



Τ.Ε.Ι. ΠΕΛΟΠΟΝΝΗΣΟΥ
Τμήμα Μηχανικών Πληροφορικής
Σχολή Τεχνολογικών Εφαρμογών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Δυναμική Εφαρμογή Δημιουργίας και Ανάλυσης του
Προφίλ των Χρηστών της**

Θανόπουλος Ιάσων

A.M. 2009014

Επιβλέπων Καθηγητής: κ. Κλεάνθης Γκατζιώλης

Περίληψη

Σκοπός της εργασίας είναι η μελέτη και η εφαρμογή τεχνολογιών δημιουργίας και ανάλυσης προφίλ μιας δυναμικής εφαρμογής, που προέρχεται σύμφωνα με τις επιλογές του κάθε χρήστη πάνω στην εφαρμογή.

Στα πλαίσια της εργασίας σχεδιάστηκε και υλοποιήθηκε μια διαδικτυακή εφαρμογή η οποία αφήνει τους χρήστες να δημιουργήσουν το προφίλ τους και στην συνέχεια η εφαρμογή το αναλύει και το παραμετροποιεί λαμβάνοντας υπόψη το ιστορικό περιήγησης του κάθε χρήστη. Τέλος, επεξεργάζεται αυτά τα δεδομένα σύμφωνα με τις αγοραστικές επιλογές του χρήστη και τις εμφανίζει ως λογικές συστάδες.

Το διαδίκτυο κατακλύζεται από έναν τεράστιο όγκο πληροφοριών καθημερινά και κάθε χρήστης έχει πολύ διαφορετικά ενδιαφέροντα από κάποιον άλλον. Γι' αυτό είναι σημαντικό αυτές οι πληροφορίες να μπορούν να φιλτράρονται και να ταξινομούνται ανάλογα με τις προτιμήσεις τους. Έτσι τα συστήματα ανάλυσης προφίλ εκμεταλλεύονται τις ιδιαιτερότητες και προτιμήσεις του κάθε χρήