, δεν υπάρχει τρόπος, εφόσον πρόκειται για html στοιχείο, ο φυλλομετρητής να γνωρίζει ότι το onclick event προέρχεται από κάποιον κακόβουλο χρήστη και όχι από το (υποτιθέμενα) ασφαλές forum. Για το λόγο αυτό πρέπει ο προγραμματιστής του forum να φιλτράρει κατάλληλα κάθε μήνυμα για να διαγράψει κακόβουλα javascript στοιχεία. Εντούτοις, η διαδικασία αυτή είναι δύσκολη, καθώς δεν υπάρχει κάποια ιδιαίτερη σημειογραφία για τον javascript κώδικα (δηλαδή δεν υπάρχει κάποιο αναγνωριστικό του τύπου <?php ή <%= που να σηματοδοτεί τον javascript κώδικα)

3.5 Flash & Flex

Η τεχνολογία Flash αναπτύχθηκε από τη Macromedia, η οποία αργότερα εξαγοράστηκε από την Adobe. Πρόκειται για μια τεχνολογία ανάπτυξης δυναμικών ιστοσελίδων που χρησιμοποιείται ευρύτατα στο διαδίκτυο, χάρις στην υποστήριξη πολυμεσικών χαρακτηριστικών (15).

Το Flash αποτελείται από δύο βασικά στοιχεία, τα οποία αλληλεπιδρούν έντονα μεταξύ τους.

1. Μια μηχανή ορισμού, προβολής και κίνησης διανυσματικών γραφικών με προχωρημένες πολυμεσικές δυνατότητες (rendering engine).

2. Μια γλώσσα προγραμματισμού, την actionscript, η οποία μπορεί να διαχειριστεί πολυμεσικά αντικείμενα και να υποστηρίξει την ανάπτυξη ολοκληρωμένων εφαρμογών. Αρχικά το

flash είχε σχεδιαστεί ως περιβάλλον ανάπτυξης, σχεδιασμού και παρουσίασης δισδιάστατων γραφικών με κίνηση και ήχο. Αργότερα επεκτάθηκε με την εισαγωγή ολοκληρωμένων στοιχείων GUI (κουμπιά, λίστες κλπ) αλλά και με τη δυνατότητα αναπαραγωγής video. Επίσης, βελτιώθηκε σημαντικά η δυνατότητα προγραμματισμού της γενικής λογικής αλλά και των γραφικών στοιχείων, ενώ η ενσωματωμένη γλώσσα προγραμματισμού βελτιστοποιήθηκε και έγινε πιο αντικειμενοστραφής. Αυτή τη στιγμή το flash video είναι ο πιο ευρέως χρησιμοποιούμενος τρόπος μετάδοσης video, σε πραγματικό χρόνο, στο διαδίκτυο, ενώ το flash χρησιμοποιείται σε ένα μεγάλο αριθμό από ιστοσελίδες ως βασικό στοιχείο που βελτιώνει τη διάδραση με το χρήστη.

Η τεχνολογία αποτελείται από μια μικρή εφαρμογή (player) που αναλαμβάνει να προβάλλει κατάλληλα επεξεργασμένο, μεταγλωττισμένο περιεχόμενο (το οποίο μπορεί να περιέχει πολυμεσικό υλικό ή και κώδικα), και από μια εφαρμογή ανάπτυξης περιεχομένου, η οποία ενοποιεί με σχετική κομψότητα τις διαδικασίες επεξεργασίας κινούμενων γραφικών και ανάπτυξης προγραμμάτων.

Τα αρχεία flash που προορίζονται για προβολή σε ιστοσελίδα, καλούνται compiled (μεταγλωττισμένα) αρχεία και συνήθως έχουν κατάληξη ".swf". Τα αρχεία που χρησιμοποιούνται από το σχεδιαστή και προγραμματιστή για τη δημιουργία περιεχομένου συνήθως έχουν κατάληξη ".fla". Στο εξής θα καλούμε τα αρχεία αυτά «πηγαία αρχεία»

Ένα .fla αρχείο περιλαμβάνει διανυσματικά δεδομένα γραφικών, εικόνες και ήχους, τα οποία και επεξεργάζονται ώστε να καθοριστούν οι οπτικές και κινητικές τους ιδιότητες από την εφαρμογή ανάπτυξης περιεχομένου. Η εφαρμογή προβολής περιεχομένου (player) δεν μπορεί να αναγνωρίσει μη-μεταγλωττισμένα αρχεία.

Τα μεταγλωττισμένα flash αρχεία μπορούν να εισαχθούν σε μια ιστοσελίδα και να παρουσιασθούν στο χρήστη, εφόσον αυτός έχει εγκαταστήσει στο σύστημά του την εφαρμογή προβολής (player). Δηλαδή το περιεχόμενο flash **δεν υποστηρίζεται εγγενώς** από το φυλλομετρητή και δεν παρεμβάλει html στοιχεία στο κώδικα. Αντίθετα, εισάγεται στην ιστοσελίδα ως **ειδικό αντικείμενο**, με τρόπο παρόμοιο με εκείνον που εισάγονται εικόνες ή ήχοι. Το ειδικό αυτό αντικείμενο ανιχνεύεται από τον υπολογιστή του χρήστη, ο οποίος καλείται να εγκαταστήσει το λογισμικό διαχείρισης τέτοιων αντικειμένων.

3.5.1 Γραφικές και πολυμεσικές δυνατότητες

Η τεχνολογία επιτρέπει στο χρήστη να σχεδιάσει εκ του μηδενός πολύπλοκες πολυμεσικές παρουσιάσεις με γραφικά που μεταβάλλονται στο χρόνο. Όλα τα πολυμεσικά δεδομένα που προβάλλονται στο χρήστη τοποθετούνται και διαχειρίζονται σε μία ειδική αφαιρετική δομή η οποία καλείται «σκηνή». Η σκηνή αντιστοιχεί με **έναν καμβά** στον οποίο ένας ζωγράφος θα δημιουργήσει το έργο του. Ο σχεδιαστής μπορεί να κατασκευάσει μια αλληλουχία από σκηνές, η κάθε μία από τις οποίες περιέχουν ένα σύνολο μεταβλητών στο χρόνο, διαδραστικών και προγραμματιζόμενων στοιχείων (μπορούν να περιέχουν και άλλες σκηνές) και να τις προβάλλει στο χρήστη κατά βούληση, μεταγλωττίζοντάς τες.

Το περιβάλλον επεξεργασίας, το οποίο είναι αντίστοιχο με αυτό ενός programming IDE (integrated development environment) επιτρέπει στους χρήστες να επεξεργαστούν όλα τα δεδομένα και ακολούθως να παράγουν τα compiled flash αρχεία που θα συμπεριληφθούν στις ιστοσελίδες τους. Οι χρήστες του περιβάλλοντος επεξεργασίας έχουν μεταξύ άλλων τις παρακάτω δυνατότητες:

• Να σχεδιάσουν, χρησιμοποιώντας τα ενσωματωμένα εργαλεία του συστήματος, οποιοδήποτε διανυσματικό σχέδιο (γραφικό)

• Να συμπεριλάβουν στη σκηνή εικόνες, τις οποίες μπορούν ακολούθως να επεξεργαστούν

• Να δημιουργήσουν συλλογές αντικειμένων

• Να ομαδοποιήσουν αντικείμενα και να δημιουργήσουν μικρότερες υπο-σκηνές, οι οποίες λειτουργούν ως «ηθοποιοί» στην ευρύτερη σκηνή

• Να κινήσουν αντικείμενα στο χώρο και να αλλάξουν δυναμικά τις ιδιότητες τους στο χρόνο.

• Να ορίσουν εξωτερικές πηγές δεδομένων στο διαδίκτυο και να τις συμπεριλάβουν δυναμικά στη σκηνή τους

• Να ορίσουν βασικές ιδιότητες των διαδραστικών στοιχείων και να ορίσουν πλήρως τις οπτικές τους ιδιότητες

Οι παραπάνω ιδιότητες γίνονται με εντελώς γραφικά μέσα, χωρίς να απαιτείται προγραμματισμός. Εντούτοις ένα σύνολο από δυνατότητες που παρέχει το flash απαιτούν την εισαγωγή και χρήση κώδικα ώστε να αξιοποιηθούν στο έπακρο.

3.5.2 Ο προγραμματισμός στο flash

Παράλληλα με τα γραφικά, το flash διαθέτει και μια πολύ ανεπτυγμένη γλώσσα προγραμματισμού, την actionscript, η οποία εκτός των άλλων μπορεί να αλληλεπιδράσει και επεξεργαστεί όλα τα γραφικά αντικείμενα της σκηνής (αντίστοιχα με τη javascript σε μια ιστοσελίδα) αλλά και να δημιουργήσει δυναμικά νέα αντικείμενα. Το περιβάλλον ανάπτυξης ενσωματώνει δυνατότητες ανάπτυξης κώδικα σε όλα τα εργαλεία του. Με τον τρόπο αυτό η συγγραφή actionscript κώδικα απλοποιείται: Σχεδιαστές μπορούν να ενσωματώσουν κώδικα actionscript τόσο σε πολυμεσικά στοιχεία (ορίζοντας δυναμικά τη συμπεριφορά τους) όσο και σε συγκεκριμένα γεγονότα ή σημεία στο χώρο και στο χρόνο.

Όμοια με την Html και τη javascript, η ένθεση κώδικα στο περιεχόμενο αφενός απλοποιεί την ανάπτυξη μικρών εφαρμογών, αφετέρου κάνει τη διαχείριση του κώδικα πιο σύνθετη. Για το λόγο αυτό επιτρέπονται τεχνικές “code-behind” δηλαδή η σύζευξη στοιχείων με κώδικα actionscript ο οποίος υπάρχει σε εξωτερικά αρχεία κειμένου. Ο εξωτερικός κώδικας διαβάζεται όταν παράγεται το μεταγλωττισμένο flash αρχείο και δε μπορεί να μεταγλωττιστεί σε

χρόνο εκτέλεσης (run time) αλλά μόνο σε χρόνο μεταγλώττισης. Συνεπώς, όταν ο προγραμματιστής αλλάξει οποιοδήποτε στοιχείο του κώδικα, είτε εσωτερικά στο πηγαίο flash αρχείο είτε σε κάποιο εξωτερικό αρχείο κώδικα, απαιτείται να μεταγλωττίσει ξανά την εφαρμογή flash και να παράγει ένα μεταγλωττισμένο flash αρχείο(16)(15).

4 ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΩΝ ΔΕΔΟΜΕΝΩΝ

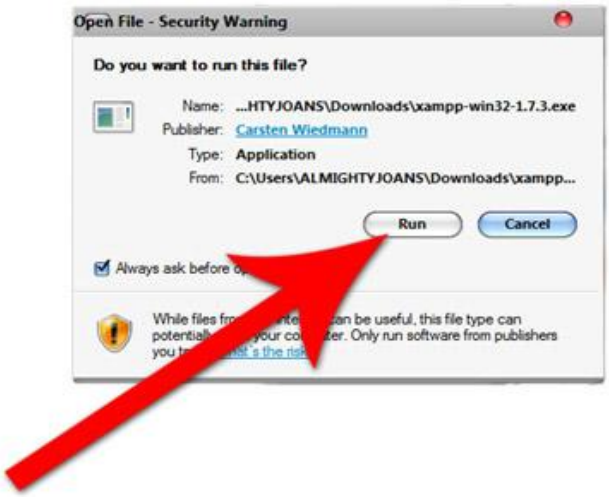
4.1 XAMPP

Το XAMPP είναι ένα πακέτο προγραμμάτων λογισμικού ανοιχτού κώδικα το όποιο περιέχει τον εξυπηρετητή σελίδων Apache, τη βάση δεδομένων MySQL και ένα διερμηνέα για τις γλώσσες προγραμματισμού PHP και Perl.

4.1.1 Εγκατάσταση XAMPP

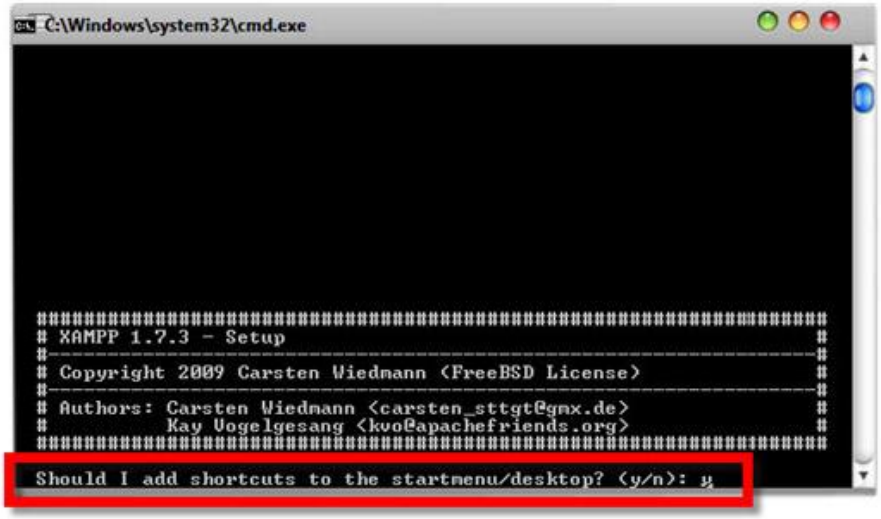
Για την εγκατάσταση του πακέτου προγραμμάτων λογισμικού ανοιχτού κώδικα XAMPP ακολουθήσαμε τα έξης βήματα:

1. Εισαγωγή στο πρόγραμμα περιήγησης τον σύνδεσμο http://www.apachefriends.org/en/xampp-windows.html
2. Λήψη του εργαλείου
3. Εγκατάσταση του xampp



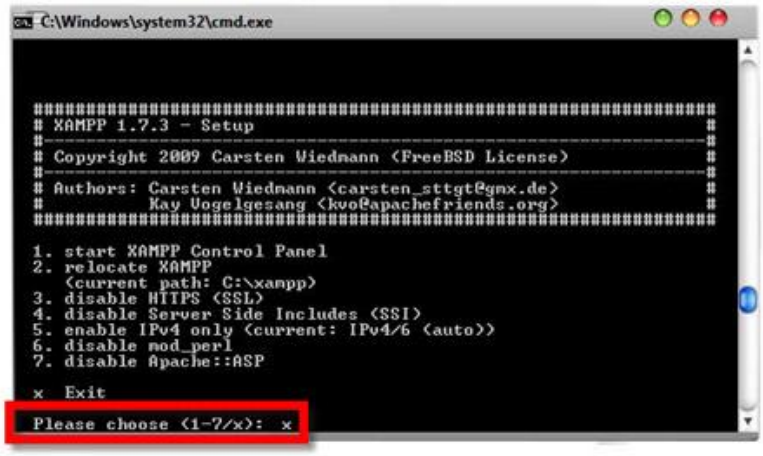
Εικόνα 1:Εγκατάσταση xampp

4. Αποδεχόμαστε τις προεπιλεγμένες ρυθμίσεις. Άπλα πατάμε το κουμπί enter.



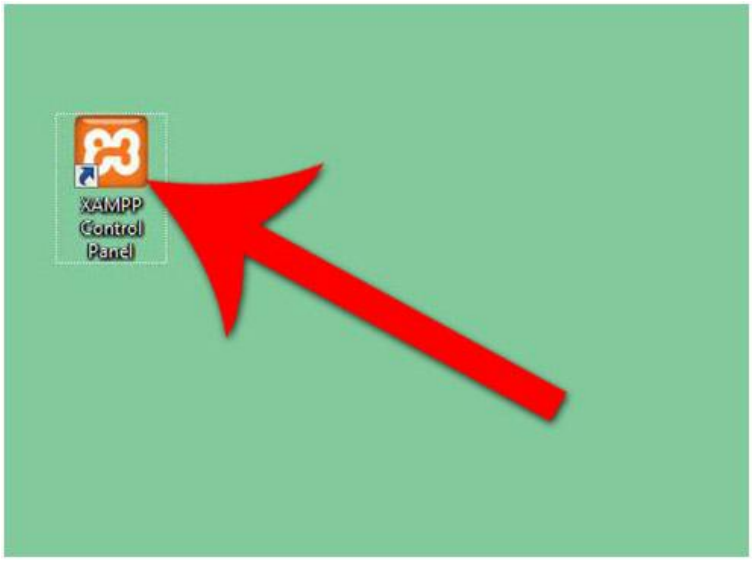
Εικόνα 2: Ρυθμίσεις xampp

5. Όταν θα έχει ολοκληρωθεί η εγκατάσταση άπλα κλείνουμε το παράθυρο.



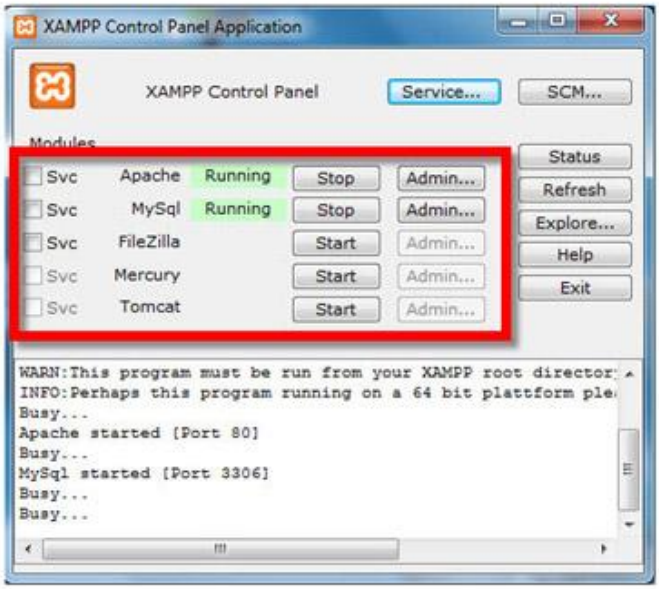
Εικόνα 3:Ολοκλήρωση Ρυθμίσεων xampp

6. Εκκίνηση του XAMPP control panel



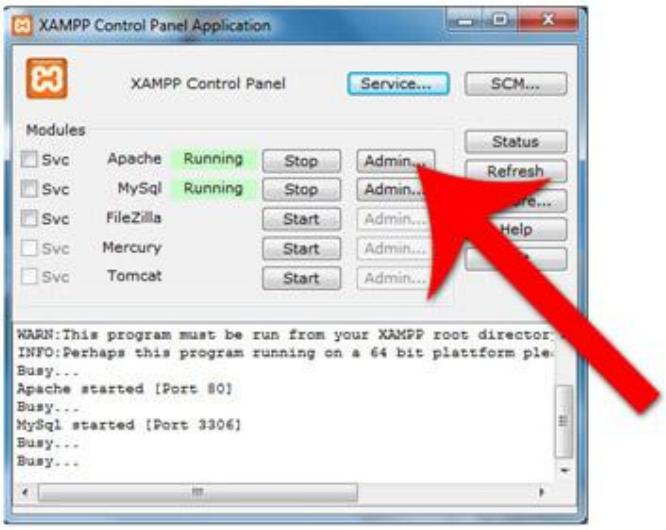
Εικόνα 4:Εκκίνηση xampp Control Panel

7. Εκκίνηση του Apache και MySQL.



Εικόνα 5:Εκκίνηση Apache και Mysql

8. Επαληθεύουμε την εγκατάσταση του Apache με το να επιλέξουμε τον σύνδεσμο του διαχειριστή του Apache από τον πίνακα ελέγχου.



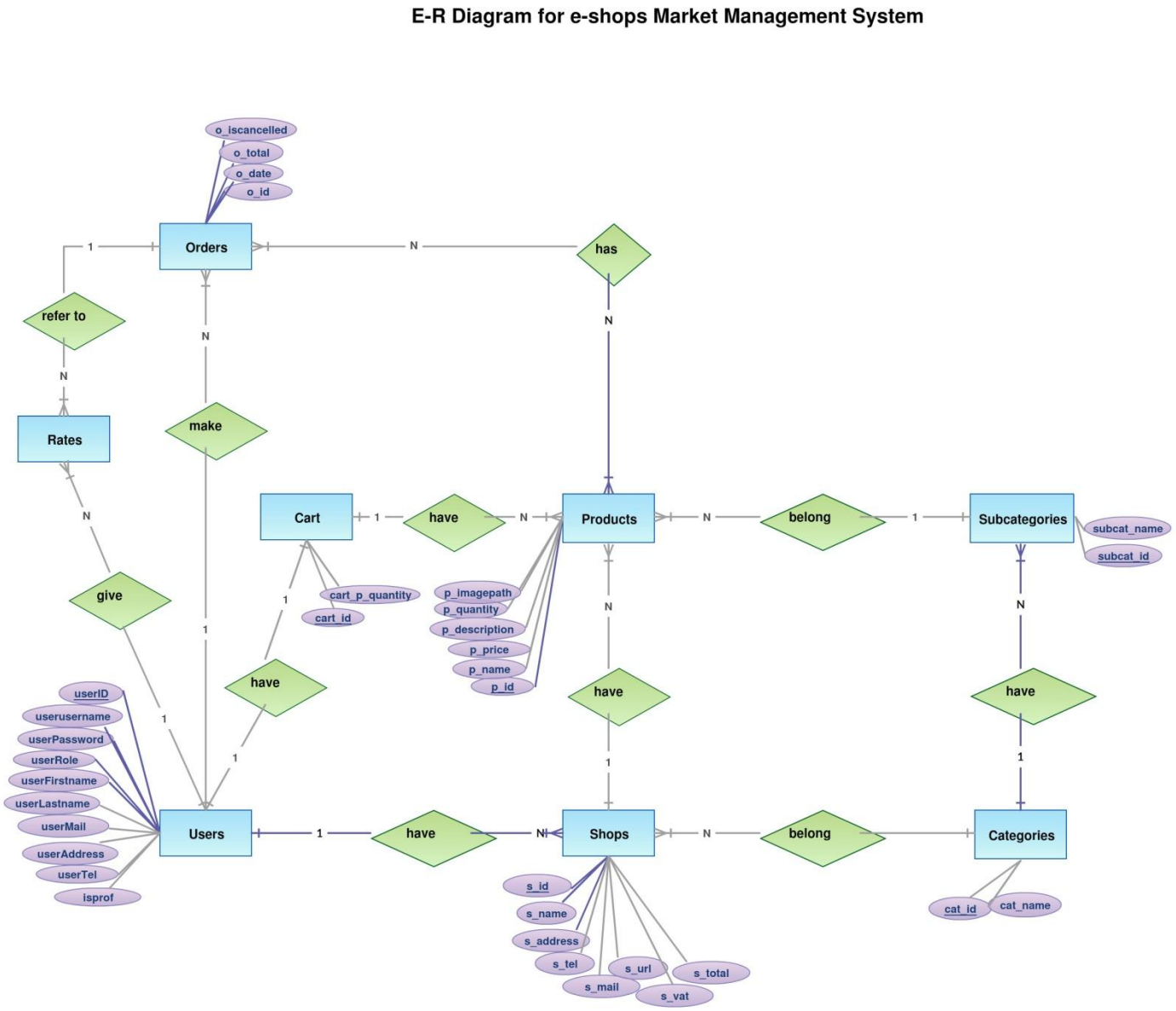
Εικόνα 6:Επιλογή admin λειτουργίας για τον Apache

9. Ακολουθούμε την αντίστοιχη διαδικασία με τον Apache για την MySQL. Αν η εγκατάσταση του XAMPP έχει πραγματοποιηθεί με επιτυχία όταν θα ανοίξουμε ένα πρόγραμμα περιήγησης (web browser) θα εμφανιστεί ένα μήνυμα που θα λέει ότι έχουμε εγκαταστήσει με επιτυχία το XAMPP στο σύστημα μας.



Εικόνα 7: Ολοκλήρωση εγκατάστασης xampp

5 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ ΣΧΕΣΕΩΝ



Εικόνα 8: Διάγραμμα Οντοτήτων Σχέσεων

6 ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Η μετάφραση του διαγράμματος οντοτήτων-σχέσεων στο σχεσιακό μοντέλο σύμφωνα με τις πληθικότητες που προκύπτουν ανάμεσα στις οντότητες. Στους παρακάτω πίνακες του σχεσιακού μοντέλου φαίνονται με υπογράμμιση τα πρωτεύοντα κλειδιά όπως επίσης (italic – όχι bold, στο τέλος κάθε πίνακα) τα κλειδιά τα οποία κληρονομούνται από την μία οντότητα στην άλλη βάσει των πληθικοτήτων.

USERS

userID	**userRole**	**userusername**	**userPassword**	**userFirstname**	**userLastname**	**userAddress**
userTel	**userMail**	**userisprof**				

CATEGORIES

cat_id	**cat_name**

SUBCATEGORIES

subcat_id	**subcat_name**	*cat_id*

SHOPS

s_id	**s_name**	**s_address**	**s_tel**	**s_mail**	**s_url**	**s_vat**	**s_total**	*userID*	*s_category*

PRODUCTS

p_id	**p_name**	**p_price**	**p_description**	**p_quantity**	**p_imagepath**	*subcat_id*	*s_id*

CART

cart_id	*cart_p_id*	*cart_p_quantity*	*cart_p_price*	*cart_userID*

ORDERS

o_id	**o_date**	**o_total**	**o_iscancelled**	*o_userID*

RATES

r_id	**r_deltime**	**r_service**	**r_totalexp**	**r_comments**	*r_userID*	*r_s_id*

ORDERS_HAS_PRODUCTS

ohp_id	*o_id*	*p_id*	*p_quantity*	*p_price*	*r_id*

7 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ

Για να δημιουργήσουμε τη βάση μας θα ακολουθήσουμε τα παρακάτω βήματα.

Ανοίγουμε τη **Γραμμή Εντολών (Command Prompt)** στον υπολογιστή μας.

Start -> πληκτρολόγηση cmd ->Enter

Πηγαίνουμε στο σωστό φάκελο όπου βρίσκεται εγκατεστημένη η mysql, γράφωντας την εντολή:

cd C:\xampp\mysql\bin

Μπαίνουμε μέσα στη mysql, γράφωντας την εντολή:

mysql –u root –p

και όταν μας ζητάει password, πατάμε enter κατευθείαν.

Η πρώτη εντολή

create database bestoffersdb;

δημιουργεί τη βάση για το σύστημά μας ενώ με τη δεύτερη εντολή, ξεκινάμε να τη χρησιμοποιούμε

use bestoffersdb;

Με τις επόμενες εντολές δημιουργούμε τους πίνακες που προέκυψαν από τη μετάφραση του σχεσιακού μοντέλου, δηλώνοντας τα πρωτεύοντα και τα ξένα κλειδιά βάση των κανόνων κληρονόμησης.

```
DROP TABLE IF EXISTS users CASCADE;
create table users
(
        userID              integer not null auto_increment,
        userRole            varchar(10) not null,
        userusername        varchar(20) not null,
        userPassword        varchar(20) not null,

        userFirstname       varchar(32) not null,
         userLastname       varchar(32) not null,
        userAddress         varchar(32) not null,
        userTel             varchar(32) not null,
        userMail            varchar(32) not null,
        userisprof          integer not null,

        primary key (userID)
);
```

```
DROP TABLE IF EXISTS categories CASCADE;
CREATE TABLE categories
(
        cat_id                  integer not null auto_increment,
        cat_name                varchar(32) not null,

        primary key(cat_id)
 );


DROP TABLE IF EXISTS subcategories CASCADE;
CREATE TABLE subcategories
(
        subcat_id               integer not null auto_increment,
         subcat_name            varchar(32) not null,
        cat_id                  integer not null,

        primary key(subcat_id),
        foreign key(cat_id) references categories (cat_id)
 );

DROP TABLE IF EXISTS shops CASCADE;
CREATE TABLE shops
(
        s_id                    integer not null auto_increment,
        s_name                  varchar(32) not null,
        s_address               varchar(32) not null,
        s_category              integer not null,
        s_tel                   varchar(32) not null,
        s_mail                  varchar(32) not null,
        s_url                   varchar(32) not null,
        s_vat                   varchar(32) not null,
        s_total                 float not null,
        userID                  integer not null,
```

```
        primary key(s_id),
        foreign key(userID) references users (userID),
        foreign key(s_category) references categories (cat_id)
 );


DROP TABLE IF EXISTS products CASCADE;
CREATE TABLE products
(
        p_id                integer not null auto_increment,
        p_name              varchar(32) not null,
        p_price             float not null,
        p_description       text not null,
        p_quantity          integer not null,
        p_imagepath         varchar(256) not null,

        subcat_id           integer not null,
        s_id                integer not null,

        primary key(p_id),
        foreign key(subcat_id) references subcategories (subcat_id),
        foreign key(s_id) references shops (s_id)
 );

DROP TABLE IF EXISTS cart CASCADE;
CREATE TABLE cart
(
        cart_id             integer not null auto_increment,

         cart_p_id          integer not null,
        cart_p_quantity      integer not null,
        cart_p_price        float not null,
        cart_userID         integer not null,

        primary key(cart_id),
```

```
        foreign key(cart_p_id) references products (p_id),
        foreign key(cart_userID) references Users (userID)
 );

DROP TABLE IF EXISTS orders CASCADE;
CREATE TABLE orders
(
        o_id            integer not null auto_increment,
        o_date          date not null,
        o_total         float not null,
        o_userID        integer not null,
        o_iscancelled   integer not null,

        primary key(o_id),
        foreign key(o_userID) references Users (userID)
 );


DROP TABLE IF EXISTS rates CASCADE;
CREATE TABLE rates
(
        r_id                    integer not null auto_increment,
        r_deltime               varchar (32) not null,
        r_service               varchar (32) not null,
        r_totalexp              varchar (32) not null,
        r_comments              text not null,
        r_userID                integer not null,
        r_s_id                  integer not null,

        primary key(r_id),
        foreign key(r_userID) references users (userID),
        foreign key(r_s_ID) references shops (s_id)
 );

DROP TABLE IF EXISTS OrdersHasProducts CASCADE;
CREATE TABLE OrdersHasProducts
```

```
(
        ohp_id          integer not null auto_increment,

        o_id            integer not null,
        p_id            integer not null,
        p_quantity      integer not null,
        p_price         float not null,
        r_id            integer not null,

        primary key(ohp_id),
        foreign key(o_id) references orders (o_id), /*on delete cascade*/
        foreign key(p_id) references products (p_id), /*on delete cascade*/
        foreign key(r_id) references rates (r_id)
 );
```

Με την ακριβώς επόμενη εντολή δημουργούμε τον πρώτο χρήστη του συστήματος μας, στον οποίο δίνουμε το ρόλο του administrator ετσι ώστε να έχει τον απόλυτο έλεγχο του όλου συστήματος.

```
insert into users (userRole, userusername, userPassword, userFirstname, userLastname, userAddress, userTel, userMail, userisprof)
values ('admin', 'admin', 'admin', 'adminF', 'adminL', 'adminAddress', '6940000000', 'info@bestoffers.gr', 0);
```

Οι ρόλοι του συστήματος θα είναι 2, ο admin και ο user. Ο user με τη σειρά του διαχωρίζεται έτσι ώστε να μπορεί να είναι απλός ή επαγγελματίας.

Ο διαχωρισμός έγινε για να μπορεί να υπάρχει γενικός έλεγχος στο σύστημα από την πλευρά του admin και γιατί ο απλός χρήστης ή ο επαγγαλματίας χρήστης θα έχει τη δυνατότητα χρήσης μόνο όσων λειτουργιών χρειάζεται.

Οι λειτουργίες που υποστηρίζει το σύστημα είναι οι εξής:

1. Login - Logout
2. Προσθήκη – Διαγραφή Χρήστη
3. Προσθήκη – Διαγραφή Καταστήματος
4. Προσθήκη – Διαγραφή Κατηγορίας
5. Προσθήκη – Διαγραφή Υποκατηγορίας
6. Προσθήκη – Διαγραφή Προϊόντος
7. Προσθήκη – Διαγραφή Παραγγελίας
8. Προβολή – Διαγραφή Αξιολόγησης
9. Συνολικά στατιστικά καταστημάτων
10. Όλα τα καταστήματα
11. Όλα τα προϊόντα
12. Προϊόντα ανα κατηγορία
13. Αναζήτηση
14. Εγγραφή Χρήστη
15. Προβολή Προφίλ

16. Καλάθι
17. Οι παραγγελίες μου – Ακύρωση Παραγγελίας
18. Προσθήκη – Προβολή Αξιολόγησης
19. Τα καταστήματα μου
20. Τα προϊόντα μου
21. Οι παραγγελίες των καταστημάτων μου
22. Τα στατιστικά των καταστημάτων μου

Ο παρακάτω πίνακας δείχνει ποιες λειτουργίες έχει διαθέσιμες ο διαχειριστής (admin)και ποιες ένας απλός χρήστης του συστήματος (user).

ΛΕΙΤΟΥΡΓΙΕΣ	**Admin**	**User**	**Professional User**
Login - Logout	Ναι	Ναι	Ναι
Προσθήκη – Διαγραφή Χρήστη	Ναι		
Προσθήκη – Διαγραφή Καταστήματος	Ναι		Ναι
Προσθήκη – Διαγραφή Κατηγορίας	Ναι		
Προσθήκη – Διαγραφή Υποκατηγορίας	Ναι		
Προσθήκη – Διαγραφή Προϊόντος	Ναι		Ναι
Προσθήκη – Διαγραφή Παραγγελίας	Ναι		
Προβολή – Διαγραφή Αξιολόγησης	Ναι		
Συνολικά στατιστικά καταστημάτων	Ναι		
Όλα τα καταστήματα	Ναι	Ναι	Ναι
Όλα τα προϊόντα	Ναι	Ναι	Ναι
Προϊόντα ανα κατηγορία	Ναι	Ναι	Ναι
Αναζήτηση	Ναι	Ναι	Ναι
Εγγραφή Χρήστη	Ναι	Ναι	Ναι
Προβολή Προφίλ	Ναι	Ναι	Ναι
Καλάθι	Ναι	Ναι	Ναι
Οι παραγγελίες μου – Ακύρωση Παραγγελίας	Ναι	Ναι	Ναι
Προσθήκη – Προβολή Αξιολόγησης	Ναι	Ναι	Ναι
Τα καταστήματα μου	Ναι		Ναι
Τα προϊόντα μου	Ναι		Ναι
Οι παραγγελίες των καταστημάτων μου	Ναι		Ναι
Τα στατιστικά των καταστημάτων μου	Ναι		Ναι

Πίνακας 1: Διαθέσιμες λειτουργίες συστήματος ανά ρόλο

8 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΥΣΤΗΜΑΤΟΣ

8.1 Βασικά Χαρακτηριστικά Σελίδων Συστήματος

Ο παρακάτω πίνακας συγκεντρώνει όλα τα αρχεία του συστήματος μας κατηγοριοποιημένα στους αντίστοιχους φακέλους, περιγράφοντας τη λειτουργία του καθενός.

Φάκελοι αρχείων	**Αρχεία**	**Περιγραφή**
bestoffers (κεντρικός φάκελος)	Index.php	Εδώ καθορίζεται ανάλογα α) με το ρόλο β)με το αν εισαι logged in ή όχι και γ) με την παράμετρο page στο url ποια σελίδα θα «φορτώσει».
admin	ShopsAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Καταστημάτων.
	CategoriesAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Κατηγοριών.
	SubcategoriesAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Υποκατηγοριών.
	ProductsAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Προϊόντων.
	OrdersAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Παραγγελιών.
	ShopsTotalsAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Στατιστικών όλων των καταστημάτων.
	RatesAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Αξιολογήσεων.
	UsersAdmin.inc	Αντιστοιχεί στη διαχειριστική σελίδα των Χρηστών.
whole	Cart.inc	Αντιστοιχεί στη σελίδα του Καλαθιού
	MyShops.inc	Αντιστοιχεί στη σελίδα των Καταστημάτων ενός επαγγελματία.
	MyShopsOrders.inc	Αντιστοιχεί στη σελίδα των Παραγγελιών στα καταστήματα ενός επαγγελματία.
	Index.inc	Αντιστοιχεί στο intro page του συστήματος.
	MyShopsTotals.inc	Αντιστοιχεί στη σελίδα των Στατιστικών των Καταστημάτων ενός επαγγελματία.
	Orders.inc	Αντιστοιχεί στη σελίδα των Παραγγελιών.
	Products.inc	Αντιστοιχεί στη σελίδα των διαθέσιμων Προϊόντων.
	ProductsCats.inc	Αντιστοιχεί στη σελίδα των διαθέσιμων Κατηγοριών των Προϊόντων.
	ProductsSubcats.inc	Αντιστοιχεί στη σελίδα των διαθέσιμων Υποκατηγοριών των Προϊόντων
	ProductsSubcatsDisplay.inc	Αντιστοιχεί στη σελίδα των Προϊόντων βάσει της επιλογής Υποκατηγορίας
	Profile.inc	Αντιστοιχεί στη σελίδα των πληροφοριών προφίλ ενός χρήστη
	Rates.inc	Αντιστοιχεί στη σελίδα των Αξιολογήσεων .
	Search.inc	Αντιστοιχεί στη σελίδα Αναζήτησης του οδηγού
	Register.inc	Αντιστοιχεί στη σελίδα της εγγραφής χρήστη.
	ShopRates.inc	Αντιστοιχεί στη σελίδα των Αξιολογήσεων για ένα

		μαγαζί.
	Shops.inc	Αντιστοιχεί στη σελίδα των διαθέσιμων Καταστημάτων.
Template	menu_admin.inc	Αντιστοιχεί στο τμήμα του Template που περιέχει τις διαχειριστικές επιλογές που έχει διαθέσιμες ο admin
	login.inc	Αντιστοιχεί στο τμήμα του Template που περιέχει τη φόρμα του Login-Logout.
	menu_user.inc	Αντιστοιχεί στο τμήμα του Template που περιέχει τις διαχειριστικές επιλογές που έχει διαθέσιμες ο κάθε εγγεγραμμένος χρήστης (απλός ή επαγγελματίας).
	menu_all.inc	Αντιστοιχεί στο τμήμα του Template που περιέχει τις επιλογές που έχει διαθέσιμες ο κάθε χρήστης, εγγεγραμμένος ή μη.
	meta.inc	Αντιστοιχεί στο τμήμα του Template που περιέχει τις metadata πληροφορίες.
	Template.inc	Αντιστοιχεί στο σύνθεση-τρόπο εμφάνισης των επιμέρους τμημάτων του Template.
	Validate.inc	Αντιστοιχεί στο τμήμα του Template που κάνει την πιστοποίηση εισαγωγής σωστών username-password στη φόρμα του Login.

Πίνακας 2: Περιεχόμενα φακέλων-αρχείων

Χρησιμοποιήσαμε .inc αρχεία ώστε να είναι πιο εύκολη

Α) η σύνθεση-εμφάνιση των επιμέρους μενού ανάλογα με το ρόλο (admin ή απλός χρήστης ή επαγγελματίας χρήστης)

Β) η διαχείριση των κάθε λειτουργικών σελίδων ανάλογα με το ρόλο.

Η βασική κατηγοριοποίηση κάθε αρχείου που αναφέρεται σε διαχειριστική λειτουργία είναι οι παρακάτω:

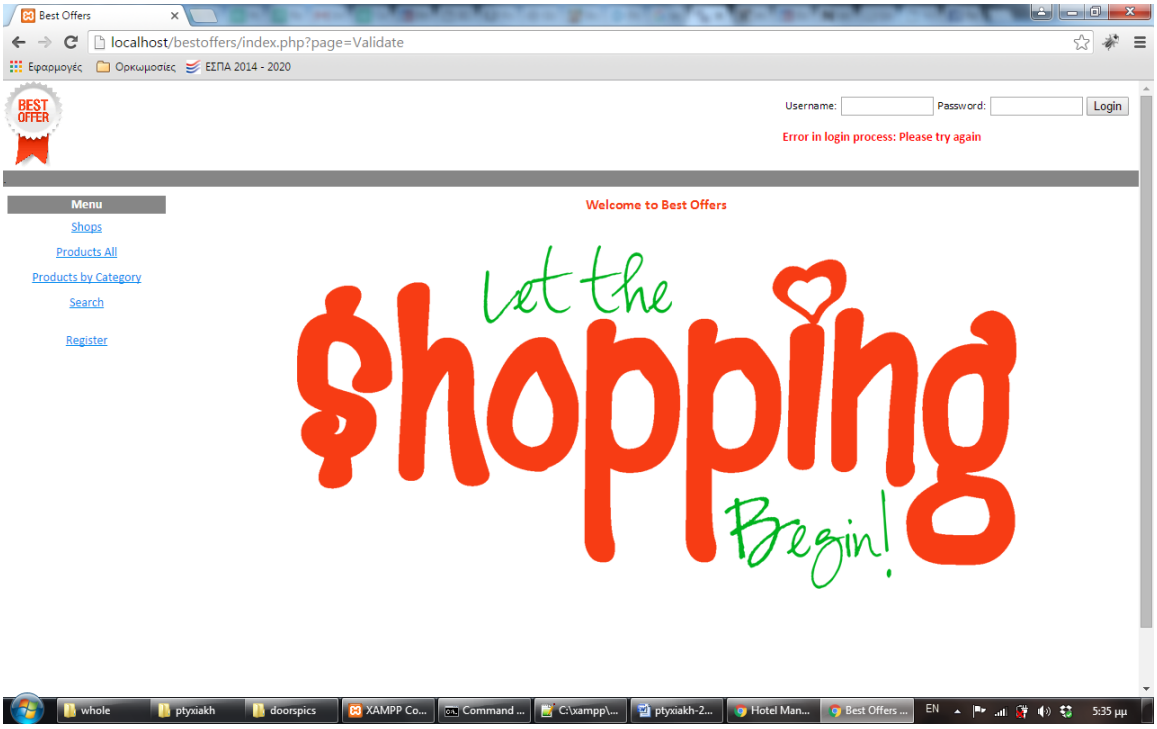
Actions	**Flag**	**Fields check**	**Id**	**Result**
add	true	All with data	-	Ολοκλήρωση Καταχώρησης
		Missing data	-	Εμφάνιση φόρμας με μήνυμα λάθους ότι δεν έχουν συμπληρωθεί όλα τα πεδία¨
	(false)	-	-	Εμφάνιση φόρμας καταχώρησης
delete	true	-	yes	Διαγραφή καταχώρησης με το συγκεκριμένο id
No action (else)	**-**	**-**	**-**	**Α) Εμφάνιση αρχικής οθόνης με σύνδεσμο για προσθήκη και Β) λίστα όλων των ήδη καταχωρημένων εγγραφών με σύνδεσμο επεξεργασίας/διαγραφής για καθεμία**

Πίνακας 3: Ενέργειες κώδικα κάθε αρχείου στον φάκελο admin

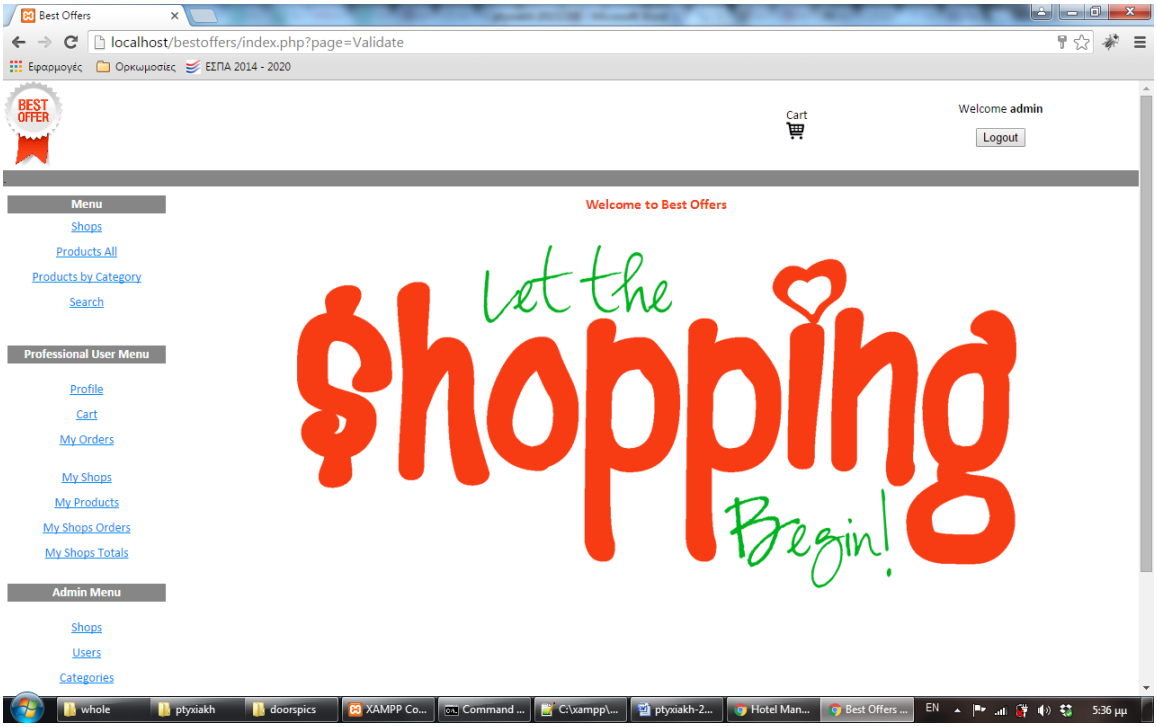
8.2 Login - Logout

Πραγματοποιείται έλεγχος σωστής εισαγωγής Ονόματος χρήστη – Κωδικού

Screenshots:



Εικόνα 9:Έλεγχος κωδικού-ονόματος χρήστη



Εικόνα 10: Εισαγωγή στο σύστημα - Μήνυμα καλωσορίσματος

SQL Query:

select * from users where UserUsername='$user' and UserPassword='$pass'

Εάν υπάρχει τέτοιος χρήστης τότε ενεργοποιείται η είσοδος στο σύστημα με καλωσόρισμα του χρήστη.

Επίσης ενεργοποιούνται οι php SESSION μεταβλητές τις οποίες θα χρειαστούμε κατά κύριο λόγο στο διαχωρισμό των λειτουργιών που πρέπει να έχει στη διάθεση του ο admin και ένας απλός user

```
$_SESSION['Logged']=1;

$_SESSION['Name']=$temp[2];

$_SESSION['Role']=$temp[1];

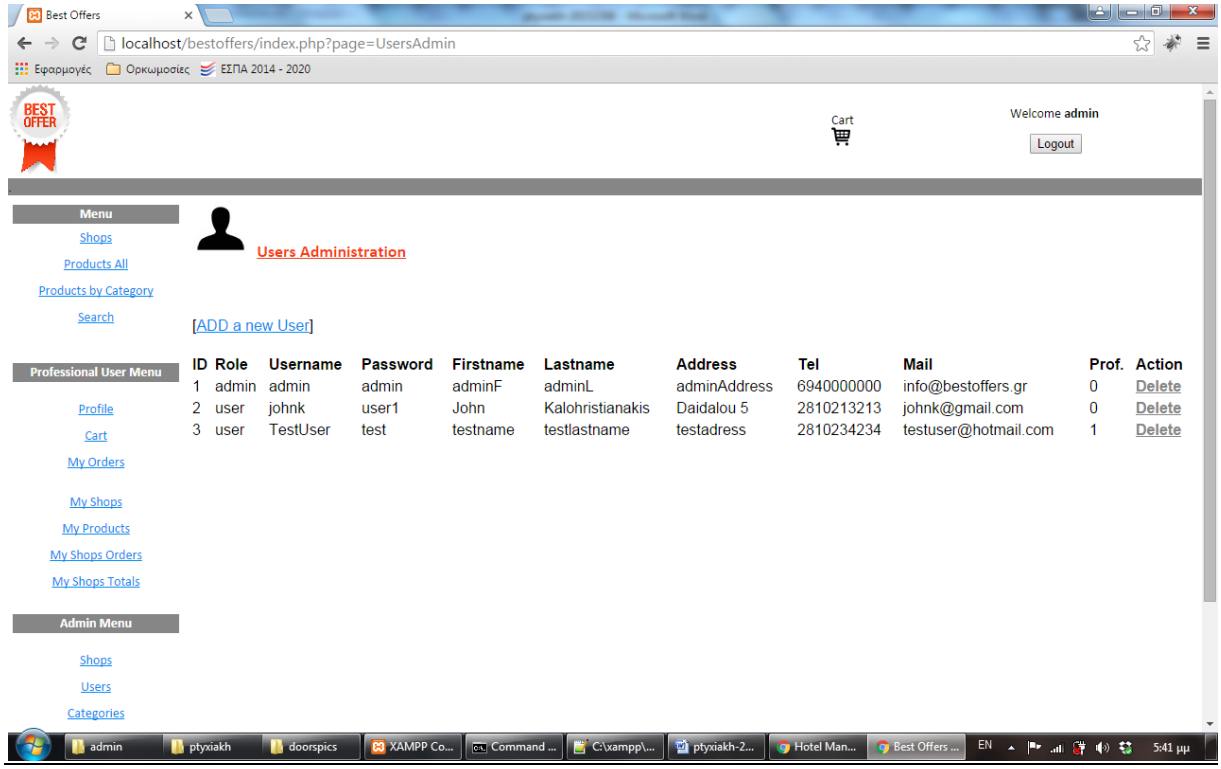
$_SESSION['userID']=$temp[0];

$_SESSION['isprof']=$temp[9];
```

8.3 Χρήστες Διαχειριστή

8.3.1 Αρχική Οθόνη Χρηστών

Screenshots:



Εικόνα 11: Αρχική Οθόνη Χρηστών

Στην αρχική οθόνη των χρηστών εμφανίζεται η λίστα με τους υπάρχοντες χρήστες και δίπλα σε κάθε έναν, οι σύνδεσμοι Διαγραφής (Delete) για κάθε έναν από αυτούς.

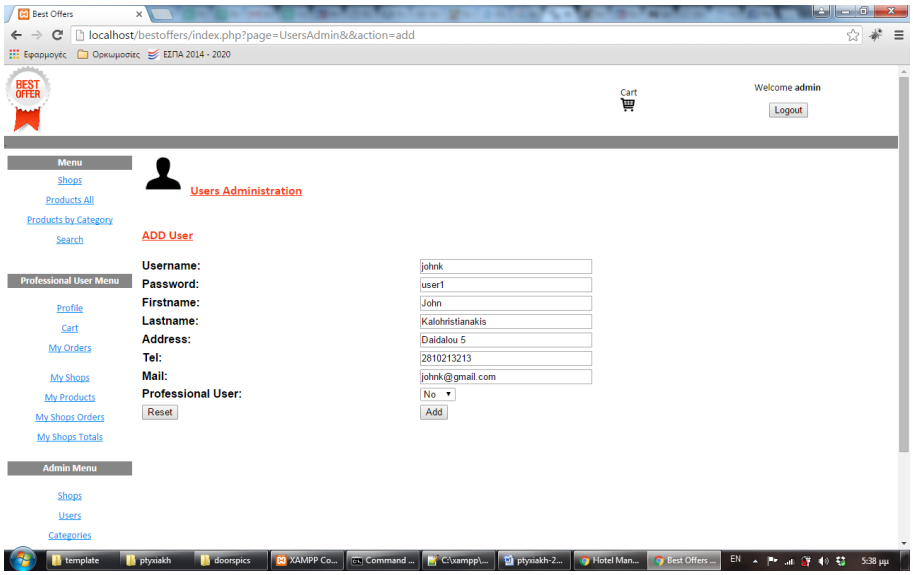
Εάν δεν έχει δημιουργηθεί κάποιος χρήστης, τότε εμφανίζεται μήνυμα μη ύπαρξης χρηστών.

Επίσης, στο πάνω μέρος της αρχικής οθόνης χρηστών, υπάρχει σύνδεσμος που μας οδηγεί στη φόρμα δημιουργίας χρήστη.

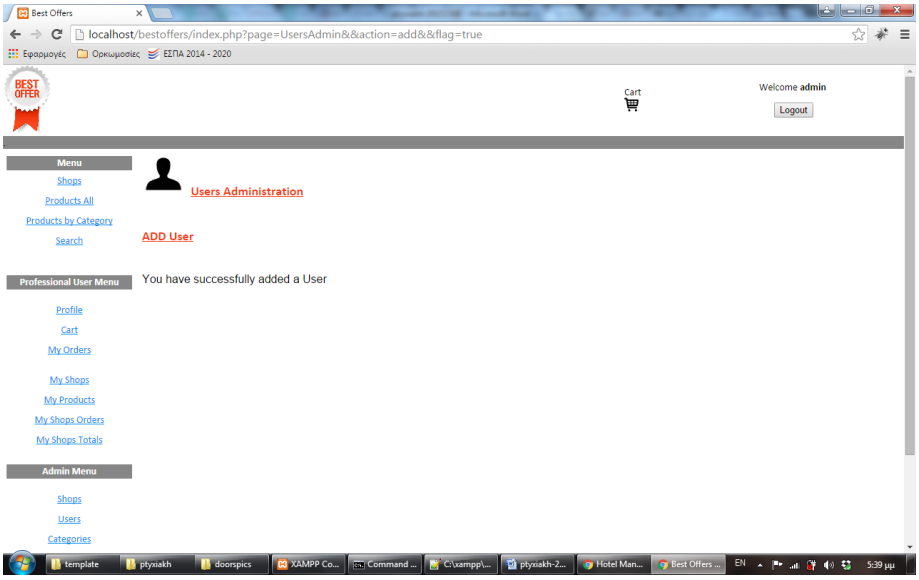
8.3.2 Δημιουργία Χρήστη

Screenshots:

Στην οθόνη Δημιουργίας Χρήστη, εμφανίζεται η φόρμα με τα απαραίτητα πεδία που πρέπει να συμπληρωθούν ώστε να δημιουργηθεί ένας χρήστης.



Εικόνα 12: Φόρμα Προσθήκης Νέου Χρήστη



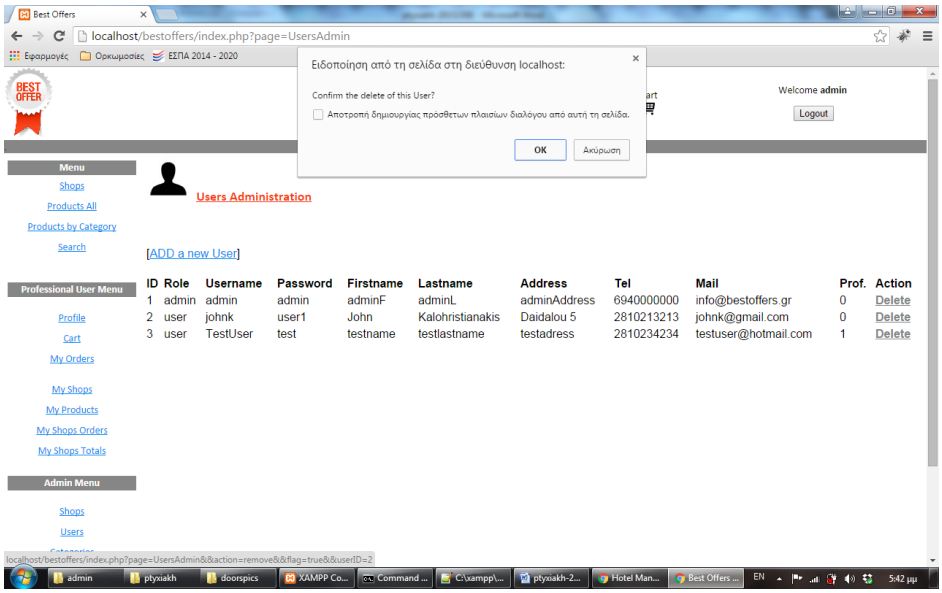
Εικόνα 13: Ολοκλήρωση προσθήκης Νέου Χρήστη

SQL query:

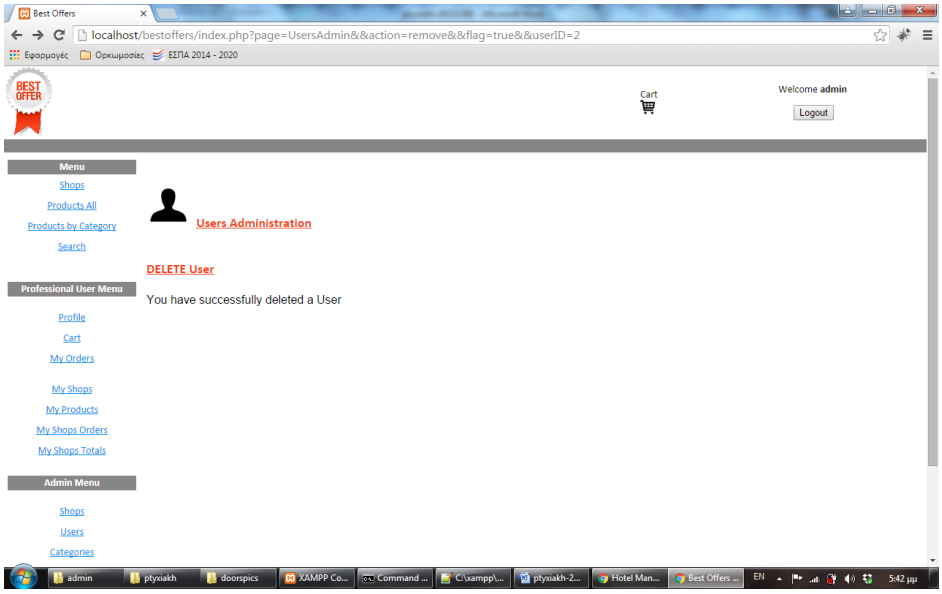
insert into users (userRole, userusername, userPassword, userFirstname, userLastname, userAddress, userTel, userMail, userisprof) values ('user', '$userusername', '$userPassword', '$userFirstname', '$userLastname', '$userAddress', '$userTel', '$userMail', $userisprof)

8.3.3 Διαγραφή Χρήστη

Screenshots:



Εικόνα 14: Διαγραφή Χρήστη 1



Εικόνα 15: Διαγραφή Χρήστη 2

SQL query:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής του συγκεκριμένου χρήστη, περνώντας το id του ως παράμετρο.

delete from users where userID=$StrtsID

8.4 Κατηγορίες Διαχειριστή

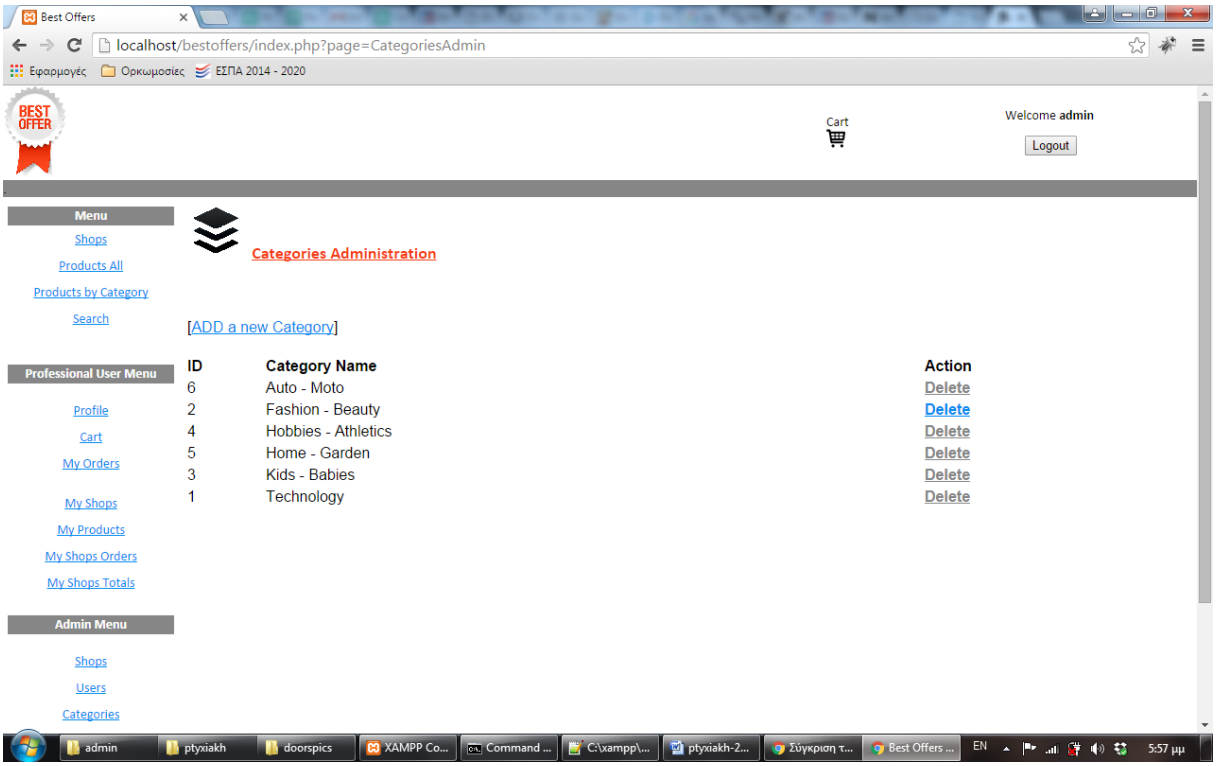
8.4.1 Αρχική Οθόνη Κατηγοριών

Screenshots:

Στην αρχική οθόνη των Κατηγορίων εμφανίζεται η λίστα με τις υπάρχουσες κατηγορίες και δίπλα σε κάθε μια, ο σύνδεσμοι Διαγραφής (Delete) για κάθε μία από αυτές.

Εάν δεν έχει δημιουργηθεί κάποιο κατηγορία, τότε εμφανίζεται μήνυμα μη ύπαρξης Κατηγορίων.

Επίσης, στο πάνω μέρος της αρχικής οθόνης Κατηγοριών, υπάρχει σύνδεσμος που μας οδηγεί στη φόρμα δημιουργίας κατηγορίων.

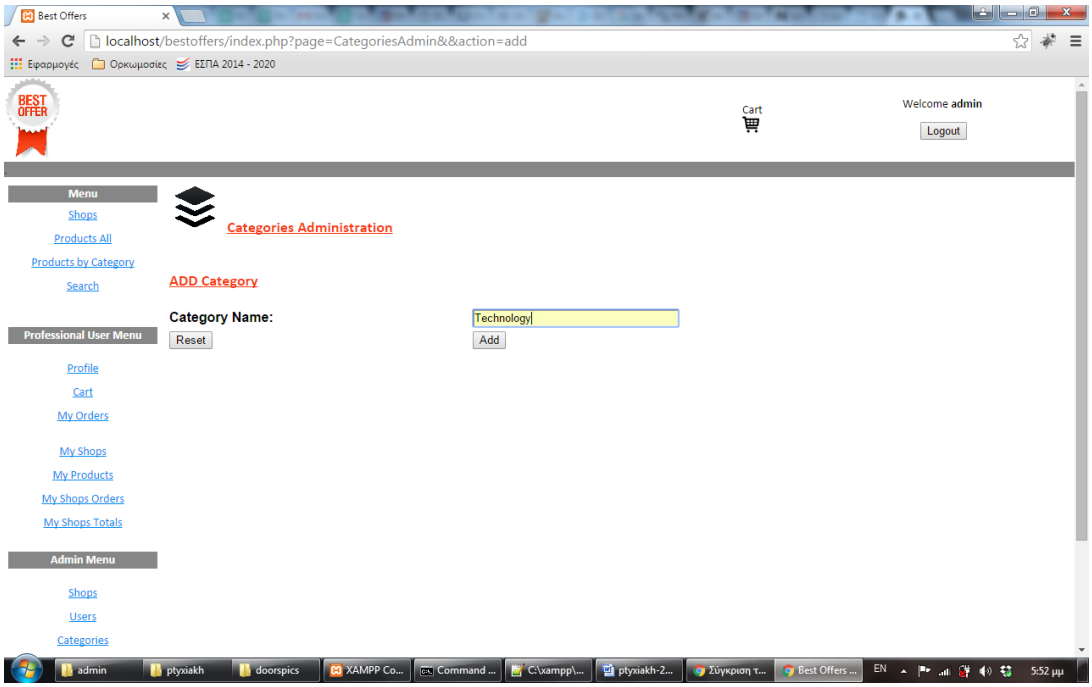


Εικόνα 16: Αρχική Οθόνη Κατηγοριών

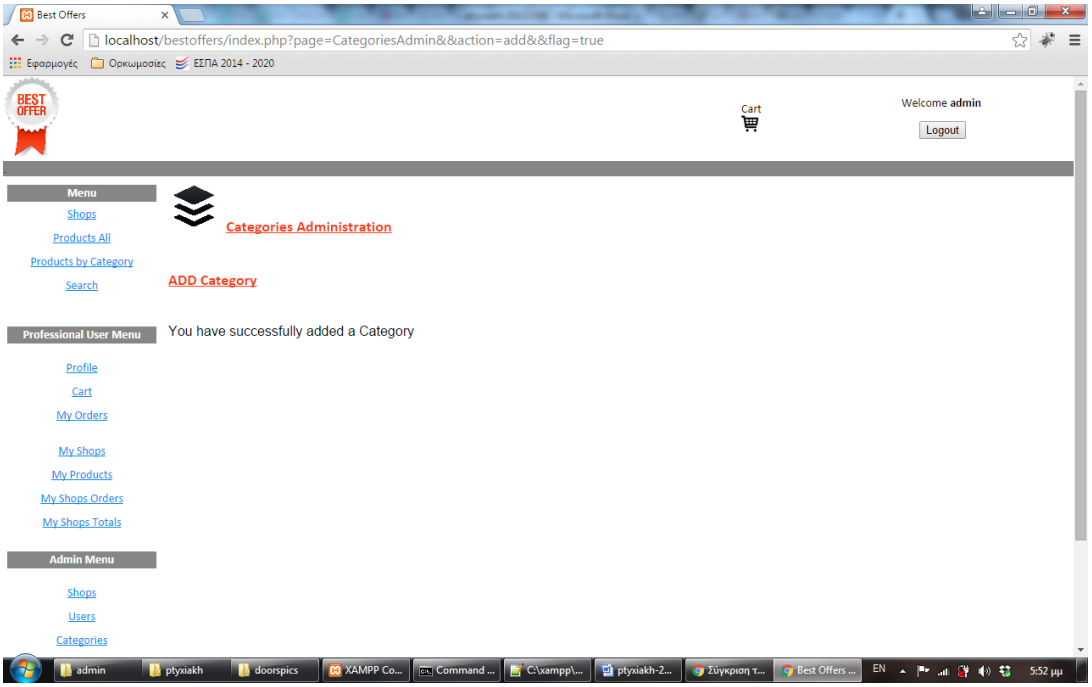
8.4.2 Δημιουργία Κατηγορίας

Screenshots:

Στην οθόνη Δημιουργίας Κατηγορίας, εμφανίζεται η φόρμα με τα απαραίτητα πεδία που πρέπει να συμπληρωθούν ώστε να δημιουργηθεί μία κατηγορία.



Εικόνα 17: Φόρμα Προσθήκης Νέας Κατηγορίας



Εικόνα 18: Ολοκλήρωση προσθήκης Νέας Κατηγορίας

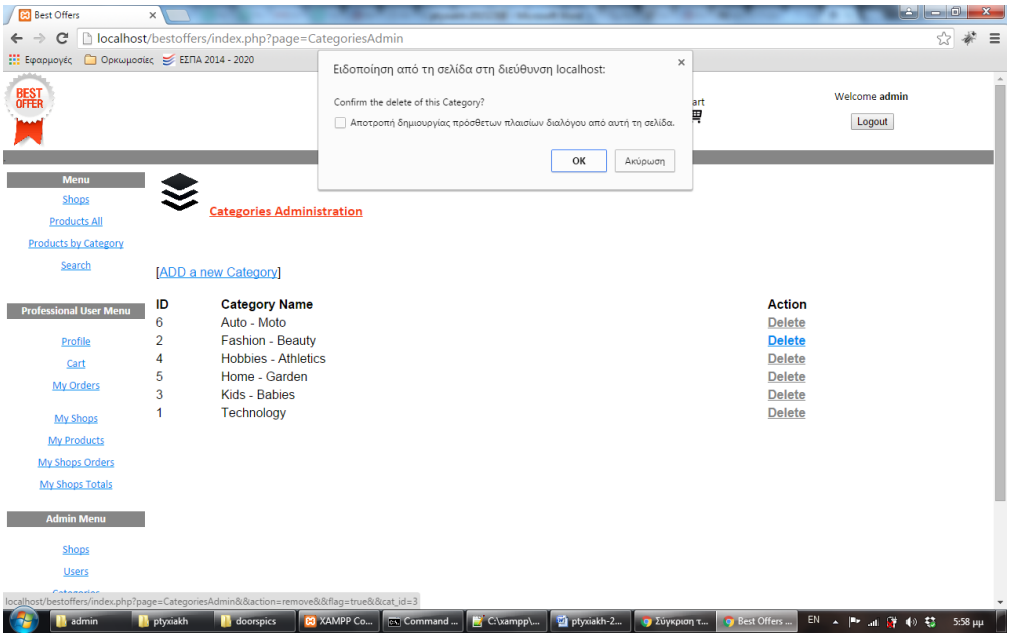
SQL query:

insert into categories(cat_name) values ('$cat_name')

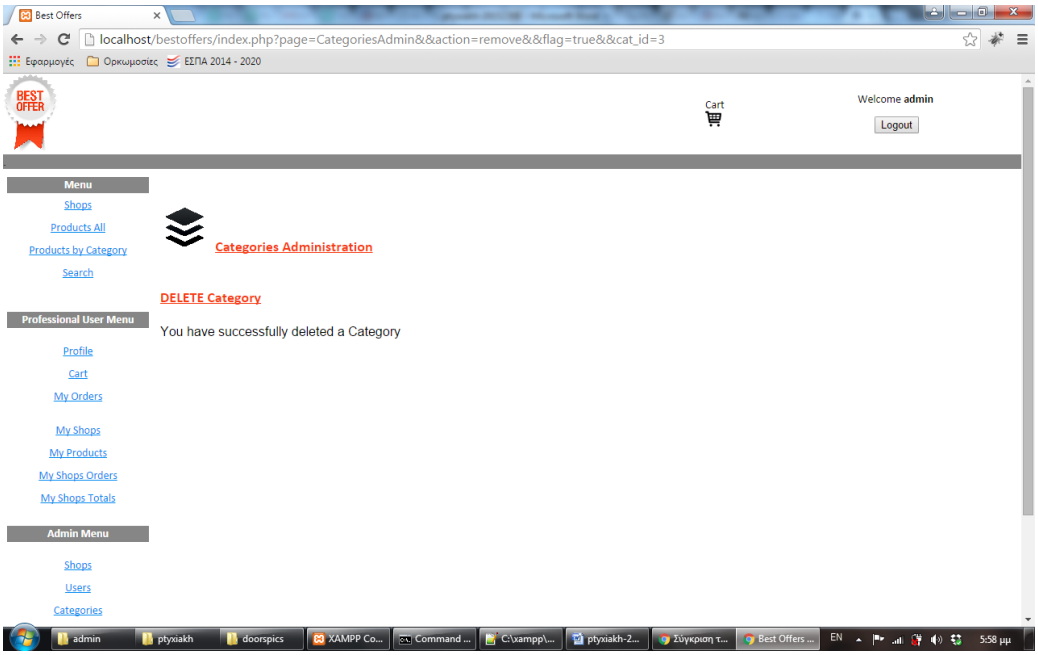
8.4.3 Διαγραφή Κατηγορίας

Screenshots:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής, περνώντας το id του ως παράμετρο.



Εικόνα 19: Διαγραφή Κατηγορίας 1



Εικόνα 20: Διαγραφή Κατηγορίας 2

SQL query:

delete from categories where cat_id=$StrtsID

8.5 Υποκατηγορίες Διαχειριστή

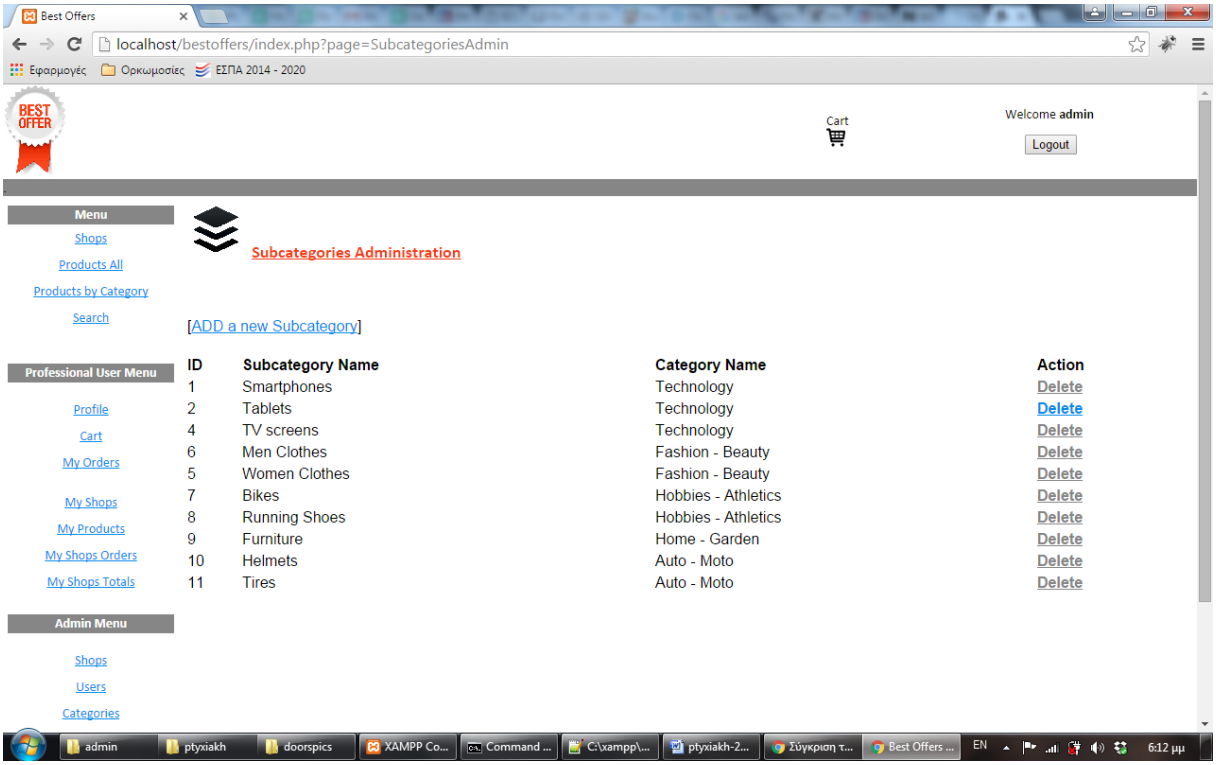
8.5.1 Αρχική Οθόνη Υποκατηγοριών

Screenshots:

Στην αρχική οθόνη των Υποκατηγορίων εμφανίζεται η λίστα με τις υπάρχουσες υποκατηγορίες και δίπλα σε κάθε μια, ο σύνδεσμοι Διαγραφής (Delete) για κάθε μία από αυτές.

Εάν δεν έχει δημιουργηθεί κάποιο υποκατηγορία, τότε εμφανίζεται μήνυμα μη ύπαρξης Υποκατηγορίων.

Επίσης, στο πάνω μέρος της αρχικής οθόνης Υποκατηγοριών, υπάρχει σύνδεσμος που μας οδηγεί στη φόρμα δημιουργίας υποκατηγορίων.

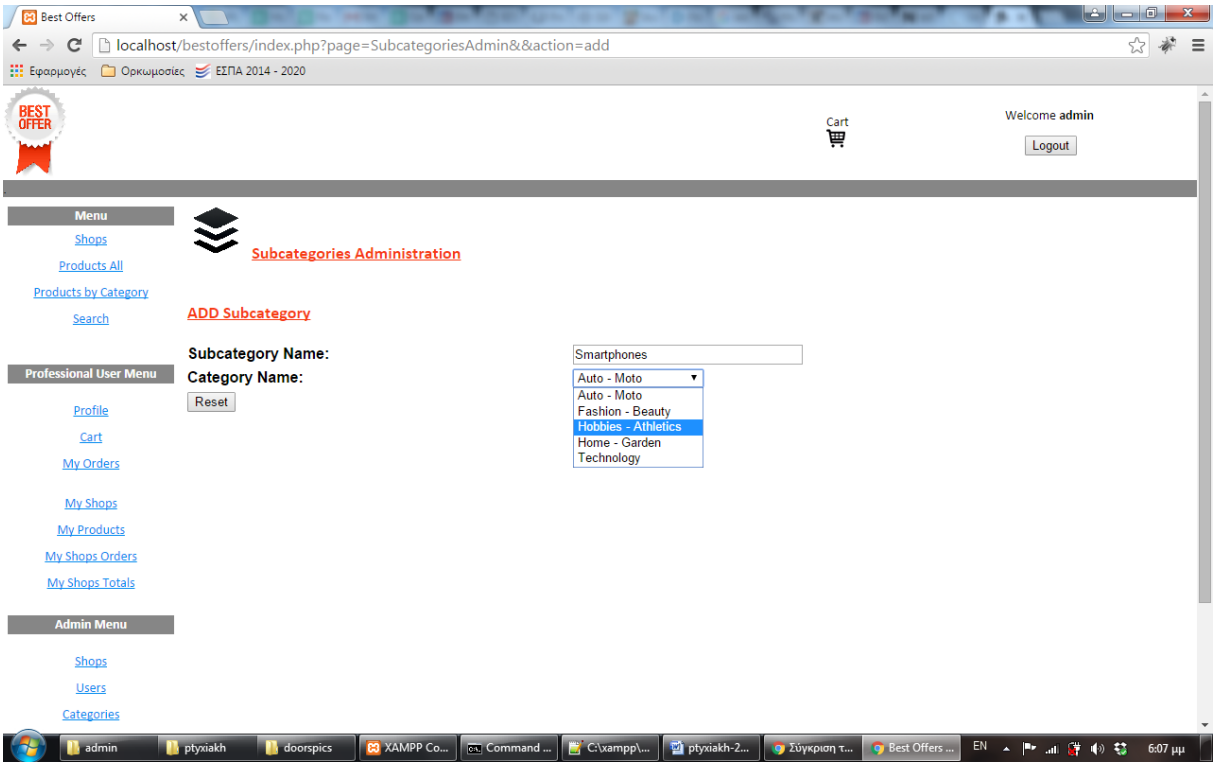


Εικόνα 21: Αρχική Οθόνη Υποκατηγοριών

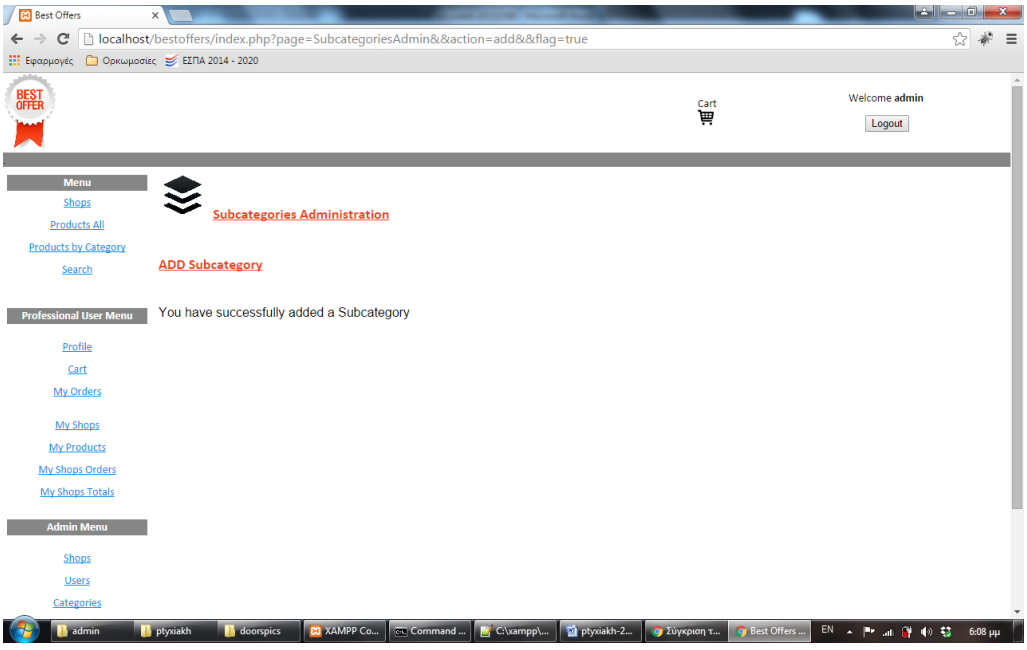
8.5.2 Δημιουργία Υποκατηγορίας

Screenshots:

Στην οθόνη Δημιουργίας Κατηγορίας, εμφανίζεται η φόρμα με τα απαραίτητα πεδία που πρέπει να συμπληρωθούν ώστε να δημιουργηθεί μία κατηγορία.



Εικόνα 22: Φόρμα Προσθήκης Νέας Υποκατηγορίας



Εικόνα 23: Ολοκλήρωση προσθήκης Νέας Υποκατηγορίας

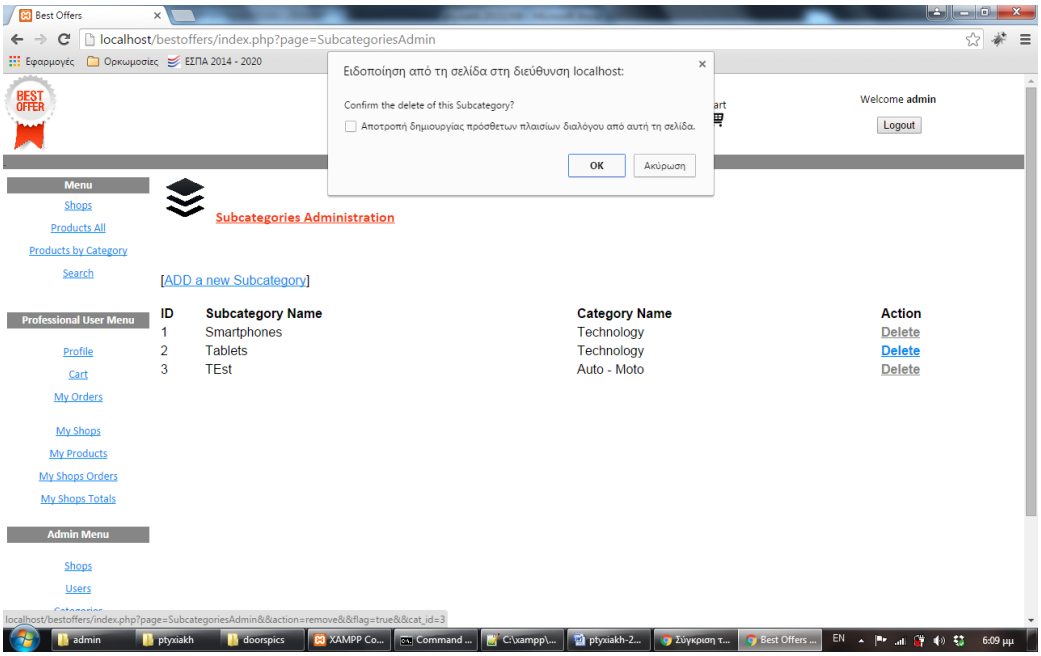
SQL query:

insert into subcategories(subcat_name, cat_id) values ('$subcat_name', $cat_id)

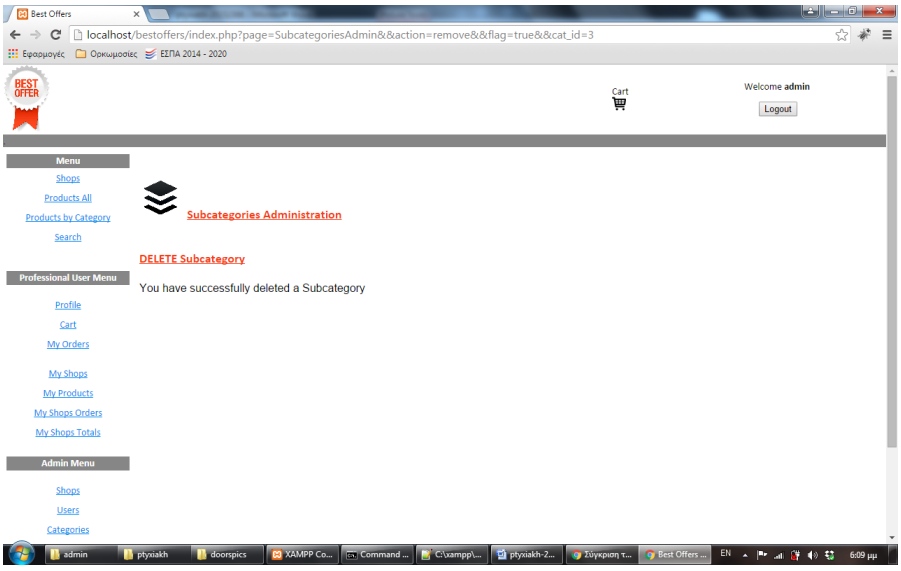
8.5.3 Διαγραφή Υποκατηγορίας

Screenshots:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής, περνώντας το id του ως παράμετρο.



Εικόνα 24: Διαγραφή ΥποΚατηγορίας 1



Εικόνα 25: Διαγραφή Υποκατηγορίας 2

SQL query:

delete from subcategories where subcat_id=$StrtsID

8.6 Καταστήματα Διαχειριστή

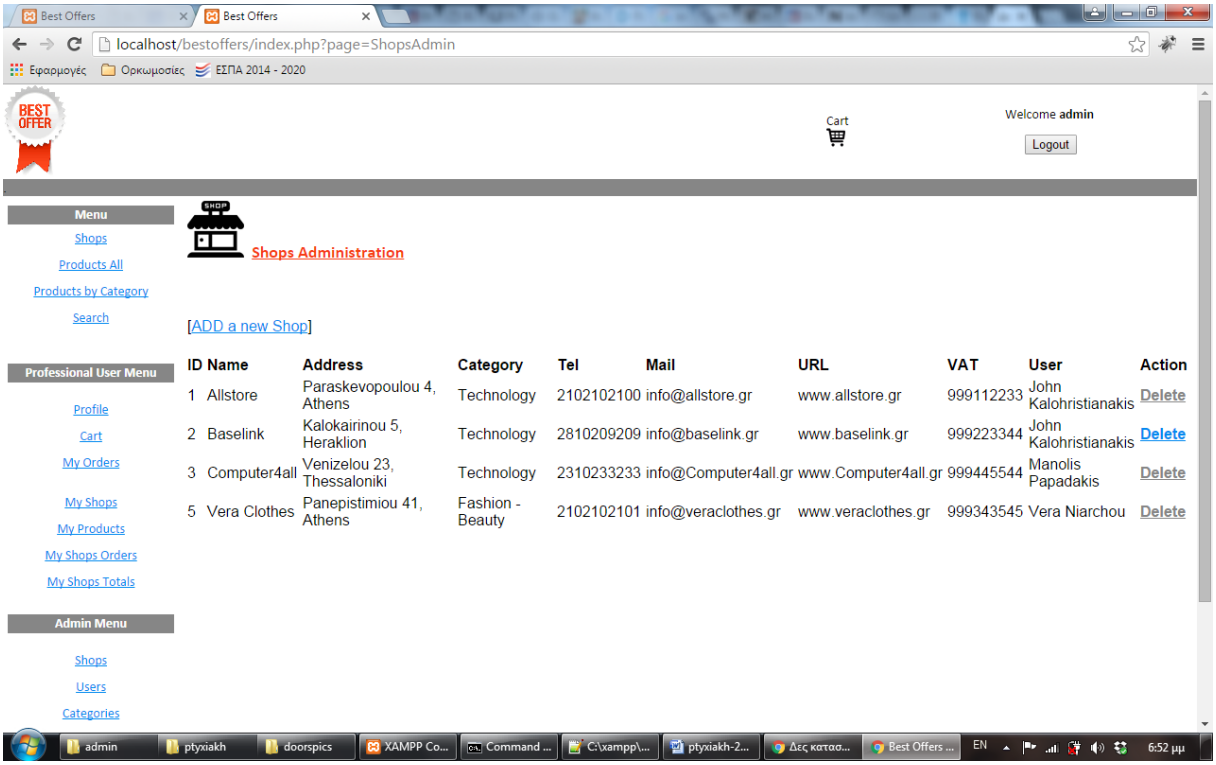
8.6.1 Αρχική Οθόνη Καταστημάτων

Screenshots:

Στην αρχική οθόνη των καταστημάτων εμφανίζεται η λίστα με τα υπάρχοντα καταστήματα και δίπλα σε κάθε ένα, οι σύνδεσμοι Διαγραφής (Delete) για κάθε ένα από αυτά.

Εάν δεν έχει δημιουργηθεί κάποιο κατάστημα, τότε εμφανίζεται μήνυμα μη ύπαρξης καταστημάτων.

Επίσης, στο πάνω μέρος της αρχικής οθόνης καταστημάτων, υπάρχει σύνδεσμος που μας οδηγεί στη φόρμα δημιουργίας καταστήματος.

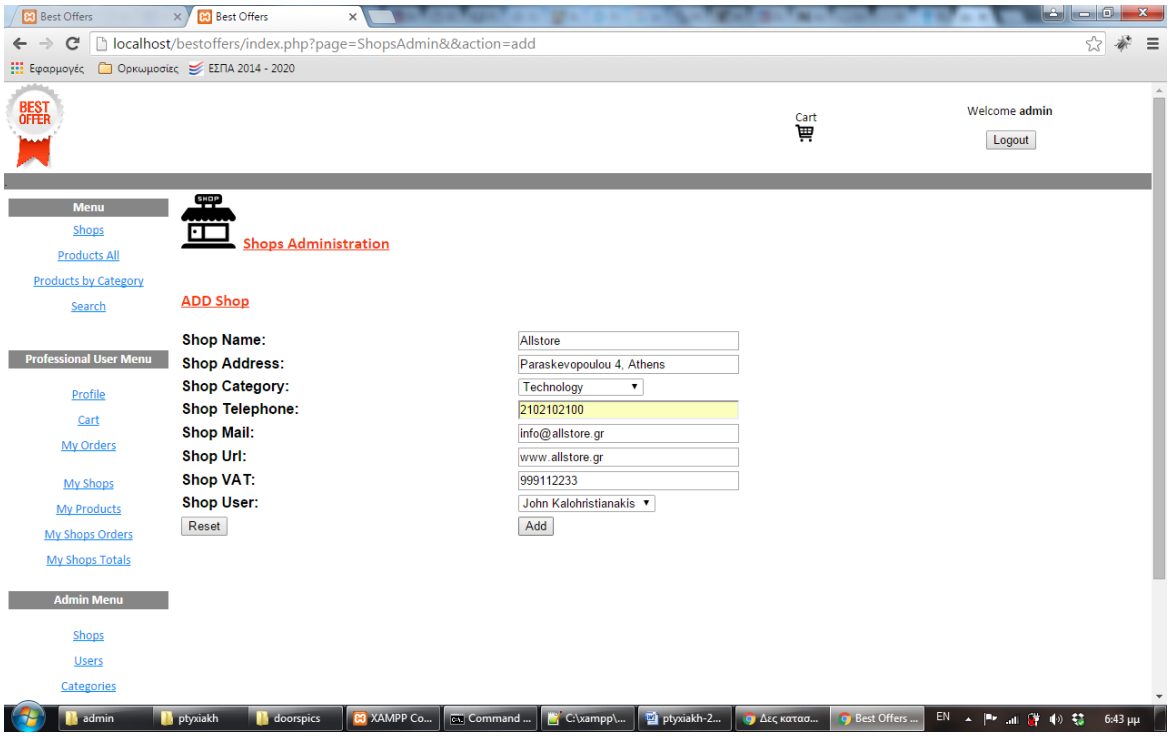


Εικόνα 26: Αρχική Οθόνη Καταστημάτων

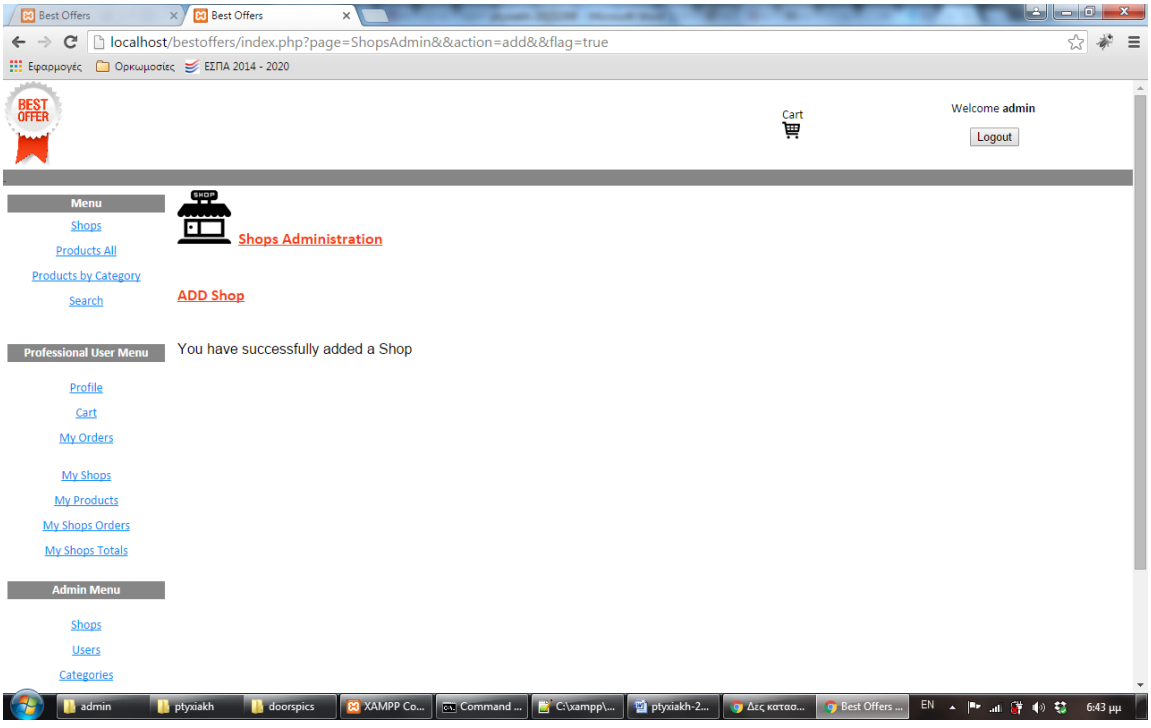
8.6.2 Δημιουργία Καταστήματος

Screenshots:

Στην οθόνη Δημιουργίας Καταστήματος, εμφανίζεται η φόρμα με τα απαραίτητα πεδία που πρέπει να συμπληρωθούν ώστε να δημιουργηθεί ένα κατάστημα.



Εικόνα 27: Φόρμα Προσθήκης Καταστήματος



Εικόνα 28:Ολοκλήρωση προσθήκης Νέου Καταστήματος

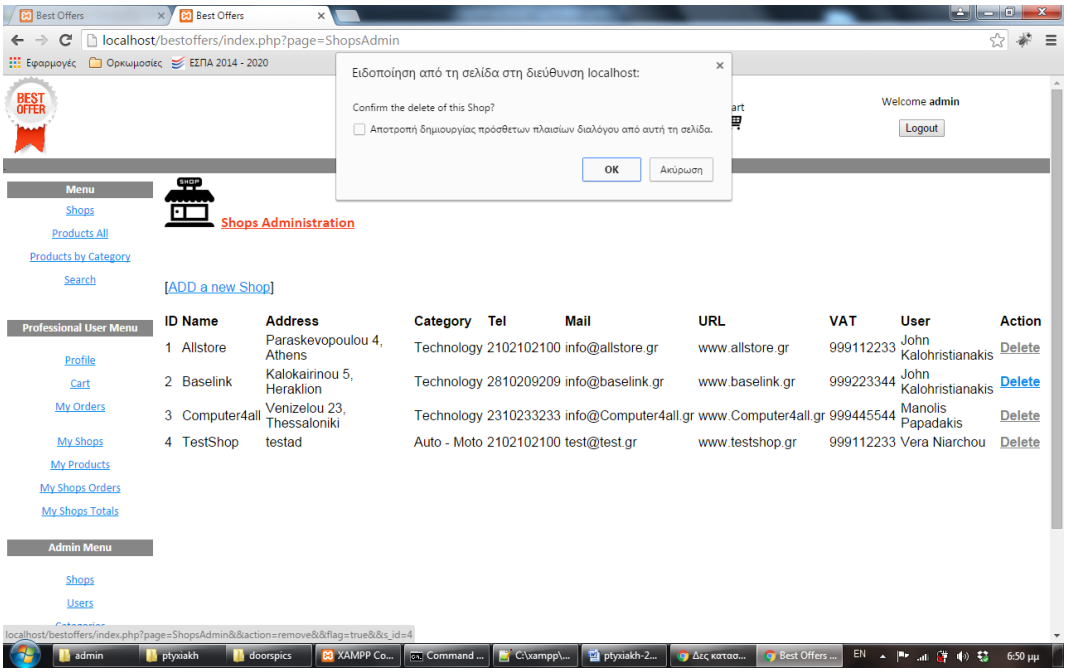
SQL query:

```
insert into shops (s_name, s_address, s_category, s_tel, s_mail, s_url, s_vat, userID, s_total)
values ('$s_name', '$s_address', '$s_category', '$s_tel', '$s_mail', '$s_url', '$s_vat', $userID, 0)
```

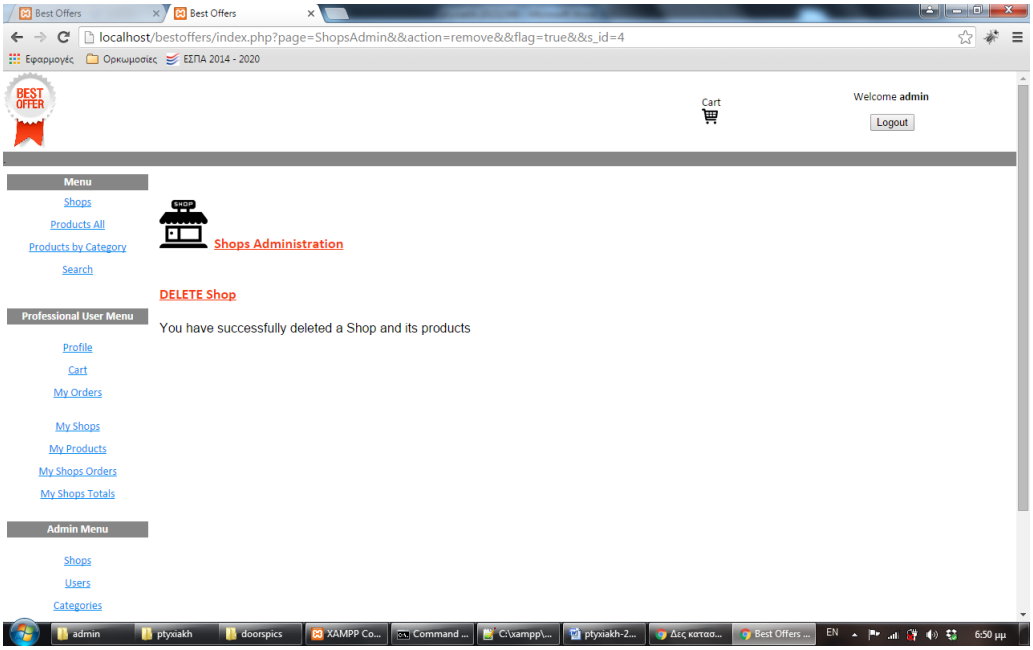
8.6.3 Διαγραφή Καταστήματος

Screenshots:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής, περνώντας το id του ως παράμετρο.



Εικόνα 29: Διαγραφή Καταστήματος1



Εικόνα 30:Διαγραφή Καταστήματος 2

SQL query:

delete from shops where s_id=$StrtsID

8.7 Προϊόντα Διαχειριστή

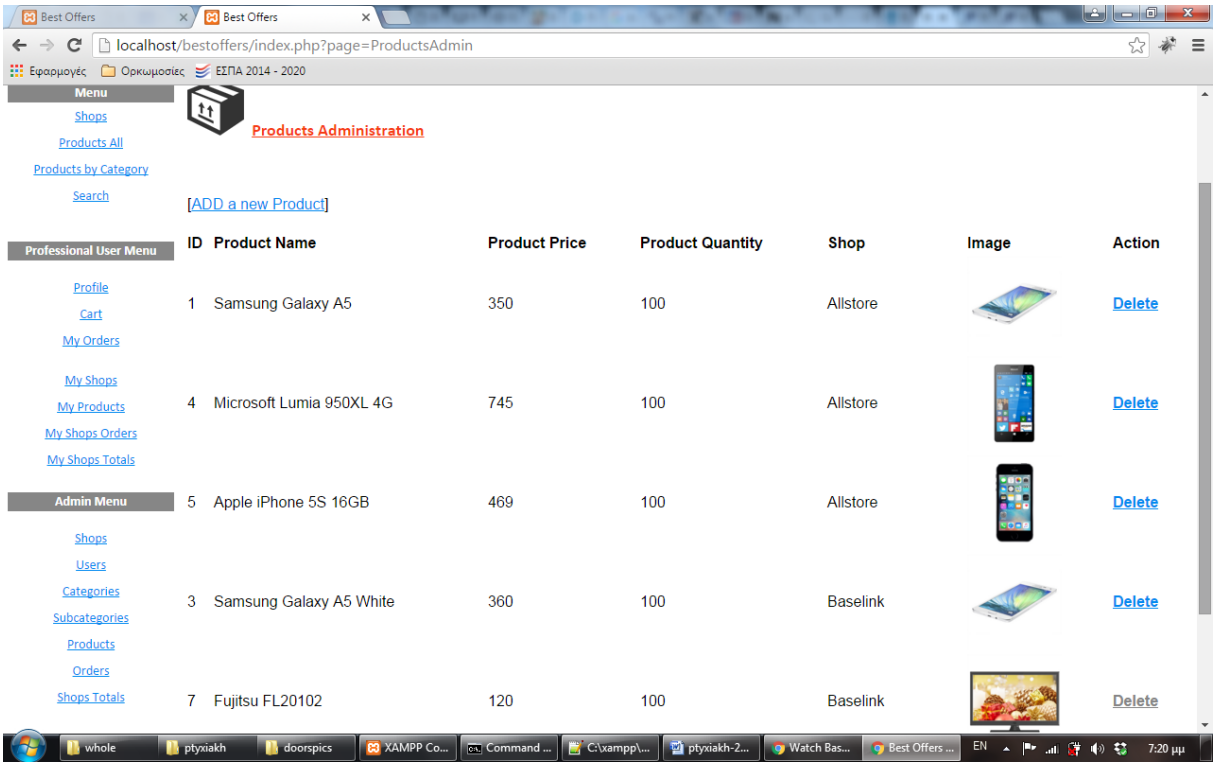
8.7.1 Αρχική Οθόνη Προϊόντων

Screenshots:

Στην αρχική οθόνη των Προϊόντων εμφανίζεται η λίστα με τα υπάρχοντα προϊοντα και δίπλα σε κάθε ένα, ο σύνδεσμος Διαγραφής (Delete) για κάθε ένα από αυτά.

Εάν δεν έχει δημιουργηθεί κάποιο προιον, τότε εμφανίζεται μήνυμα μη ύπαρξης Προϊόντων.

Επίσης, στο πάνω μέρος της αρχικής οθόνης Προϊόντων, υπάρχει σύνδεσμος που μας οδηγεί στη φόρμα δημιουργίας Προϊόντων.

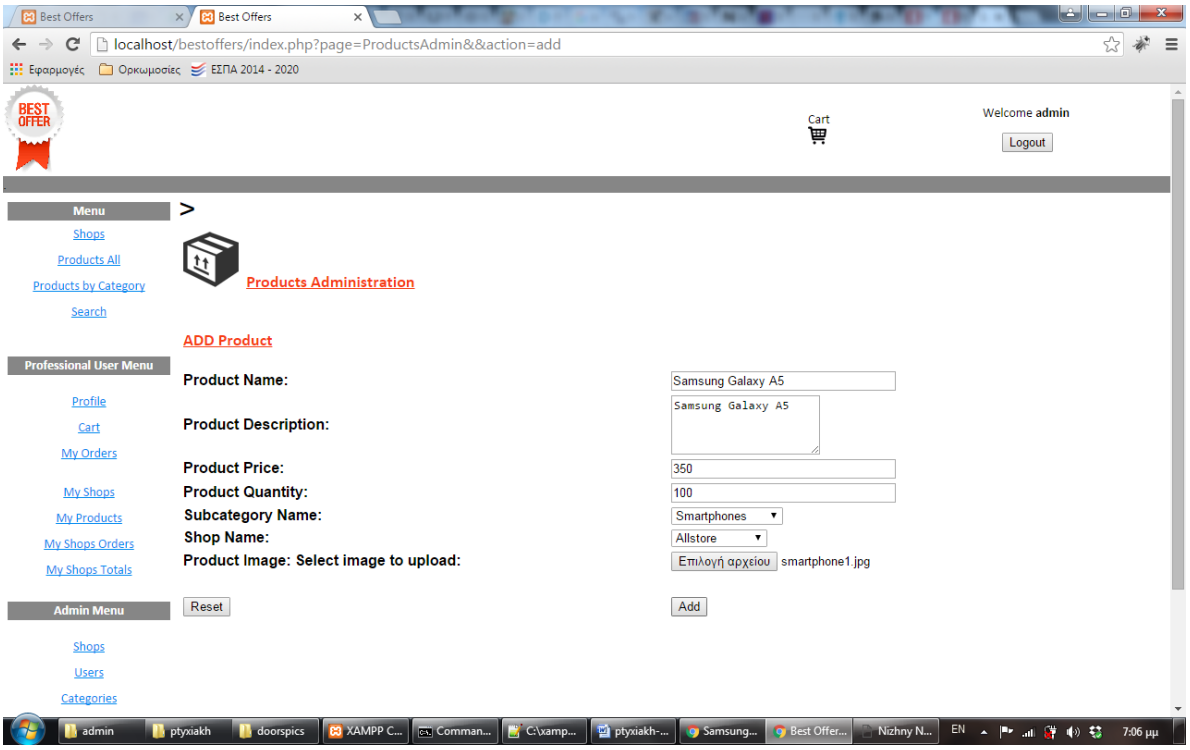


Εικόνα 31: Αρχική Οθόνη Προϊόντων

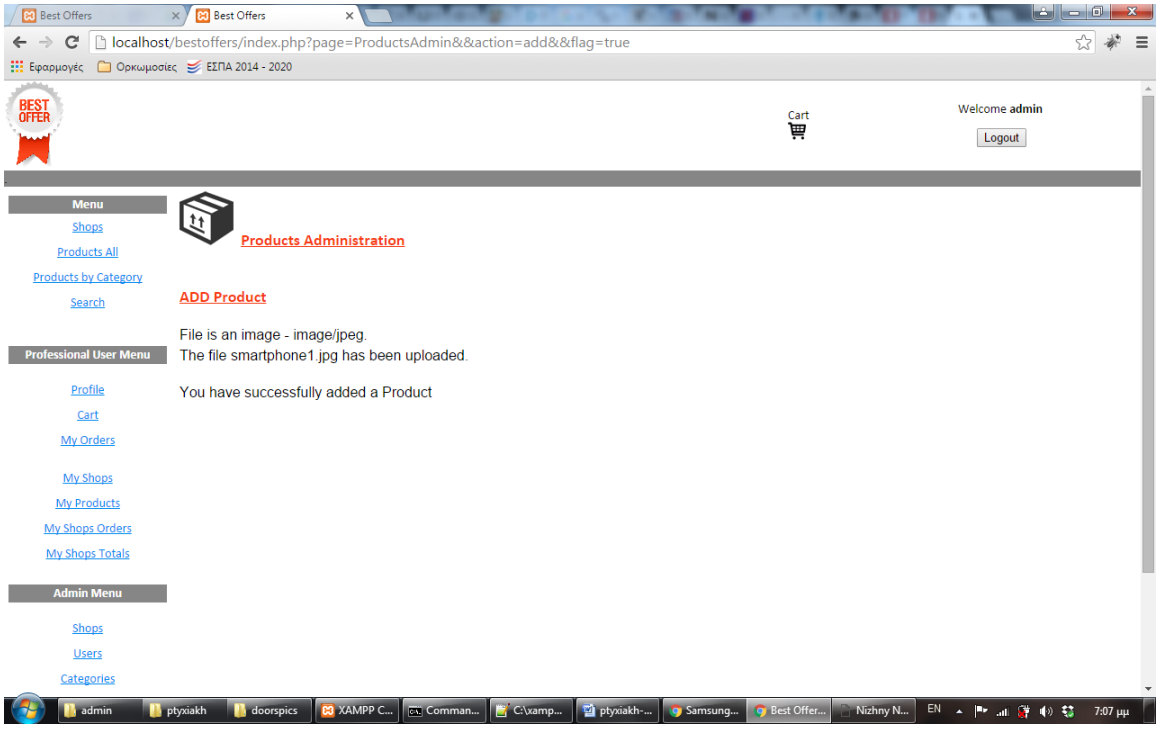
8.7.2 Δημιουργία Προϊόντος

Screenshots:

Στην οθόνη Δημιουργίας Προϊόντος, εμφανίζεται η φόρμα με τα απαραίτητα πεδία που πρέπει να συμπληρωθούν ώστε να δημιουργηθεί ένας προϊόν.



Εικόνα 32: Φόρμα Προσθήκης Νέου Προϊόντος



Εικόνα 33: Ολοκλήρωση προσθήκης Νέου Προϊόντος

SQL query:

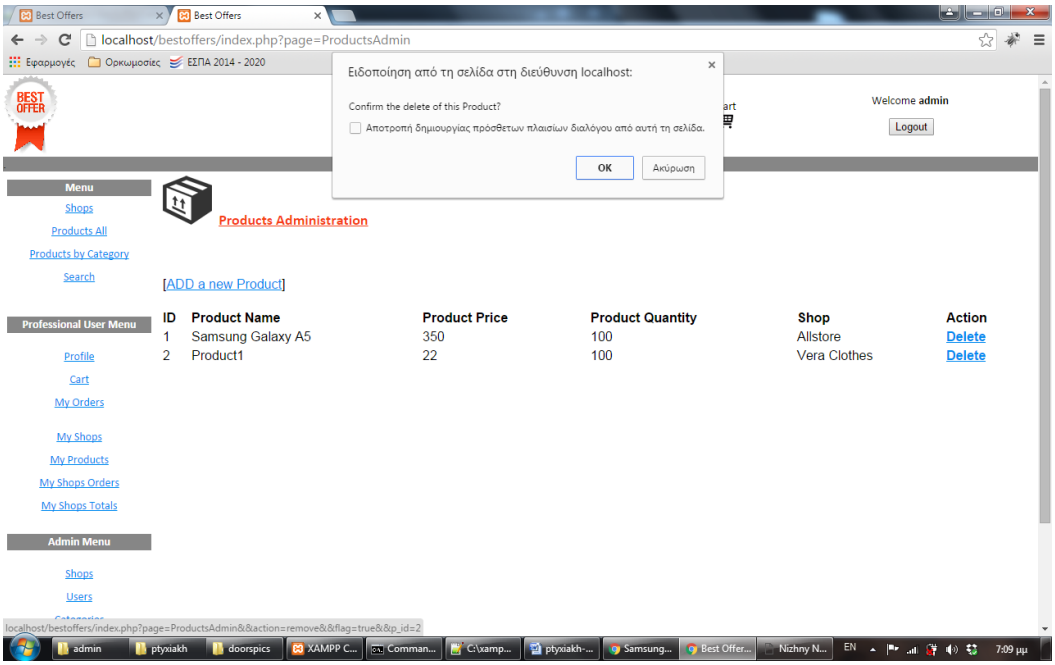
insert into products(p_name, p_price, p_description, p_quantity, p_imagepath, subcat_id, s_id)

values ('$p_name', $p_price, '$p_description', $p_quantity, '$target_file', $subcat_id, $s_id)

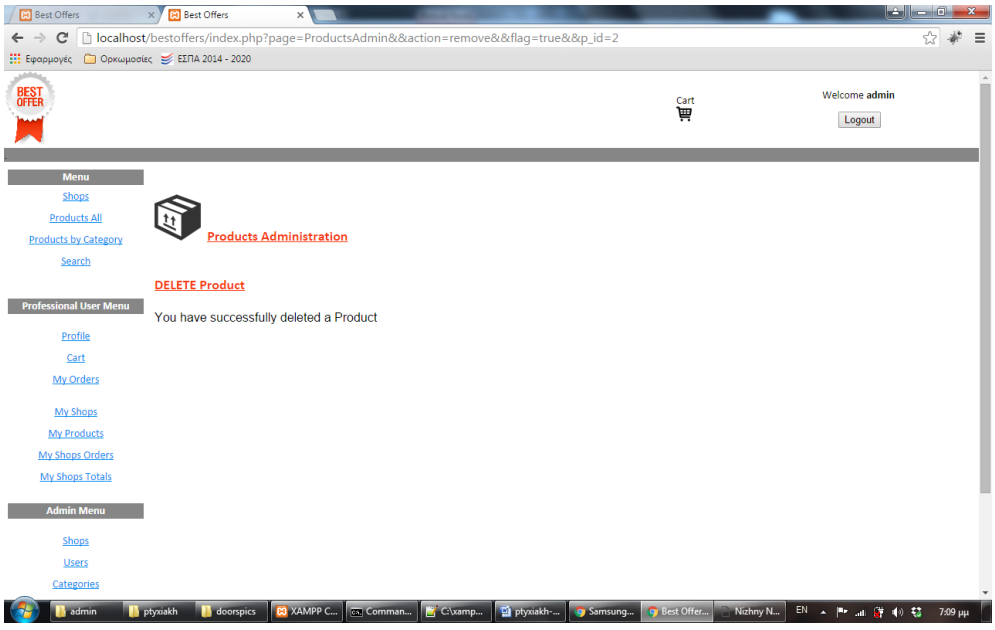
8.7.3 Διαγραφή Προϊόντος

Screenshots:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής του συγκεκριμένου Προϊόντος, περνώντας το id του ως παράμετρο.



Εικόνα 34: Διαγραφή Προϊόντος 1



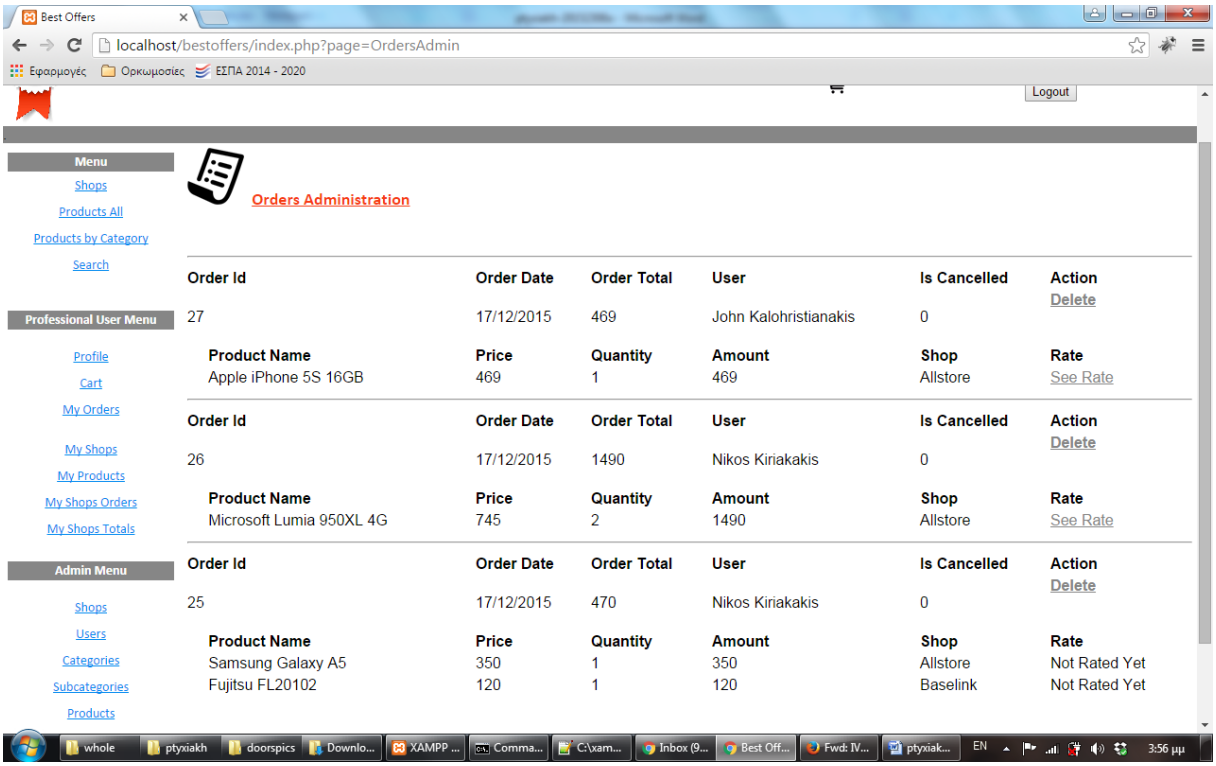
Εικόνα 35: Διαγραφή Προϊόντος 2

SQL query:

delete from products where p_id=$StrtsID

8.8 Παραγγελίες Διαχειριστή

Σε αυτή την οθόνη, ο διαχειριστή βλέπει όλες τις παραγγελίες που έχουν γίνει από όλους τους χρήστες και έχει τη δυνατότητα α) να τις διαγράψει β) να δει τις Αξιολογήσεις που έχουν καταχωρηθεί γ) να διαγράψει Αξιολόγηση με κακόβουλα σχόλια.



Εικόνα 36: Παραγγελίες - Περιβάλλον Διαχειριστή

8.9 Εγγραφή χρήστη

Ένας χρήστης της πλατφόρμας, μπορεί να εγγραφεί / δημιουργήσει λογαριασμό στην πλατφόρμα μέσω αυτής της λειτουργίας.

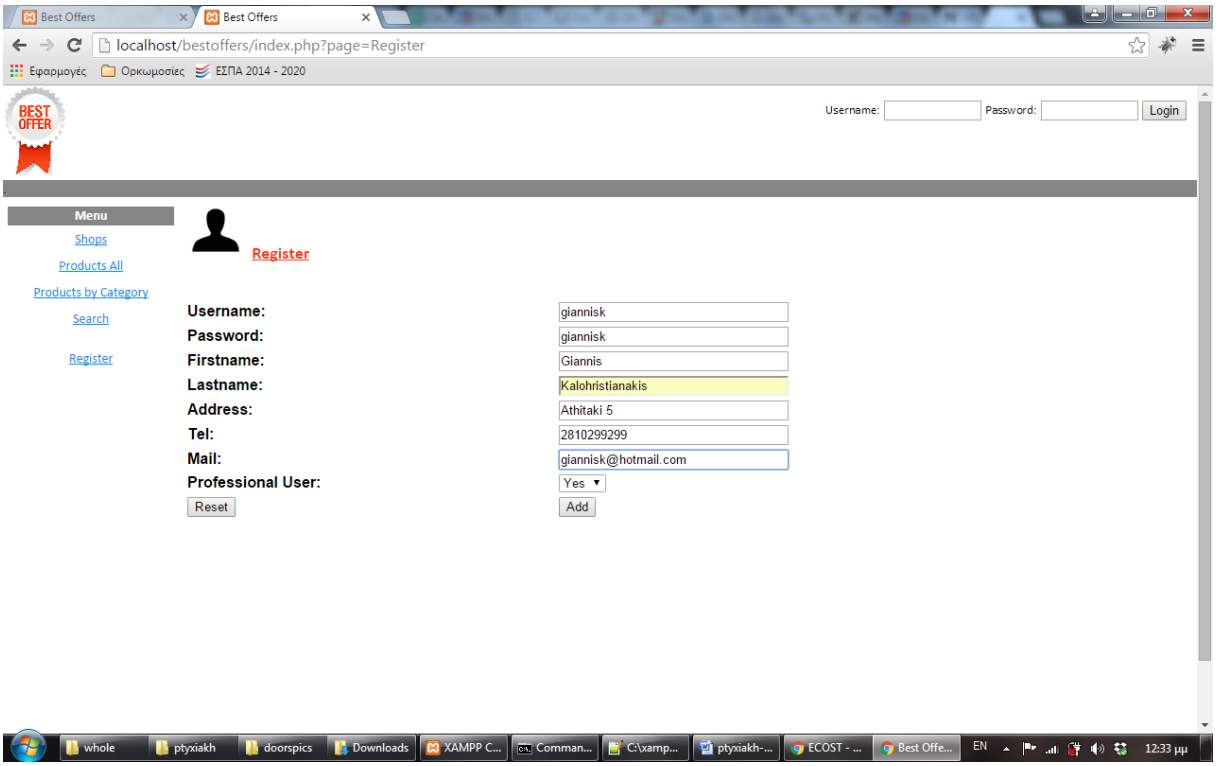
Η εγγραφή / δημιουργία λογαριασμού θα του δώσει τη δυνατότητα:

Αν είναι απλός χρήστης

- να πραγματοποιεί αγορές
- να βαθμολογεί καταστήματα από τα οποία αγόρασε
- να βλέπει τις πληροφορίες του προφίλ του

Επιπλέον, αν είναι επαγγελματίας χρήστης μπορεί:

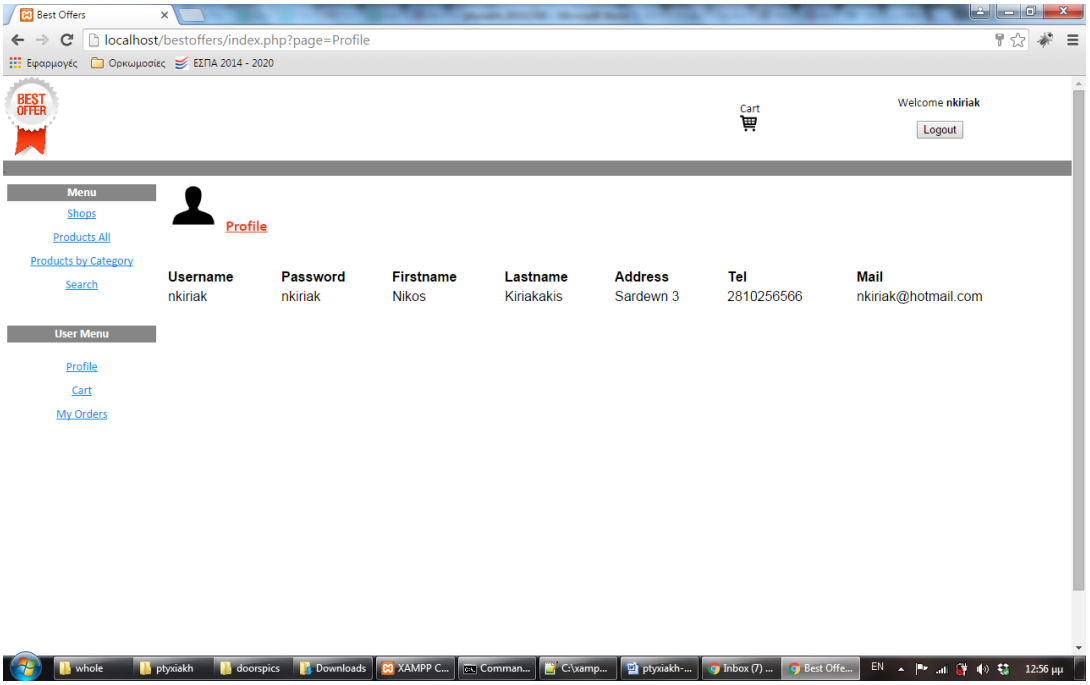
- Να προσθέτει ένα δικό του κατάστημα
- Να βλέπει τα προϊόντα του.
- Να προσθέτει προιόντα για τα καταστήματα του.
- Να βλέπει τις παραγγελίες των καταστημάτων του.
- Να βλέπει τα συνολικά στατιστικά των καταστημάτων του.



Εικόνα 37: Εγγραφή χρήστη

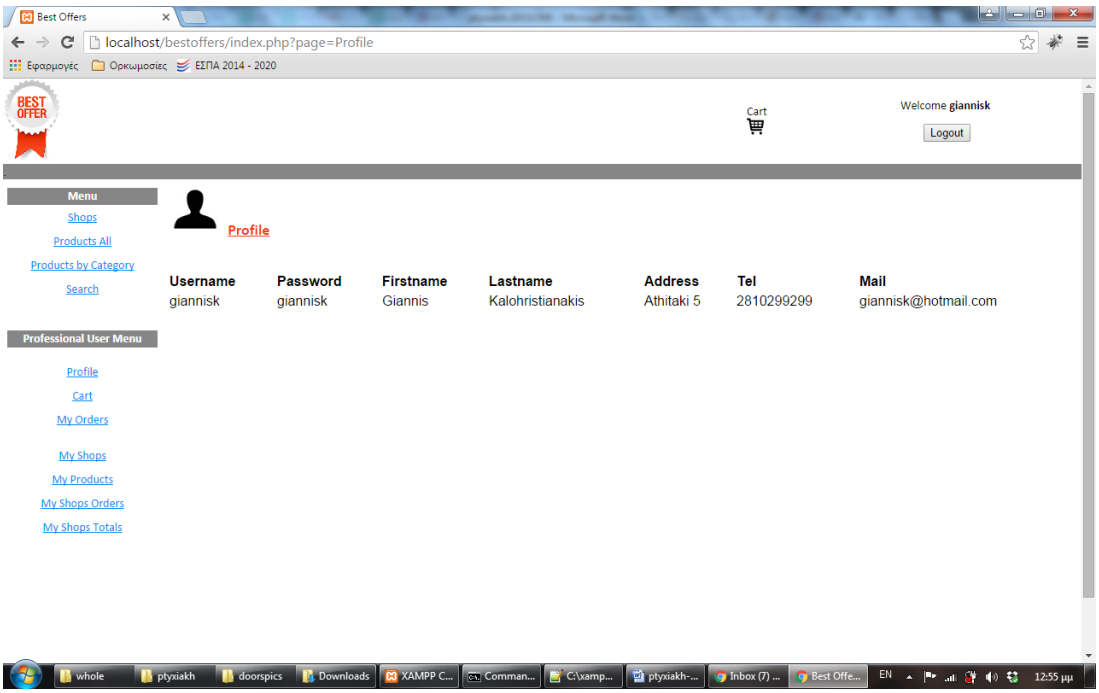
8.10 Σύνδεση στην πλατφόρμα

8.10.1 Σύνδεση απλού χρήστη



Εικόνα 38: Μενού απλού χρήστη

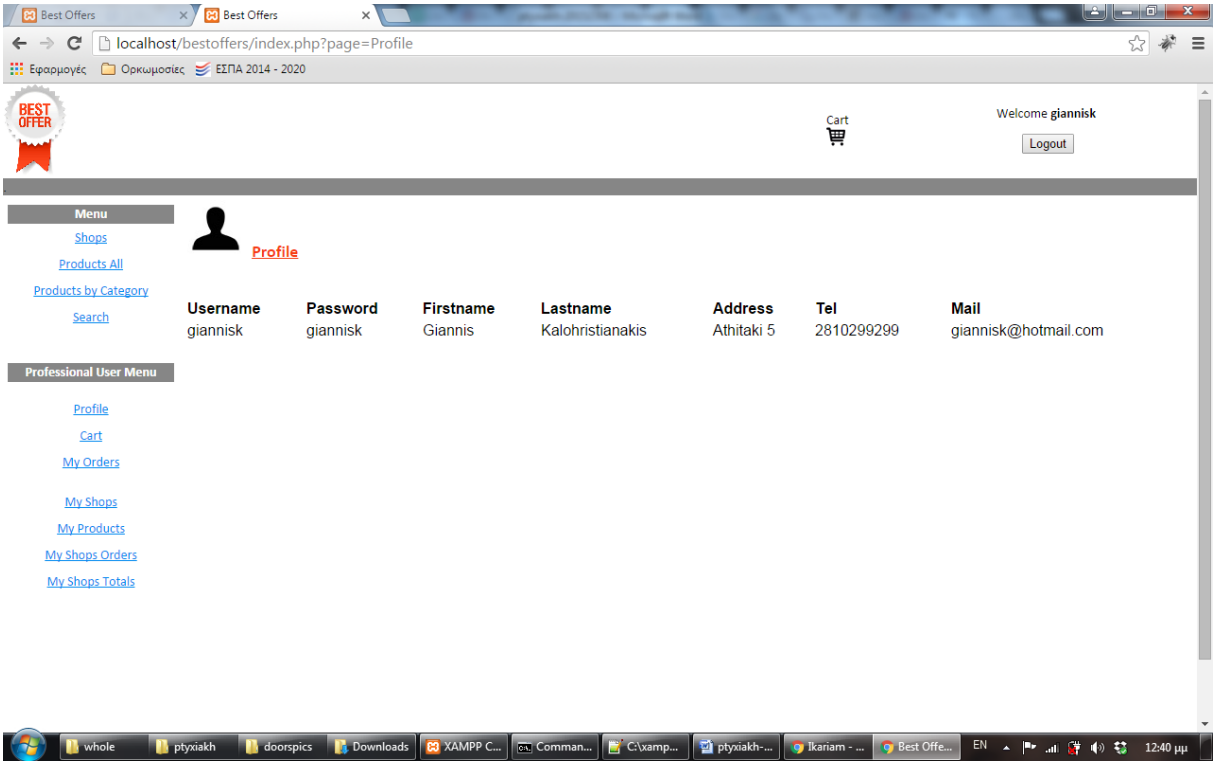
8.10.2 Σύνδεση επαγγελματία χρήστη



Εικόνα 39: Μενού επαγγελματία χρήστη

8.11 Προφίλ χρήστη

Ένας χρήστης της πλατφόρμας, μπορεί να να βλέπει τις πληροφορίες του προφίλ του, εφόσον έχει κάνει login στο σύστημα.



Εικόνα 40: Προφίλ χρήστη

8.12 Αναζήτηση

Ένας απλός χρήστης της πλατφόρμας, μπορεί να πραγματοποιήσει αναζήτηση για τα διαθέσιμα προϊόντα.

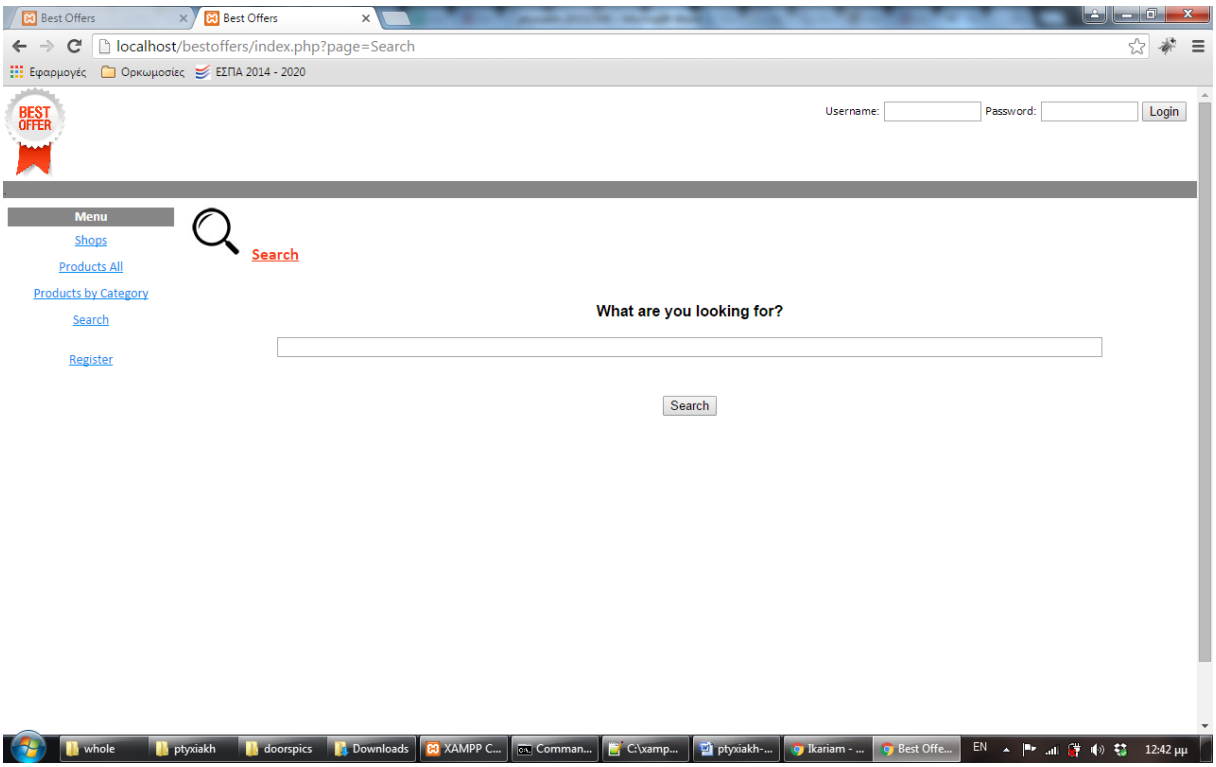
Αναζήτηση μπορεί να πραγματοποιηθεί από οποιονδήποτε επισκεφτεί την πλατφόρμα, χωρίς να χρειάζεται να έχει εγγραφει/δημιουργήσει λογαριασμό.

Τα αποτελέσματα αναζήτησης μπορεί να τα χρησιμοποιήσει ένας χρήστης για να προχωρήσει σε αγορά , μόνο εάν έχει λογαριασμό και είναι συνδεδεμένος. (Το κουμπί **Προσθήκη στο καλάθι** στα αποτελέσματα αναζήτησης εμφανίζεται μόνο εάν είναι συνδεδεμένος ένας χρήστης)

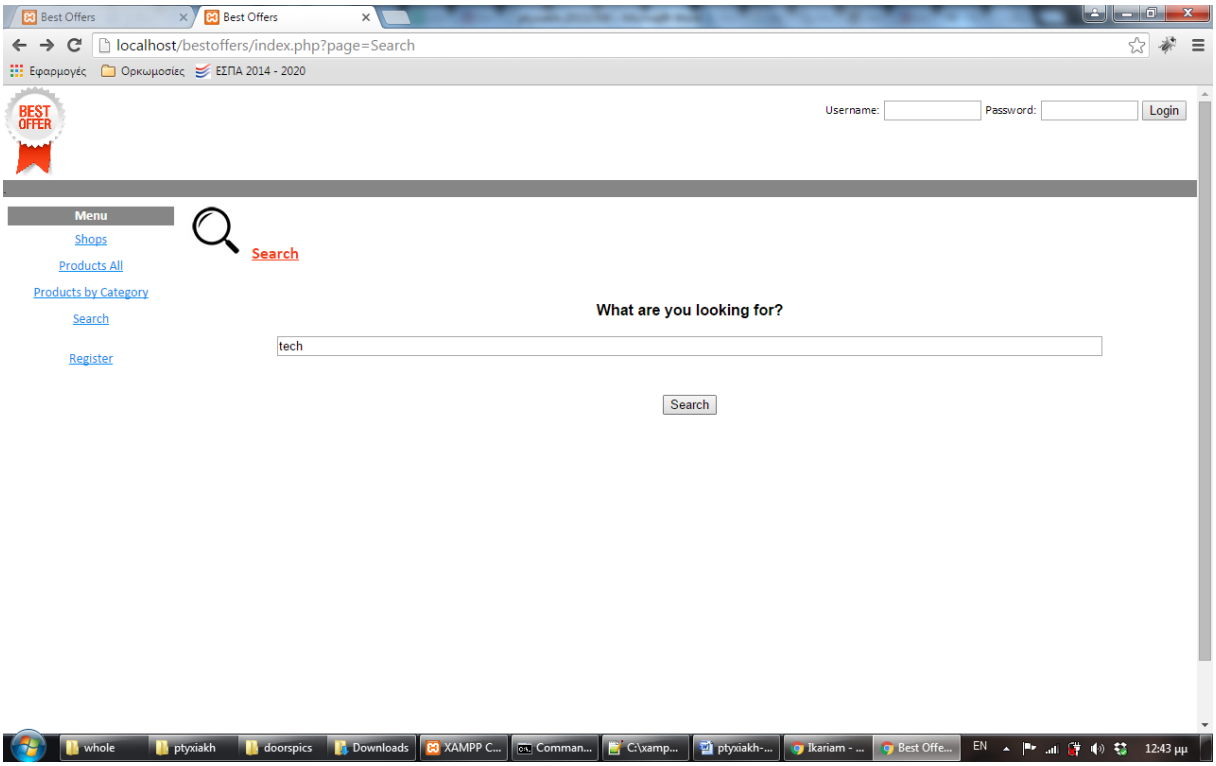
8.12.1 Κριτήρια Αναζήτησης

Κάθε χρήστης μπορεί να αναζητήσει προιόντα βάζοντας οποιοδήποτε όρο επιθυμεί. Αν ο όρος αναζήτησης υπάρχει ακριβώς ή ως μέρος σε κάποιο από τα παρακάτω:

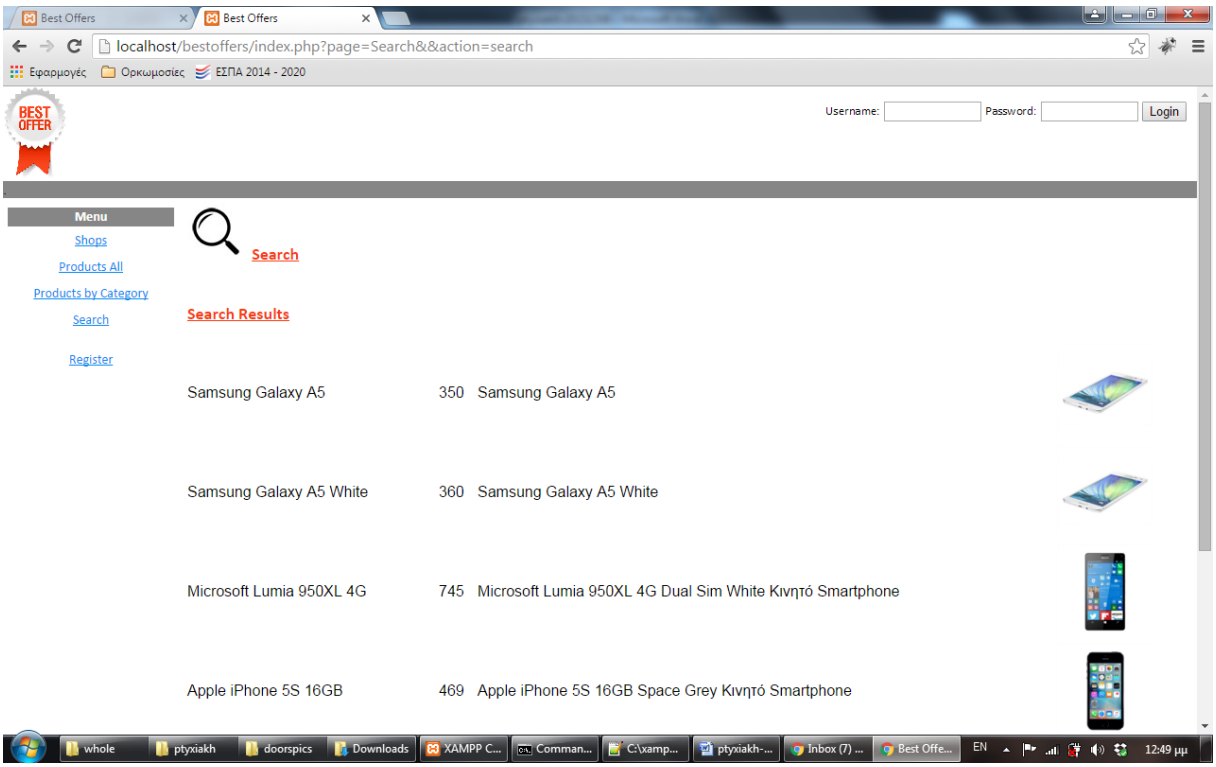
- Στο όνομα του προιοντος
- Στην κατηγορία των προϊόντων
- Στην υποκατηγορία των προϊόντων
- Στην περιγραφή του προϊόντος



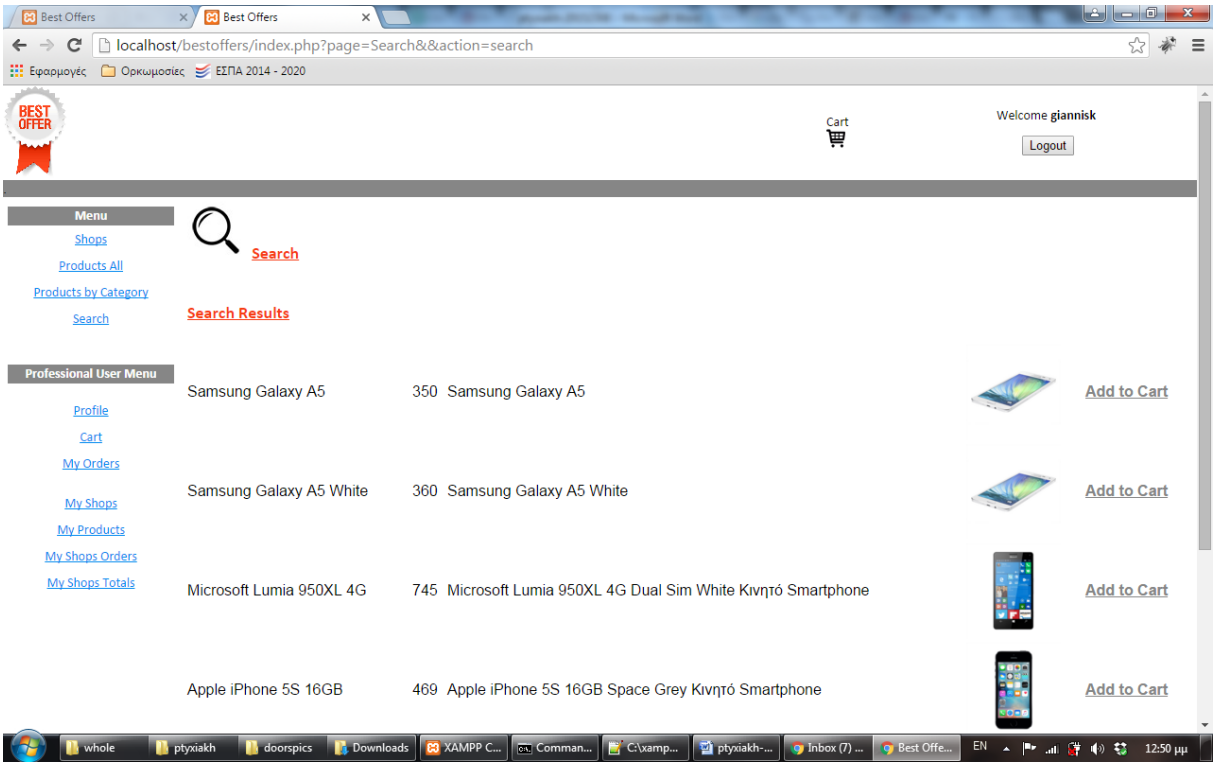
Εικόνα 41: Αρχική Οθόνη Αναζήτησης (μη συνδεδεμένος)



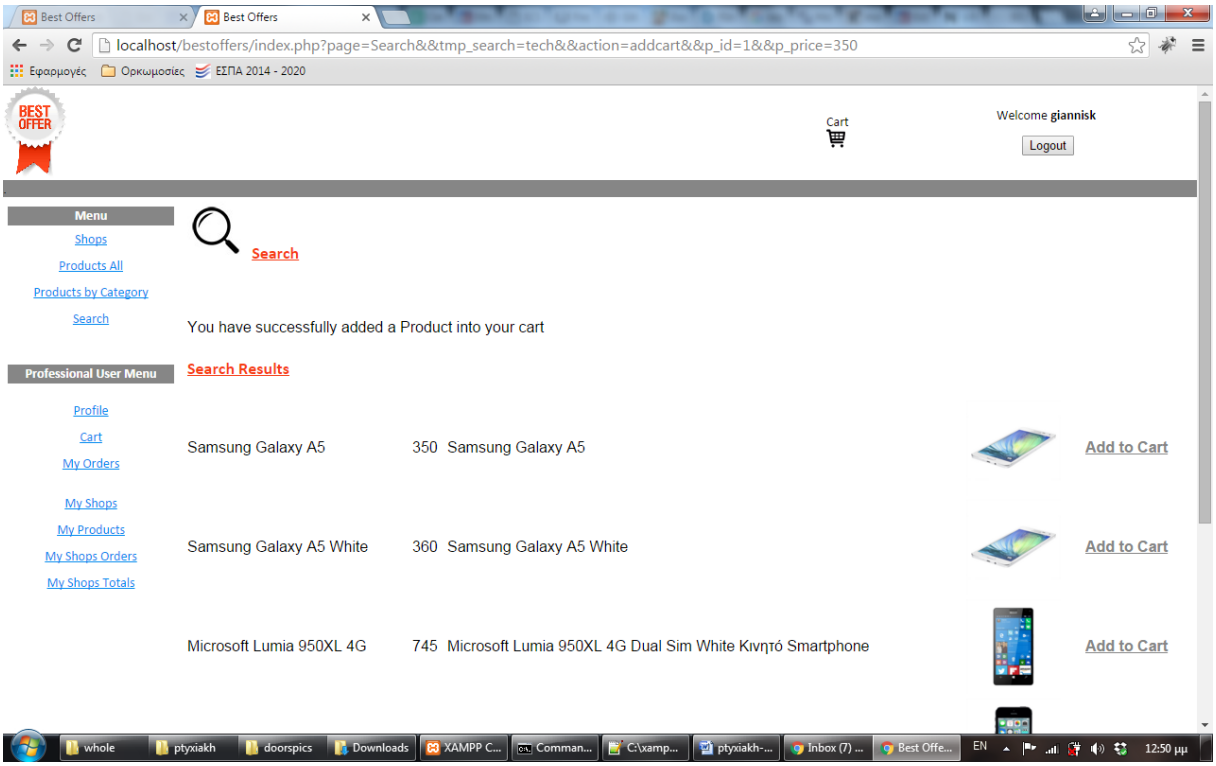
Εικόνα 42: Αναζήτηση του όρου tech



Εικόνα 43: Επιστροφή αποτελεσμάτων που περιέχουν είτε στο όνομα, είτε στην περιγραφή, είτε στην κατηγορία, είτε στην υποκατηγορίατον όρο tech (μη συνδεδεμένος)



Εικόνα 44: Επιστροφή αποτελεσμάτων που περιέχουν είτε στο όνομα, είτε στην περιγραφή, είτε στην κατηγορία, είτε στην υποκατηγορίατον όρο tech (συνδεδεμένος)



Εικόνα 45: Προσθήκη στο καλάθι από τα αποτελέσματα αναζήτησης και διατήρηση αυτών

SQL queries:

ΑΝΑΖΗΤΗΣΗ ΠΡΟΪΟΝΤΩΝ

```
SELECT DISTINCT p_id, p_name, p_price, p_description, p_imagepath FROM products a, subcategories b, categories c

where a.subcat_id=b.subcat_id and b.cat_id=c.cat_id  and

        (c.cat_id in

                (select cat_id from categories where cat_name LIKE '%".$tmp_search."%')

        or b.subcat_id in

                (select subcat_id from subcategories where subcat_name LIKE '%".$tmp_search."%')

        or a.p_name LIKE '%".$tmp_search."%'

        or a.p_description LIKE '%".$tmp_search."%')
```

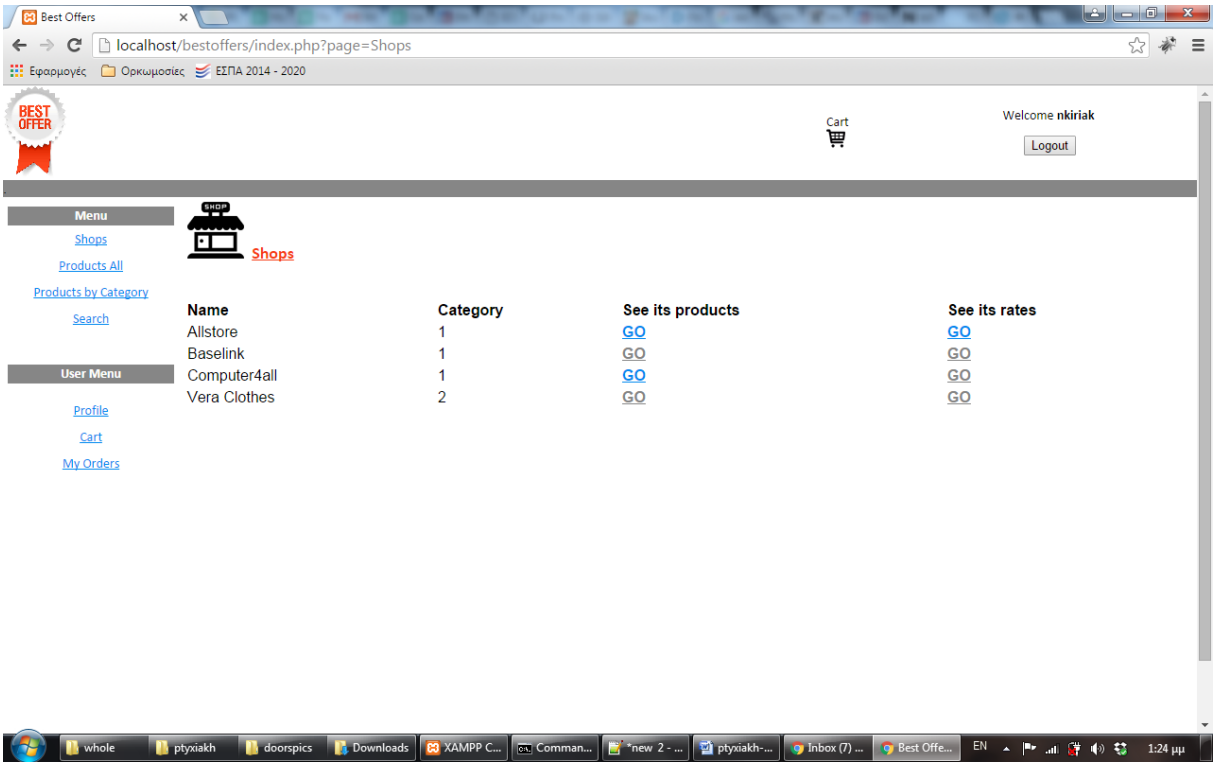
8.13 Όλα τα Καταστήματα

8.13.1 Αρχική Οθόνη Όλα τα Καταστήματα

Screenshots:

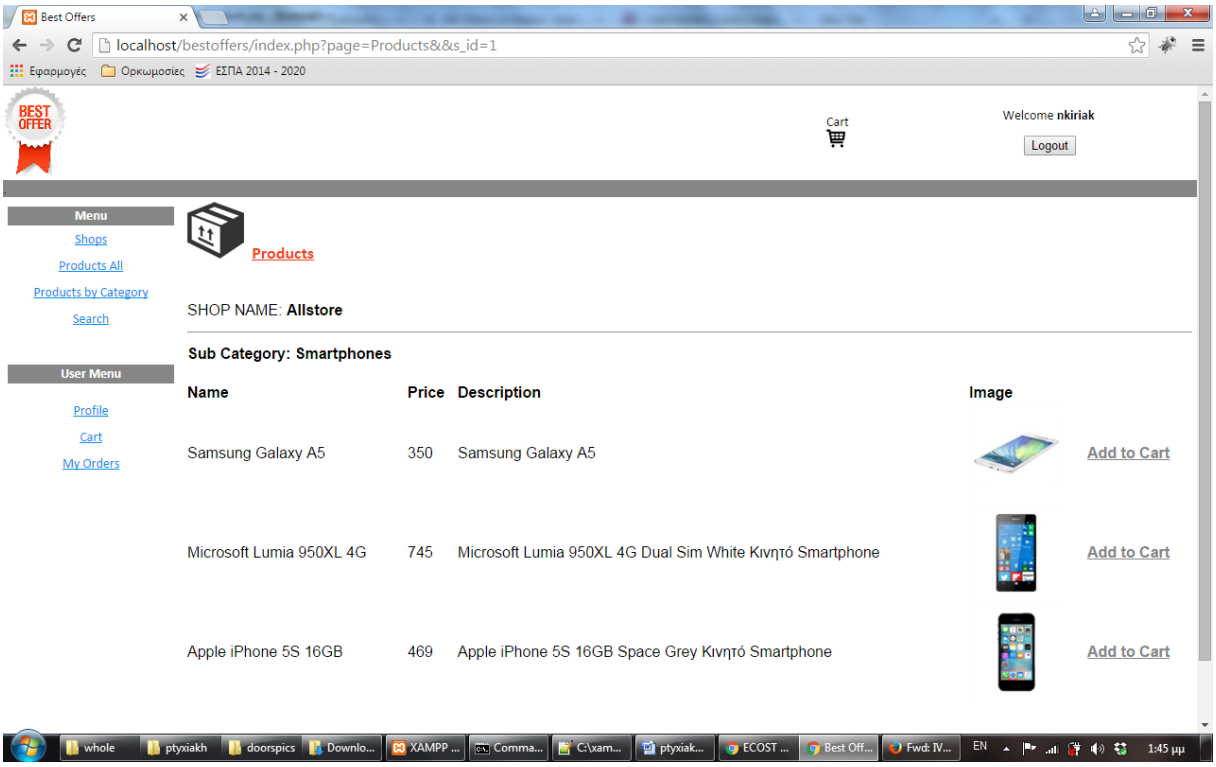
Στην αρχική οθόνη όλων των καταστημάτων εμφανίζεται η λίστα με τα υπάρχοντα καταστήματα στην πλατφόρμα και δίπλα σε κάθε ένα, οι σύνδεσμοι **Δες τα Προιόντα του** και **Δες τις Αξιολογήσεις του** για κάθε ένα από αυτά.

Εάν δεν έχει δημιουργηθεί κάποιο κατάστημα, τότε εμφανίζεται μήνυμα μη ύπαρξης Καταστημάτων.

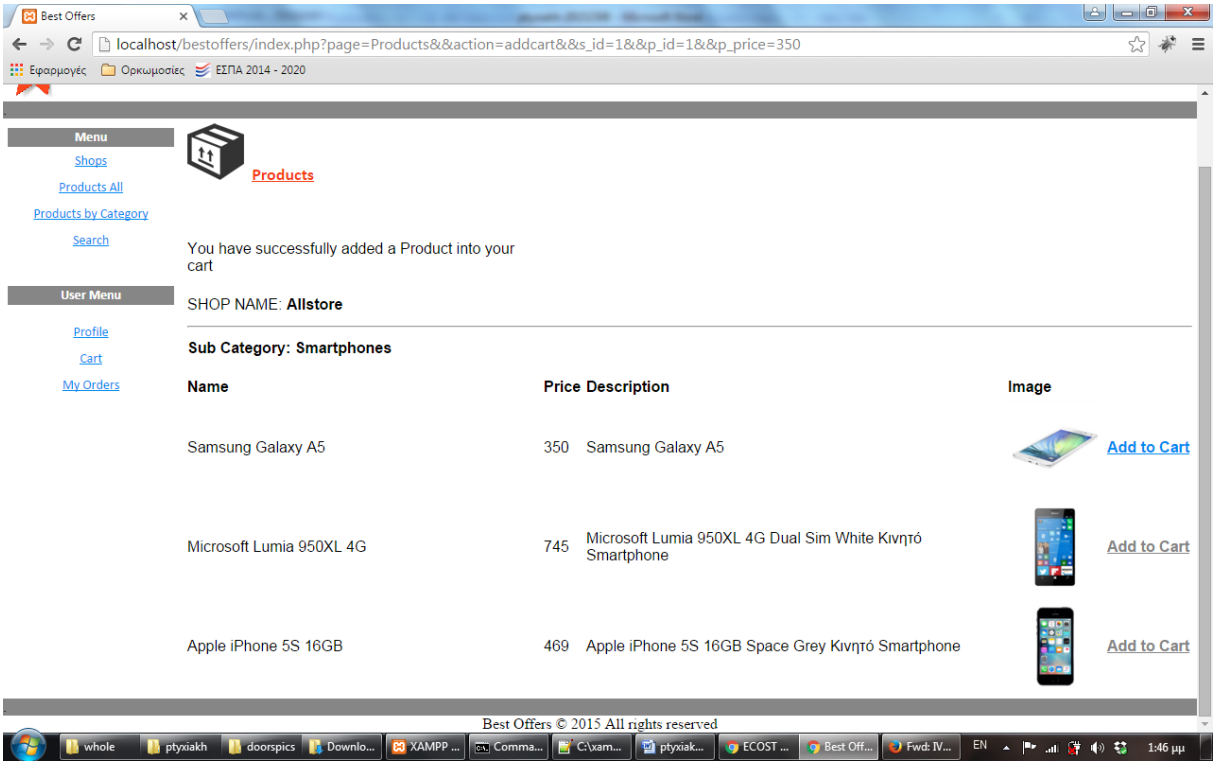


Εικόνα 46: Αρχική Οθόνη όλων των καταστημάτων

8.13.2 Παρουσίαση προϊόντων καταστήματος



Εικόνα 47: Παρουσίαση Προϊόντων ενός συγκεκριμένου καταστήματος

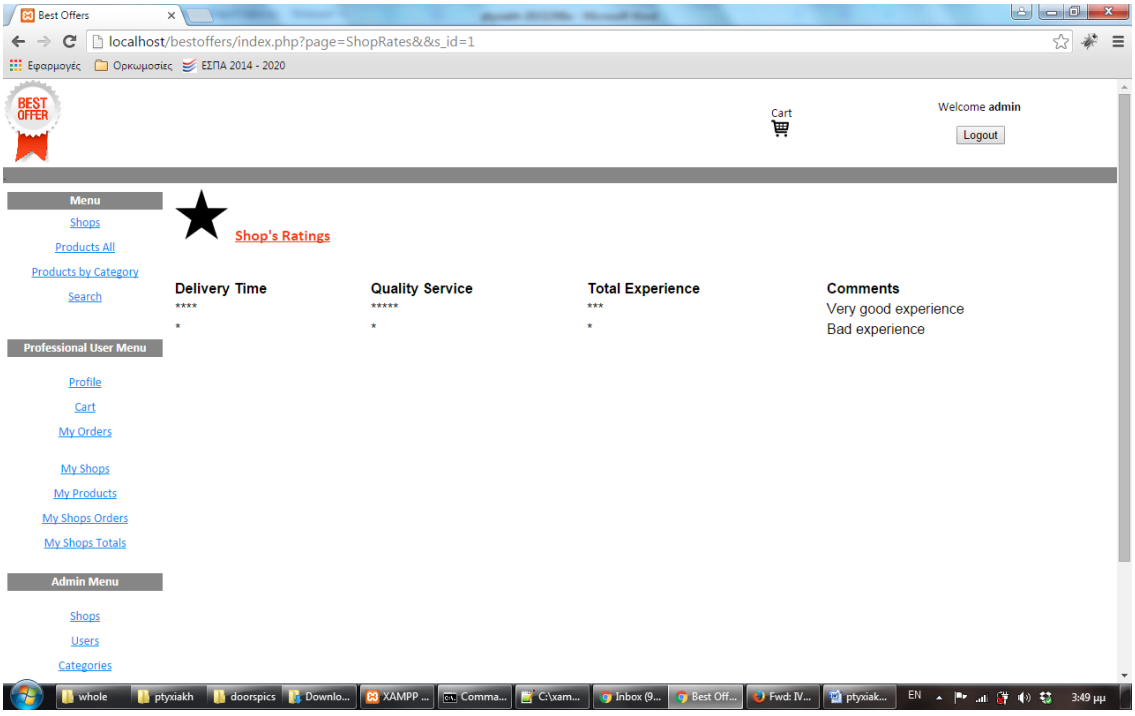


Εικόνα 48: Προσθήκη ενός προϊόντος στο καλάθι και διατήρηση των προϊόντων του καταστήματος που επιλέξαμε

SQL query:

```
//show products of a specific shop
if (isset($_GET['s_id'])){
        if($_GET['s_id']!=-1){
                $y=$_GET['s_id'];
                $query = "SELECT * FROM products where s_id=$y and subcat_id=$temp3[0]";
                $db=mysql_query($query);
        }
```

8.13.3 Παρουσίαση όλων των αξιολογήσεων για συγκεκριμένο κατάστημα



Εικόνα 49: Παρουσίαση όλων των αξιολογήσεων για ένα κατάστημα

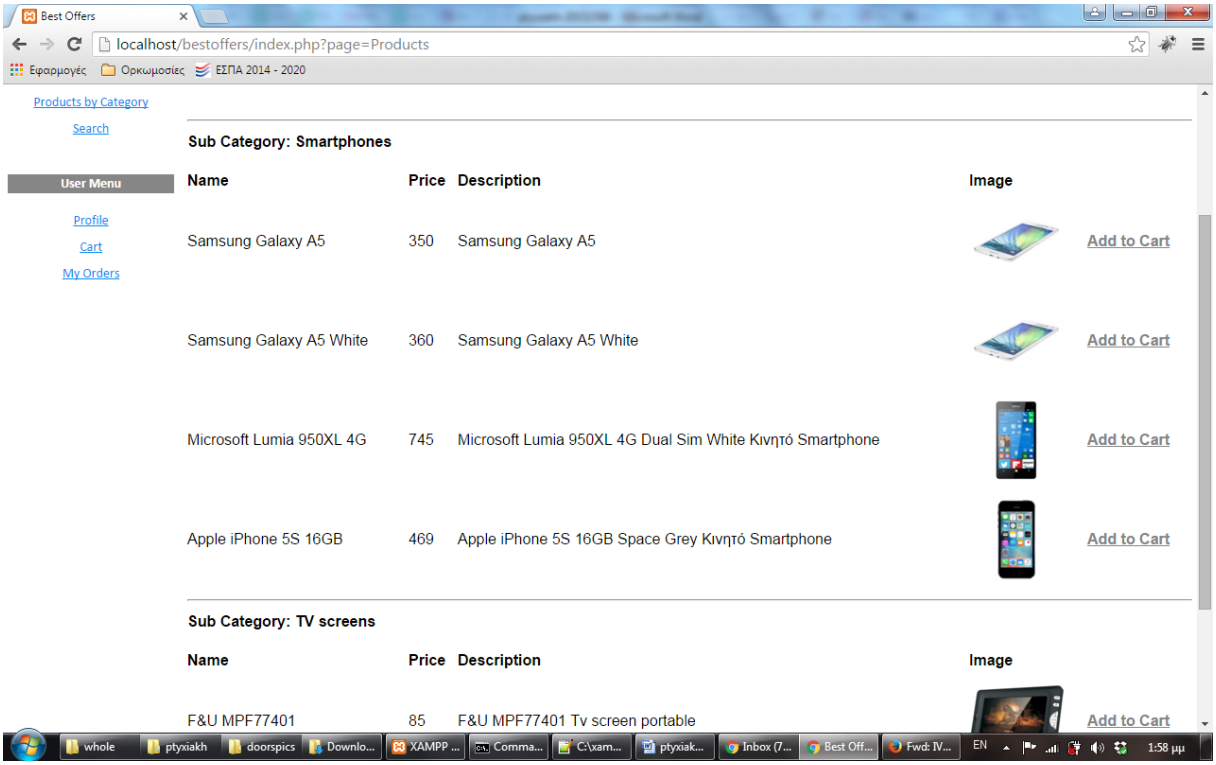
SQL Query:

SELECT * FROM rates where r_s_id=$s_id

8.14 Όλα τα προϊόντα

Screenshots:

Στην αρχική οθόνη των Προϊόντων εμφανίζεται η λίστα με όλα τα υπάρχοντα Προϊόντα ταξινομημένα ανά Υποκατηγορία και δίπλα σε κάθε ένα, ο σύνδεσμος Προσθήκη στο καλάθι (Add to cart) για κάθε ένα από αυτά (εφόσον ο χρήστης έχει συνδεθεί).



Εικόνα 50: Αρχική Οθόνη Προϊόντων

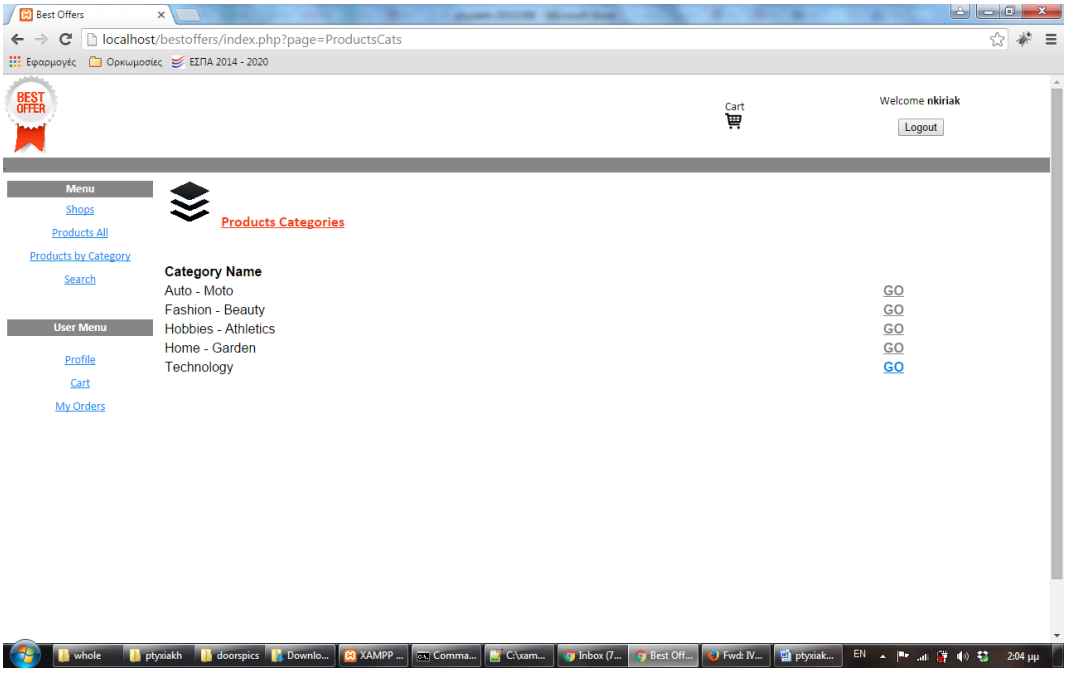
SQL query:

SELECT * FROM products where subcat_id=$temp3[0]

8.15 Προϊόντα ανα κατηγορία

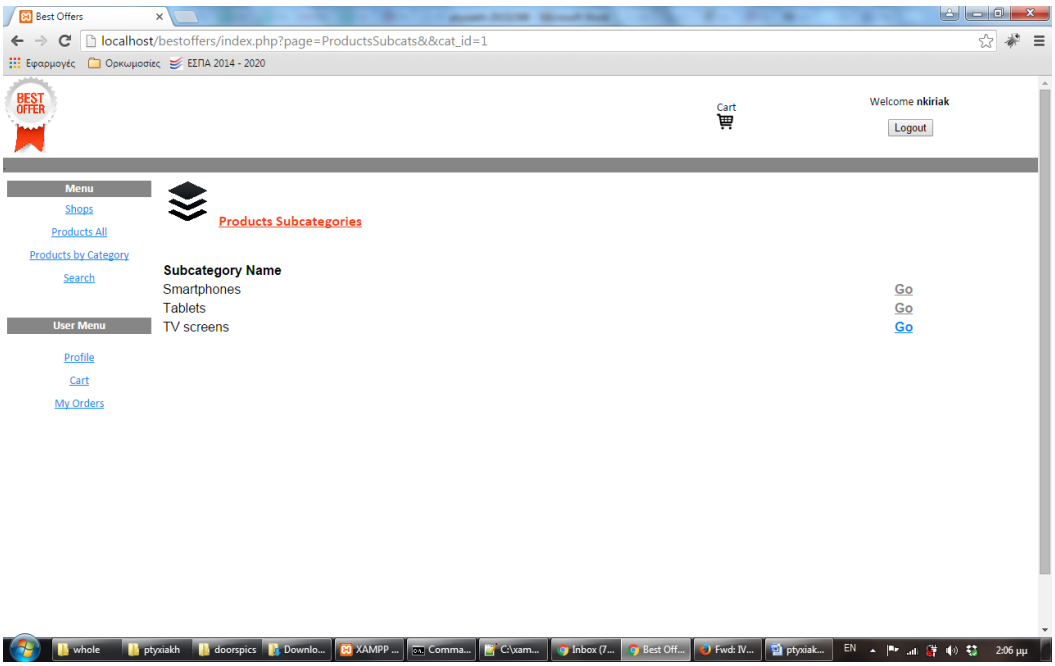
Screenshots:

Στην αρχική οθόνη των Προϊόντων ανά κατηγορία εμφανίζεται η λίστα με τις διαθέσιμες κατηγορίες Προϊόντων και δίπλα σε κάθε μια, ο σύνδεσμος για να προωθηθείς στις υποκατηγορίες αυτής.



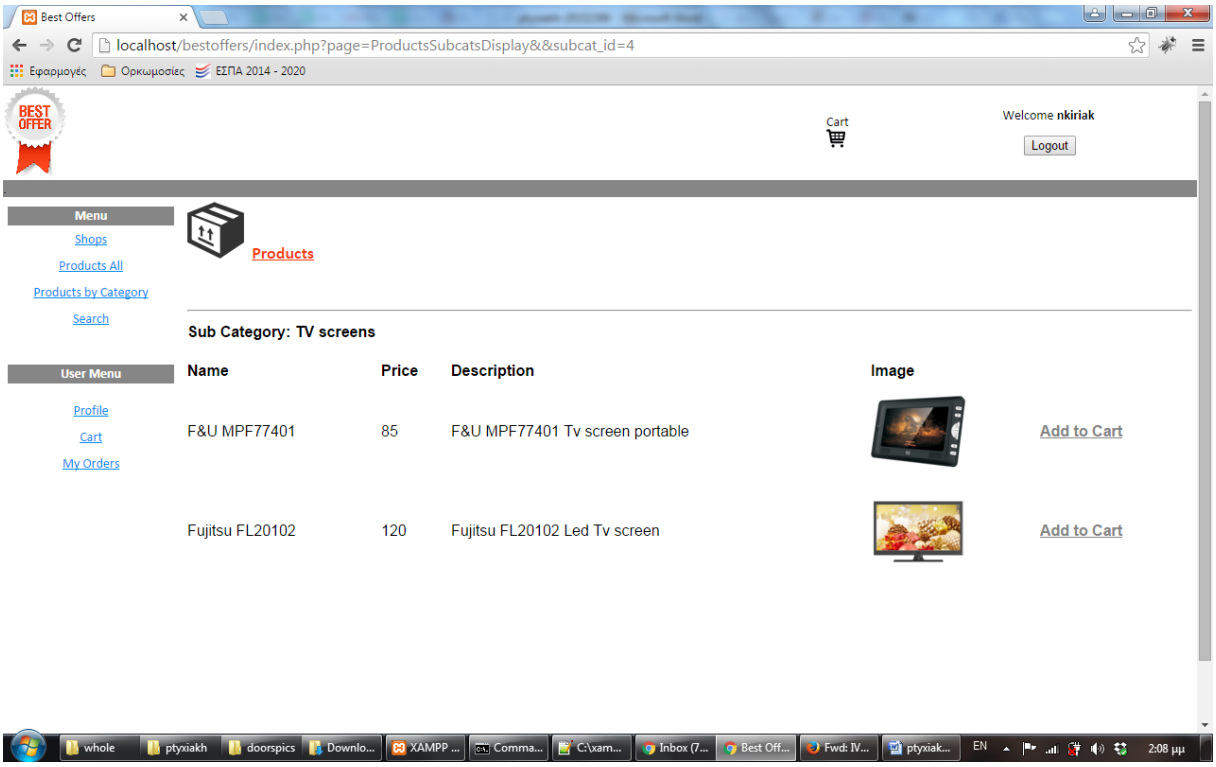
Εικόνα 51: Αρχική Οθόνη Προϊόντων ανά κατηγορία

Αν επιλέξει ο χρήστης μια κατηγορία, τότε εμφανίζονται όλες οι υποκατηγορίες προϊόντων που έχει αυτή η κατηγορία διαθέσιμες.

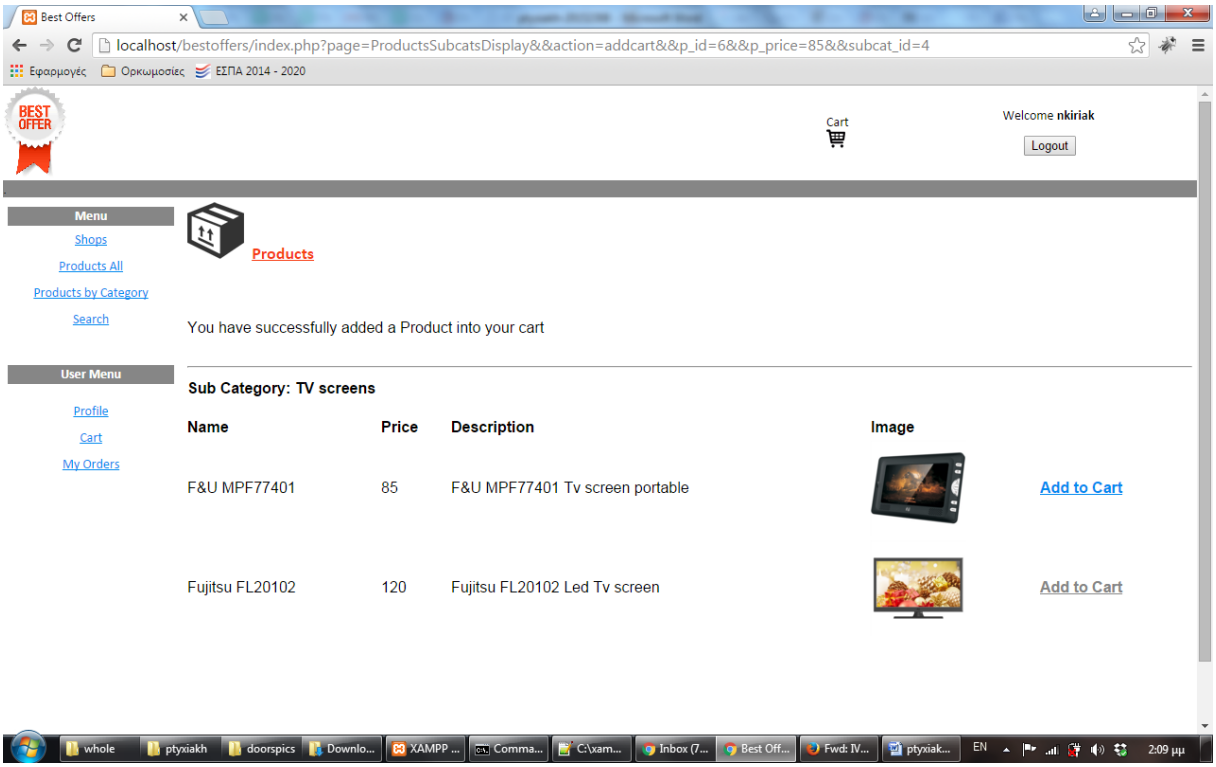


Εικόνα 52: Επιλογή κατηγορίας Προϊόντων – Εμφάνιση Υποκατηγοριών

Αν επιλέξει ο χρήστης μια υποκατηγορία, τότε εμφανίζονται όλα τα προϊόντα που έχει αυτή η υποκατηγορία διαθέσιμα.



Εικόνα 53: Επιλογή υποκατηγορίας Προϊόντων – Εμφάνιση Προϊόντων Υποκατηγοριας



Εικόνα 54: Προσθήκη ενός προιόντος στο καλάθι και διατήρηση των προιόντων της υποκατηγορίας που επιλέξαμε

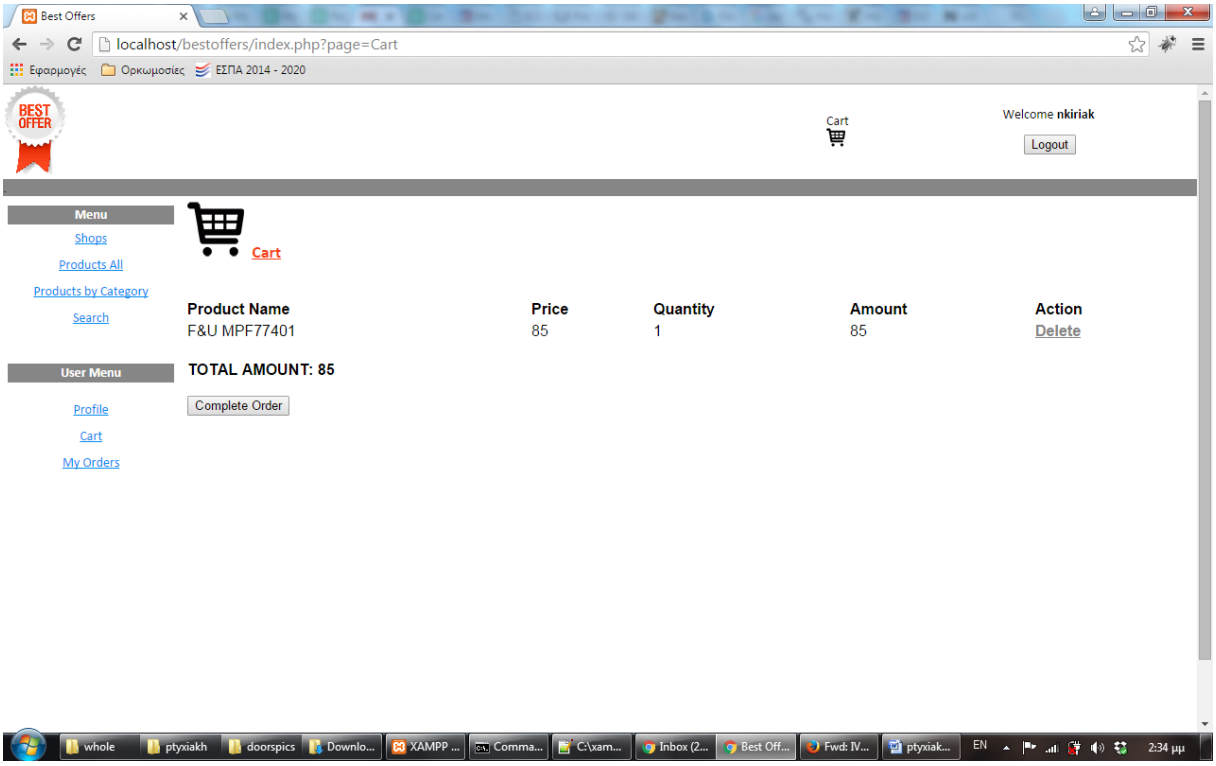
8.16 Καλάθι

8.16.1 Αρχική Οθόνη Καλαθιού

Screenshots:

Στην αρχική οθόνη του Καλαθιού εμφανίζεται η λίστα με τα υπάρχοντα Προϊόντα που ο χρήστης έχει προσθέσει στο καλάθι και δίπλα σε κάθε ένα, ο σύνδεσμος Διαγραφής (Delete) για κάθε ένα από αυτά.

Εάν δεν έχει προστεθεί κάποιο προιόν στο καλάθι, τότε εμφανίζεται μήνυμα άδειου καλαθιού.

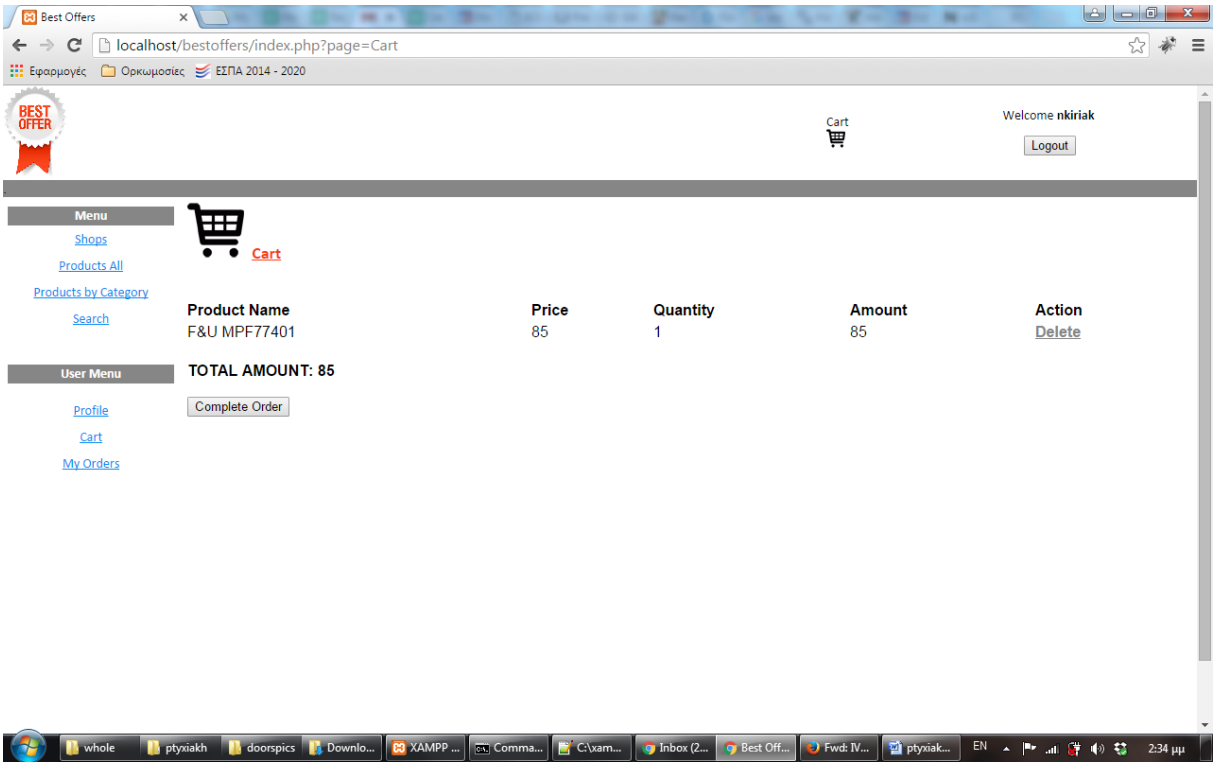


Εικόνα 55:Αρχική Οθόνη Καλαθιού

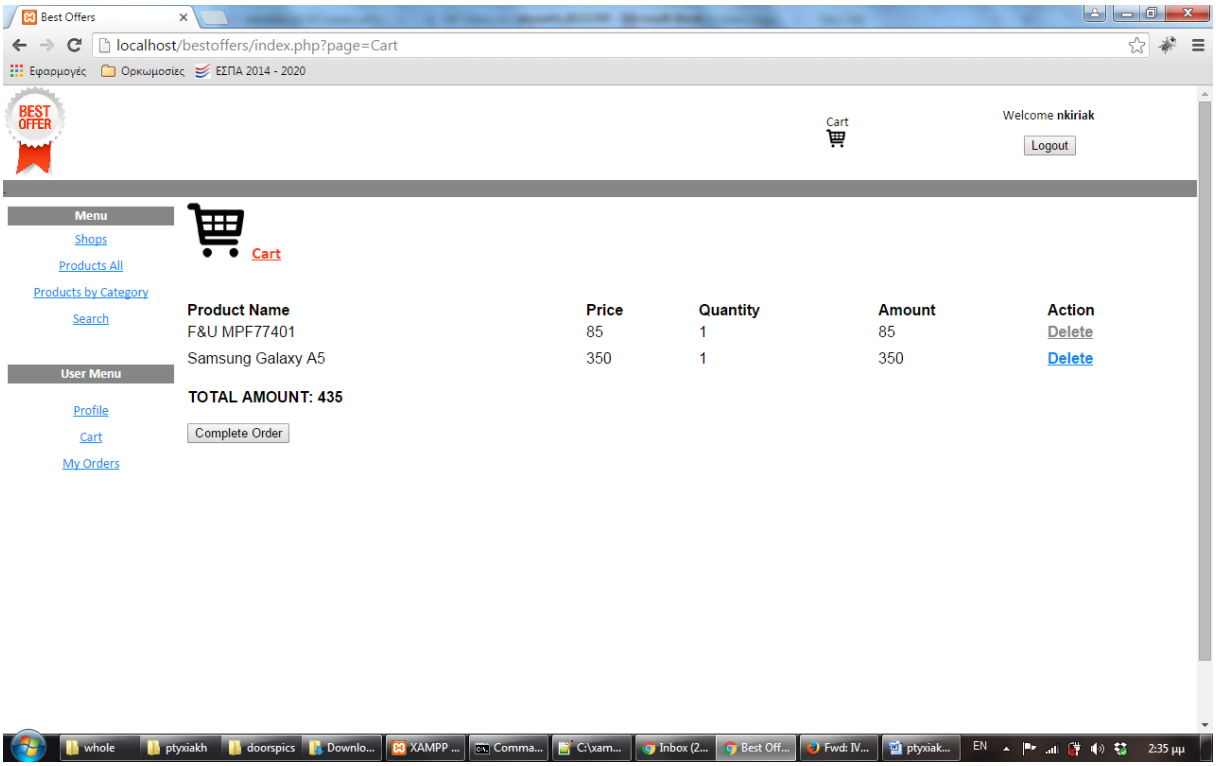
8.16.2 Προσθήκη στο καλάθι

Screenshots:

Κάθε συνδεδεμένος χρήστης έχει το καλάθι του στο οποίο εμφανίζονται τα προιόντα που προσθέτει αυτός και ενημερώνεται για το σύνολο της παραγγελίας του. Αν κάποιο προιόν επιλεχθεί, δευτερη φορά, τότε ανανεώνεται η ποσότητα του προιόντος στο καλάθι και το μερικό σύνολο για αυτό το προιον. Στο καλάθι έχει επίσης τη δυνατότητα να πατήσει το κουμπί **Ολοκλήρωση Παραγγελίας (Complete Order).** Αν μια παραγγελία ολοκληρωθεί, τότε αυτόματα διαγράφονται όλα τα προιοντα από το καλάθι.



Εικόνα 56: Καλάθι 1



Εικόνα 57: Καλάθι 2

SQL query:

```
 if ($_GET['action']=="addcart"){


        $x=$_SESSION['userID'];
        $cart_p_id = $_GET['p_id'];


        $p_price = $_GET['p_price'];


        //prepei na perase kai to quantity
        //$p_cart_quantity = $_POST['p_quantity'];

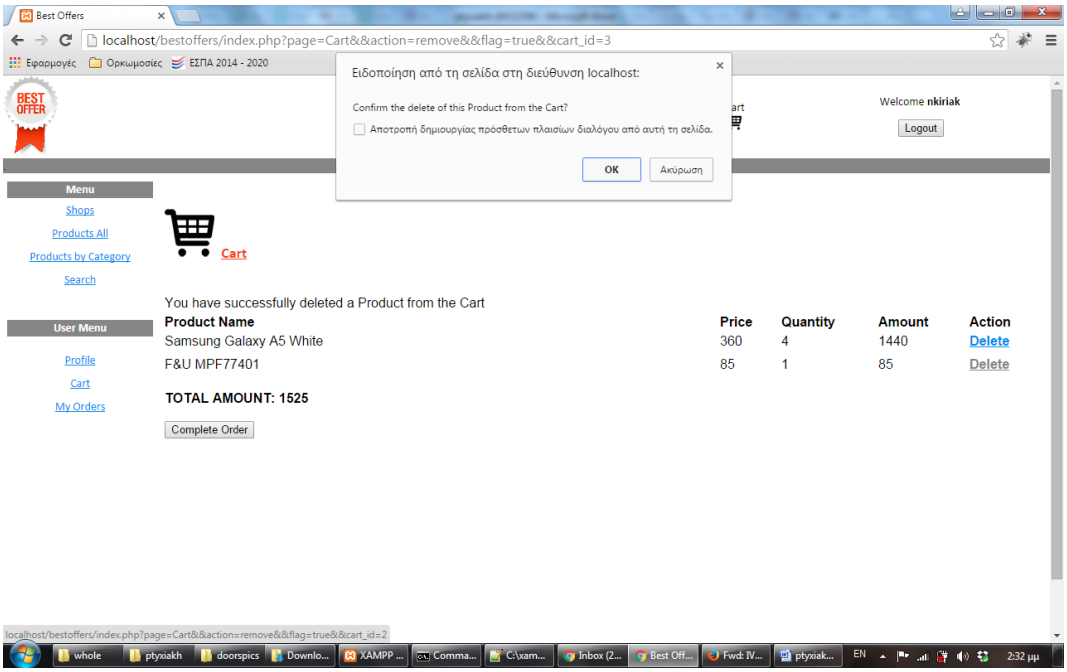

        //check if the first add of the product or not
        $query = "SELECT * FROM cart where cart_userID=$x and cart_p_id=$cart_p_id";
        $db=mysql_query($query);
        $test = mysql_num_rows($db);


        //First add of the product in the cart of the user
        if ($test==0){
                $query2="insert into cart(cart_p_id, cart_p_quantity, cart_p_price, cart_userID)
                        values ($cart_p_id, 1, $p_price, $x)";
                $db2=mysql_query($query2);
        }
        //Second, third, etc add of the product in the cart of the user /update quantity
        else{
                $temp=mysql_fetch_row($db);
                $query2="update cart set cart_p_quantity=cart_p_quantity+1 where
cart_id=$temp[0]";
                $db2=mysql_query($query2);
        }
echo( "<tr><td colspan=\"5\"></br>You have successfully added a Product into your
cart</br></br></td></tr>");
```

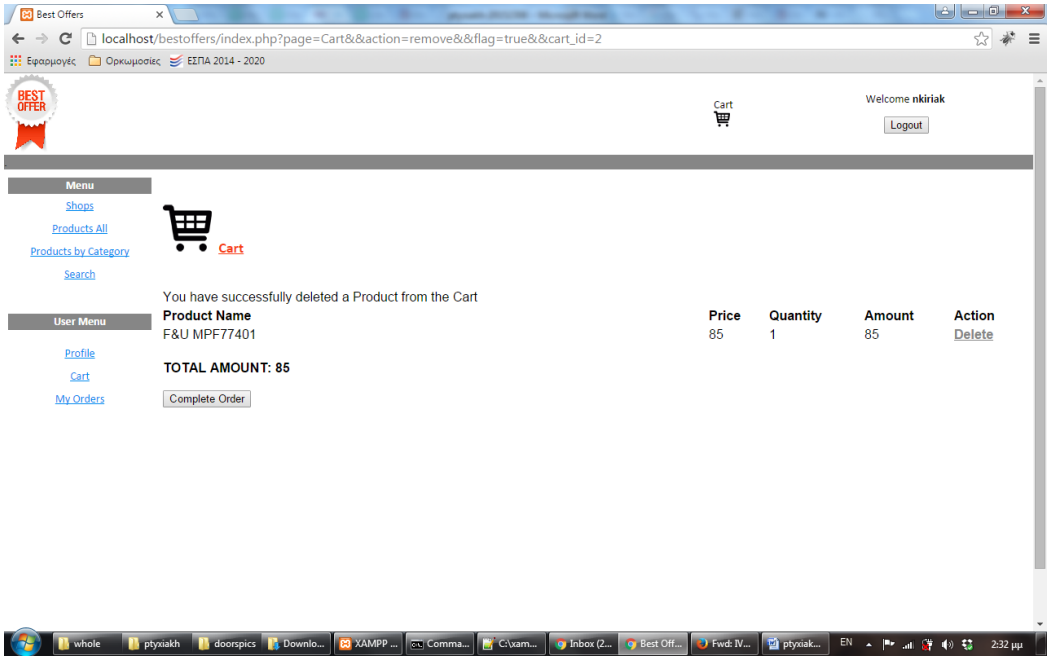
8.16.3 Διαγραφή από το Καλάθι

Screenshots:

Κάθε σύνδεσμος Delete, αντιστοιχεί στην διαδικασία Διαγραφής, περνώντας το id του ως παράμετρο.



Εικόνα 58: Διαγραφή από το καλάθι 1



Εικόνα 59: Διαγραφή από το Καλάθι 2

SQL query:

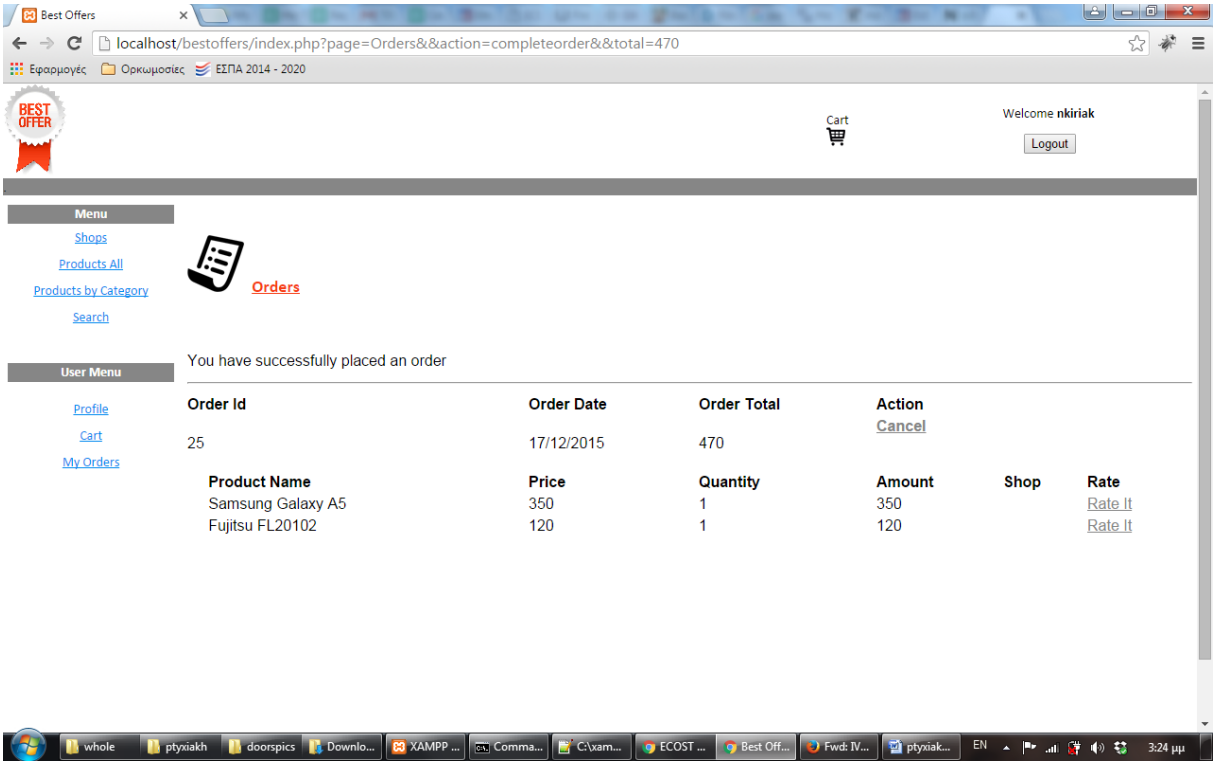
delete from cart where cart_id=$StrtsID

8.17 Παραγγελίες

8.17.1 Αρχική Οθόνη Παραγγελίας

Screenshots:

Στο καλάθι έχει επίσης τη δυνατότητα να πατήσει το κουμπί **Ολοκλήρωση Παραγγελίας (Complete Order).** Αν ολοκληρώσει την παραγγελία του, τότε εμφανίζεται μήνυμα ολοκλήρωσης , αναλυτικά στοιχεία των προιόντων και ο σύνδεσμος αξιολόγησης.



Εικόνα 60: Αρχική Οθόνη Παραγγελίας – Ολοκλήρωση Παραγγελίας

SQL Query:

```
if ($_GET['action'] == "completeorder"){

        $x=$_SESSION['userID'];
        $date = date('Y-m-d H:i:s');

        $total=$_GET['total'];

        $query1="insert into orders (o_date, o_total, o_userID, o_iscancelled)
                values (now(), $total, $x, 0)";
        $db1=mysql_query($query1);
```

```
$query_id="SELECT LAST_INSERT_ID()";
$db_id=mysql_query($query_id);
$temp_id=mysql_fetch_row($db_id);
$last_id=$temp_id[0];


$query2 = "select * from cart where cart_userID=$x";
$db2=mysql_query($query2);
while($temp2=mysql_fetch_row($db2)){

        $query3="insert into OrdersHasProducts (o_id, p_id, p_quantity, p_price )
                values ($last_id, $temp2[1], $temp2[2], $temp2[3])";
        $db3=mysql_query($query3);

        $query4 = "SELECT s_id FROM products where p_id=$temp2[1]";
        $db4=mysql_query($query4);
        $temp4=mysql_fetch_row($db4);

        $addvalue=$temp2[2]*$temp2[3];

        $query5="update shops set s_total=s_total+$addvalue where s_id=$temp4[0]";
        $db5=mysql_query($query5);

}

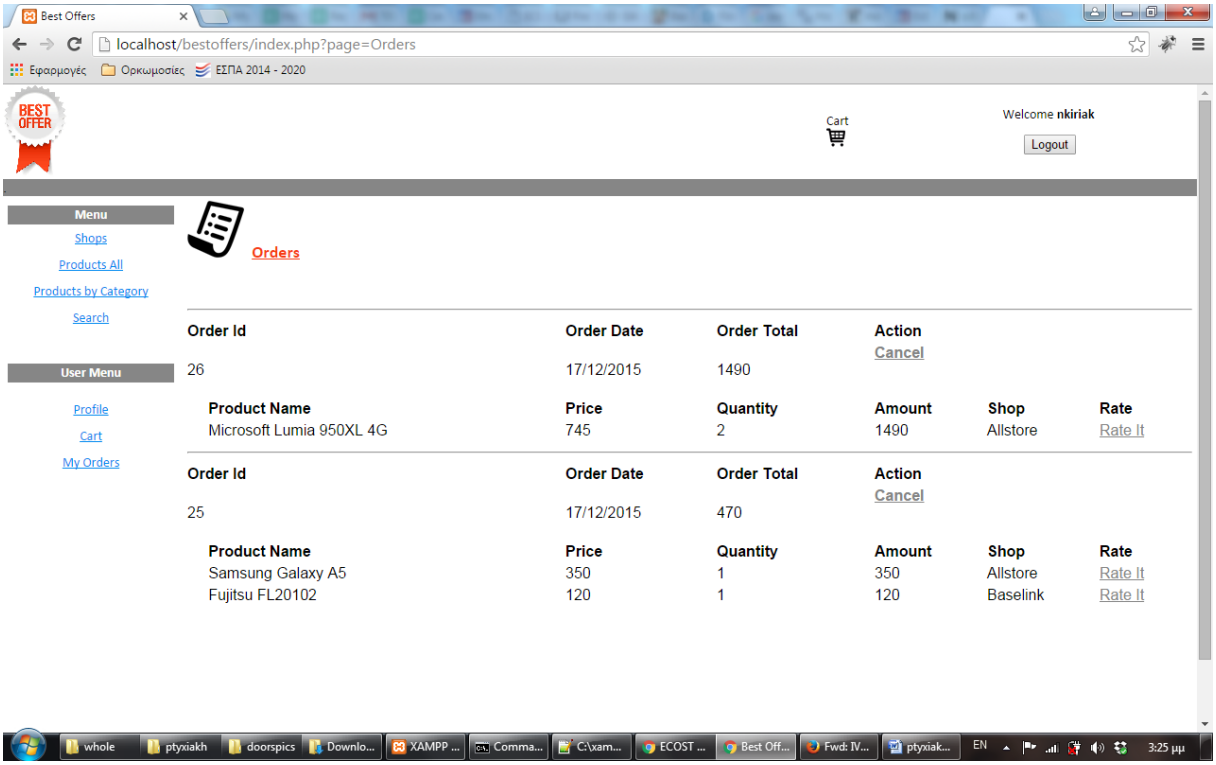
$query6 = "delete from cart where cart_userID=$x";
$db6=mysql_query($query6);

echo( "<br><tr><td colspan=\"6\"><br>You have successfully placed an order</td></tr>");
```

8.17.2 Οι παραγγελίες μου

Screenshots:

Στην οθόνη Οι παραγγελίες μου εμφανίζεται λίστα με όλες τις παραγγελίες που έχω πραγματοποιήσει με αναλυτικά στοιχεία των προϊόντων για την καθεμία, σύνδεσμο αξιολόγησης για το κάθε προιόν και σύνδεσμο ακύρωσης για την παραγγελία.



Εικόνα 61: Οι παραγγελίες μου

SQL query:

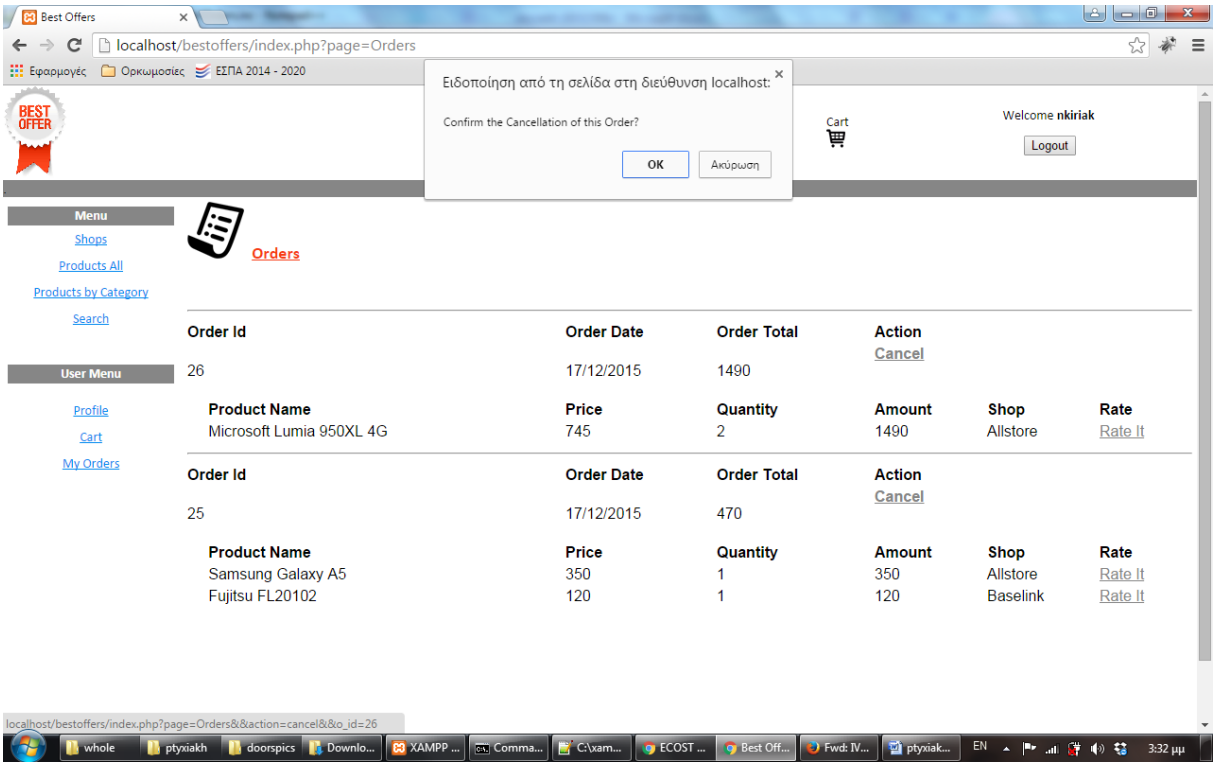
$x=$_SESSION['userID'];

$query = "SELECT * FROM orders where o_userID=$x and o_iscancelled=0 order by o_id desc";

8.17.3 Ακύρωση Παραγγελίας

Screenshots:

Κάθε σύνδεσμος Cancel, αντιστοιχεί στην διαδικασία Ακύρωσης της συγκεκριμένης Παραγγελίας, περνώντας το id της ως παράμετρο.



Εικόνα 62: Ακύρωση Παραγγελίας

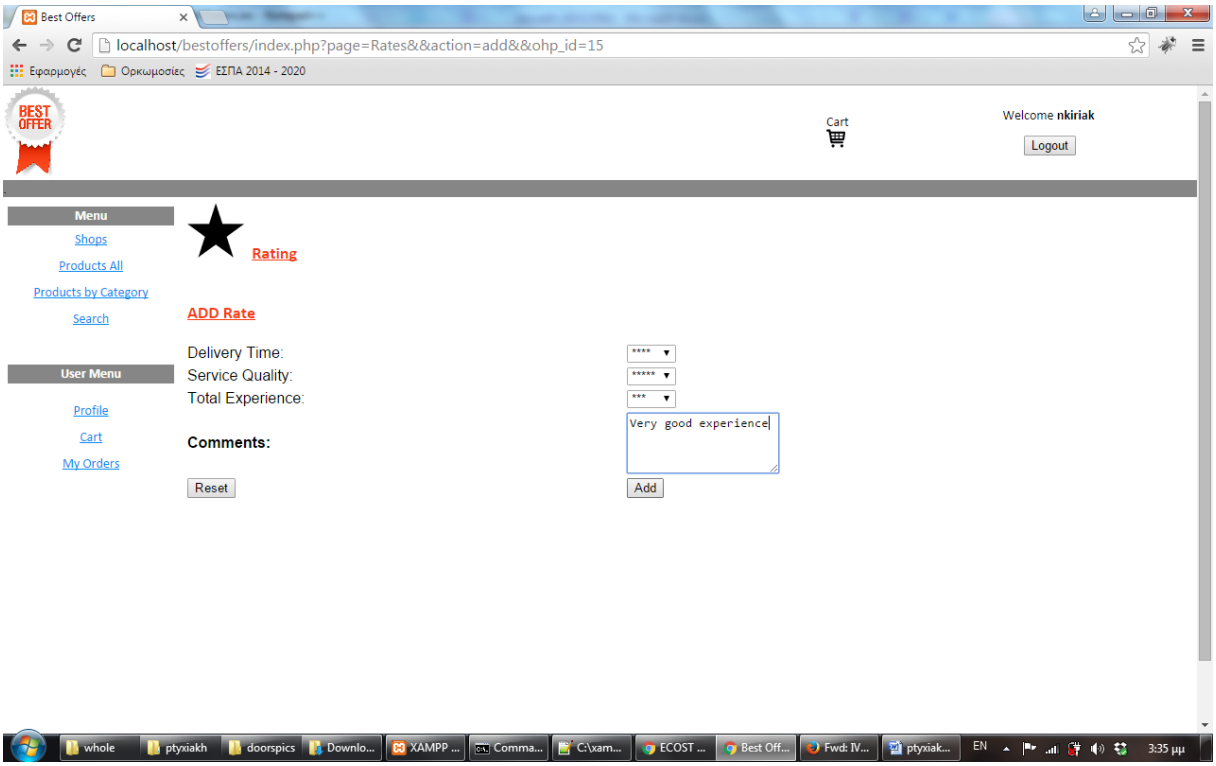
SQL query:

update orders set o_iscancelled=1 where o_id=$StrtsID

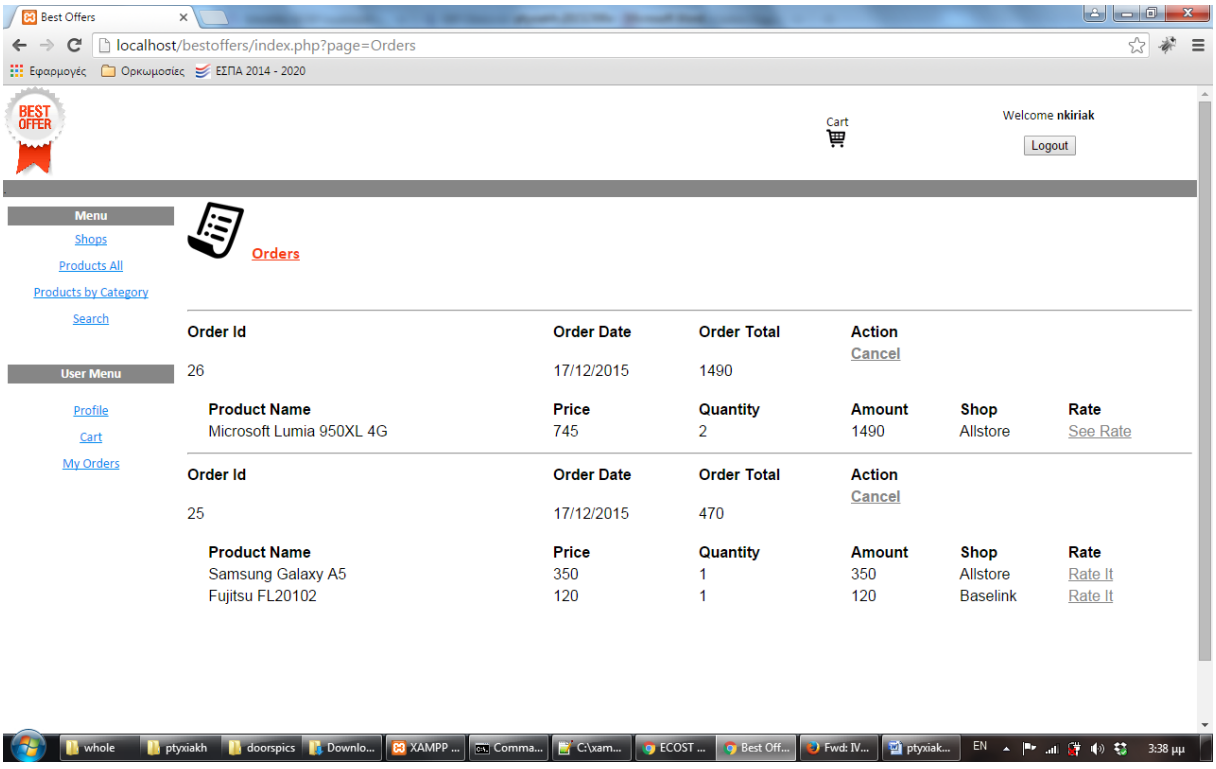
8.17.4 Αξιολόγηση Παραγγελίας

Screenshots:

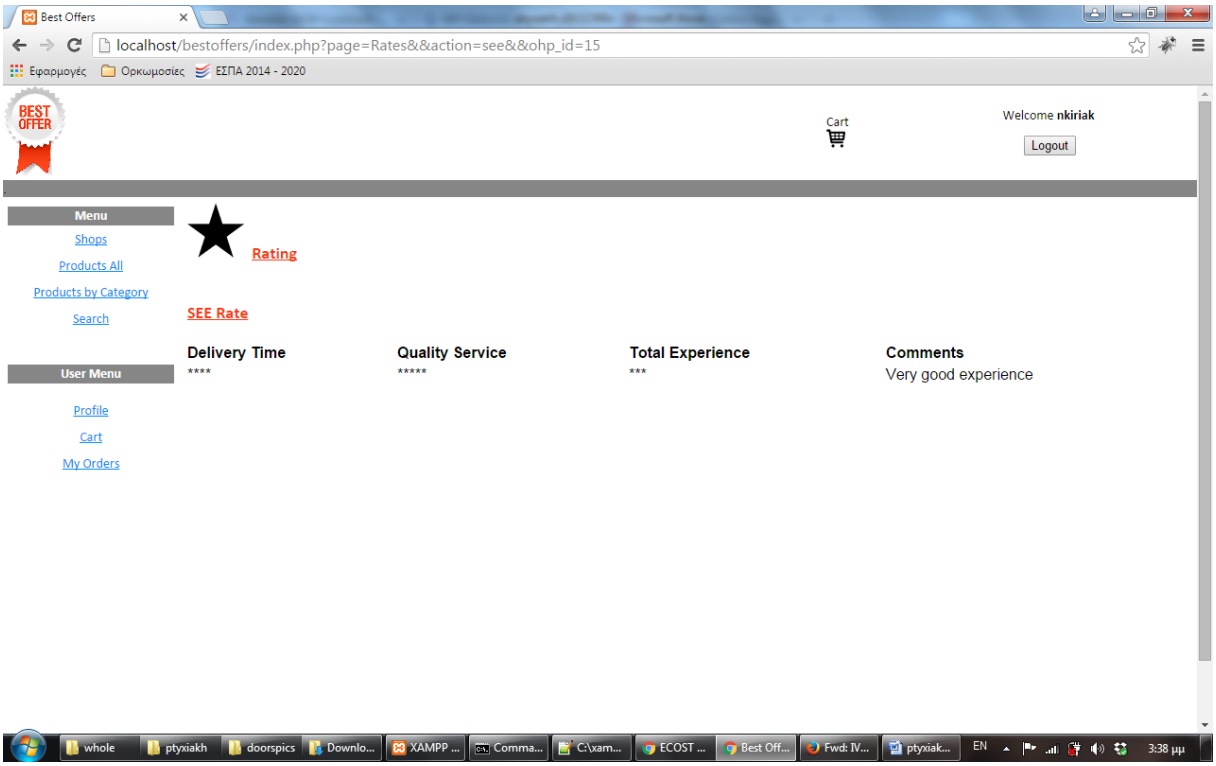
Κάθε σύνδεσμος **Αξιολόγησέ το (Rate it)**, προωθεί το χρήστη στη φόρμα αξιολόγησης της παραγγελίας για το εκάστοτε προιόν, ενώ κάθε σύνδεσμος **Δες την αξιολόγηση (See Rate),** προωθεί το χρήστη στην αξιολόγηση που έβαλε για αυτό το προιόν.



Εικόνα 63: Αξιολόγηση Παραγγελίας1



Εικόνα 64: Αξιολόγηση Παραγγελίας 2



Εικόνα 65: Αξιολόγηση Παραγγελίας 3

SQL Query:

```
if($_GET['flag']=="true" ){
        $ohp_id=$_GET['ohp_id'];
        $r_deltime = $_POST['r_deltime'];
        $r_service = $_POST['r_service'];
        $r_totalexp = $_POST['r_totalexp'];
        $r_comments = $_POST['r_comments'];

        $userID = $_SESSION['userID'];

        $query4 = "SELECT p_id FROM OrdersHasProducts where ohp_id=$ohp_id";
        $db4=mysql_query($query4);
        $temp4=mysql_fetch_row($db4);
```

```
$query5 = "SELECT s_id FROM products where p_id=$temp4[0]";
$db5=mysql_query($query5);
$temp5=mysql_fetch_row($db5);

$query1="insert into rates (r_deltime, r_service, r_totalexp, r_comments, r_userID, r_s_id )
        values ('$r_deltime', '$r_service', '$r_totalexp', '$r_comments', $userID, $temp5[0])";
$db1=mysql_query($query1);

$query_id="SELECT LAST_INSERT_ID()";
$db_id=mysql_query($query_id);
$temp_id=mysql_fetch_row($db_id);
$last_id=$temp_id[0];

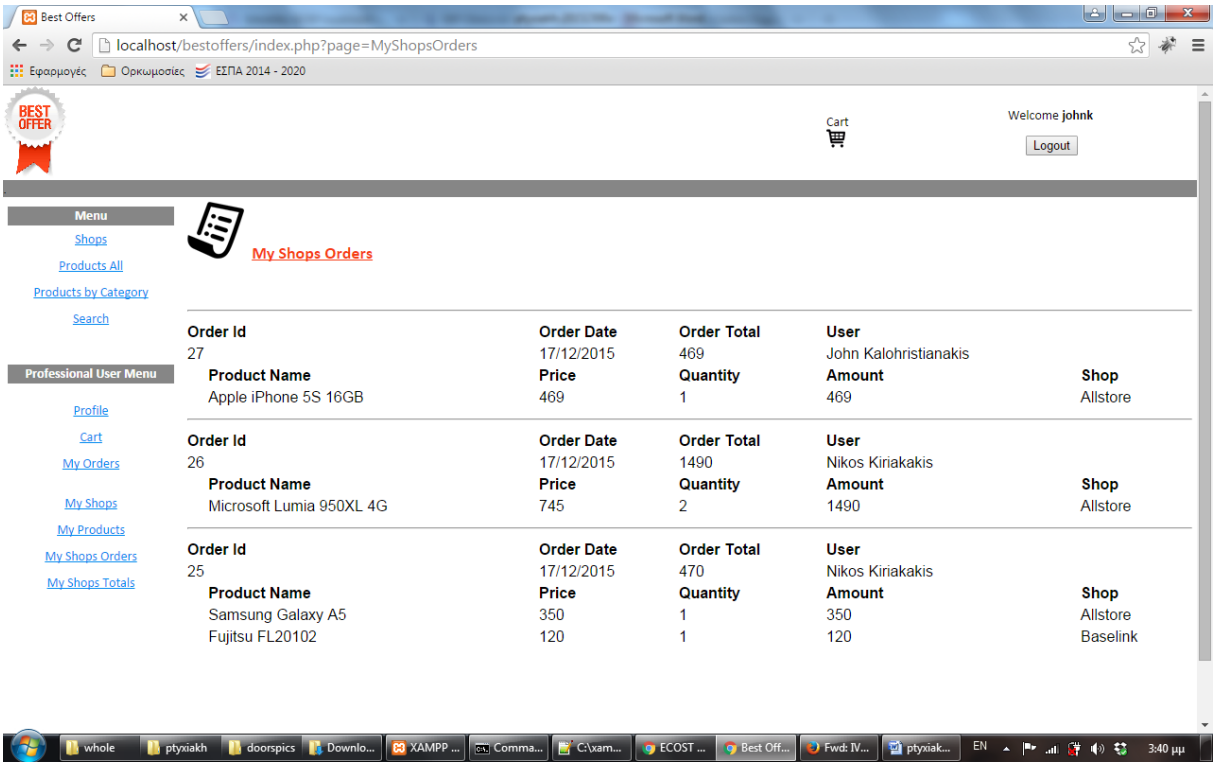
$query2="update OrdersHasProducts set r_id=$last_id where ohp_id=$ohp_id";
$db2=mysql_query($query2);

echo( "<tr><td colspan=\"4\"><br>You have successfully added a Rate of yours</td></tr>");
}
```

8.18 Λογαριασμός επαγγελματία χρήστη

8.18.1 Οι παραγγελίες των καταστημάτων μου

Screenshots:



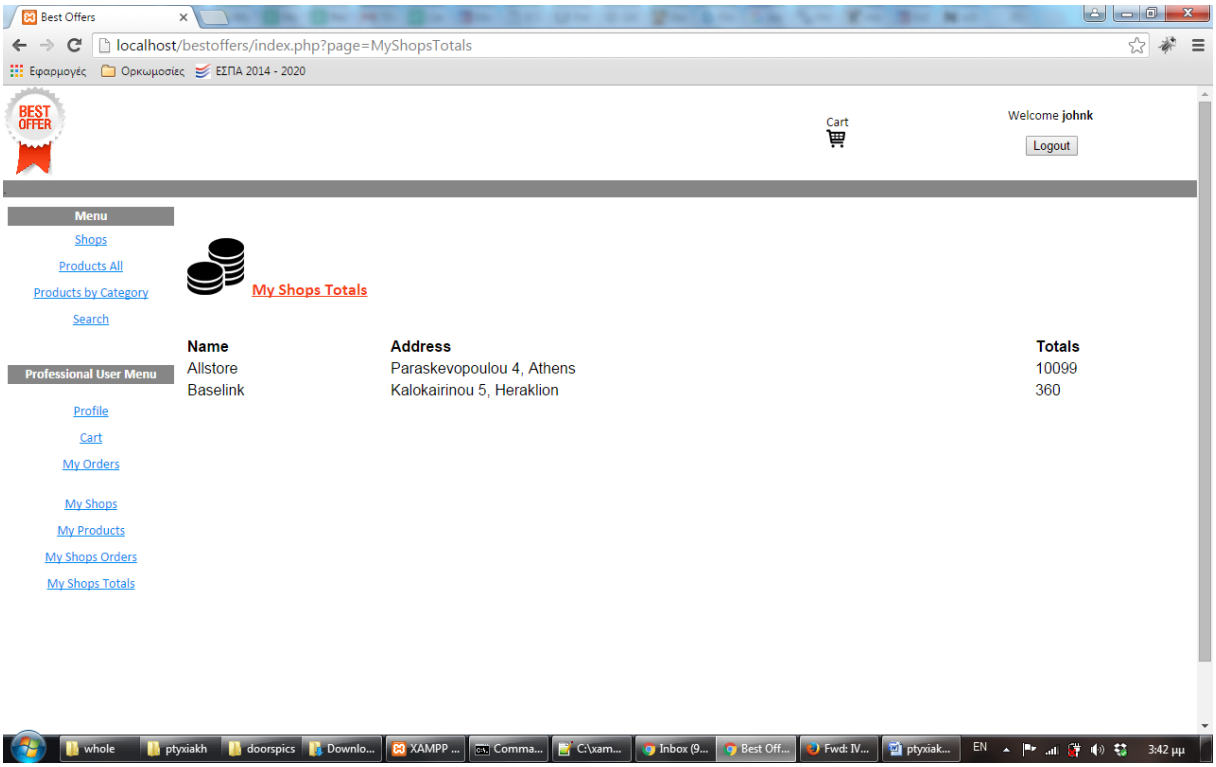
Εικόνα 66: Οι Παραγγελίες των καταστημάτων μου

SQL Query:

```
SELECT DISTINCT a.o_id FROM orders a, products b, shops c, OrdersHasProducts d
WHERE a.o_id=d.o_id
AND d.p_id IN
        (SELECT p_id FROM products where s_id IN
                (SELECT s_id FROM shops where userid=$x))
AND a.o_iscancelled=0
order by a.o_id desc
```

8.18.2 Τα συνολικά στατιστικά των καταστημάτων μου

Screenshots:



Εικόνα 67: Τα συνολικά στατιστικά των καταστημάτων μου

8.18.3 Προσθήκη καταστήματων

Ακολουθείται η διαδικασία προσθήκης καταστήματος όπως στον διαχειριστή, με τη διαφορά ότ το κατάστημα θα ανήκει στον χρήστη που το προσθέτει. (ενώ ο διαχειριστής το αναθέτει σε κάποιον χρήστη του συστήματος)

8.18.4 Προσθήκη Προιόντων για τα καταστήματα του

Ακολουθείται η διαδικασία προσθήκης Προιόντων όπως στον διαχειριστή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Wikipedia.** Code Examples on Ajax, CGI, Java, Javascript, Actionscript, Asp, Php, C++ and .Net. www.wikipedia.com.
2. *Some Trends in Web Application Development.* **Jazayeri, Mehdi.** 2007. Future of Software Engineering. σ. International conference on Software Engineering.
3. **Google.** Google Chrome: Μια νέα προσέγγιση στο πρόγραμμα περιήγησης. http://www.google.com/chrome/intl/el-GR/why.htm.
4. **Baldwin, Richard.** Similarities and Differences between Java and C++. 1997. http://www.dickbaldwin.com/java/Java008.htm.
5. **Wheeler, Scott και Marshall, Brian.** *LScript Reference manual.* s.l. : Newtek, 2002.
6. **Dawson, Bruce.** Game Scripting in Python. (Games Developer Conference 2002). http://www.gamasutra.com/features/20020821/dawson_01.htm.
7. **Sweeny, Tim.** UnrealScript language Reference (Epic Megagames).
    a. http://unreal.epicgames.com/UnrealScript.htm.
8. **W3 Schools.** On-line Tutorials and Examples on XML, Javascript, PHP, Asp. http://www.w3schools.com/.
9. *A comparison of VBScript, Javascript and Jscript.* **Kailash, Chandra, Suhani, Chandra Sapana και Suhana, Chandra Shyamal.** Journal of Computing Sciences in Col eges.
10. *Server-centric Web frameworks: An overview.* **Reahl, Iwan Vosloo και Kourie, Derrick G.** ACM Computing Surveys (CSUR).
11. **The PHP Group.** Php language Documentation. http://www.php.net/docs.php.
12. *An online bookstore using JSP technology.* **Hughes, Jose.** s.l. : ACM-SE 43: , Τόμ. Proceedings of the 43rd annual Southeast regional conference, Volume 2.
13. *Java servlets and Java server pages (JSP).* **Gordon, Aaron.** 2002, Journal of Computing Sciences in Col eges.
14. **Microsoft Inc.** ASP.NET reference. *Microsoft Developer Network.*
    a. http://msdn.microsoft.com/en-us/asp.net/default.aspx.
15. **Adobe Inc.** Adobe Flash Resources and Reference.
    a. http://www.adobe.com/support/documentation/en/flash/.
16. *ActionScript: a gentle introduction to programming.* **Crawford, Stewart και Boese, Elizabeth.**
    a. 2006, Journal of Computing Sciences in Col eges.
17. **Ecma.** Ecma 2007. ECMAScript Edition 4 specification wiki. *http://wiki.ecmascript.org.* 2007. http://www.ecma-international.org/publications/standards/Ecma-262.htm.
18. **Adobe Inc.** Adobe Flex Resources & Reference.
    a. http://www.adobe.com/support/documentation/en/flex/.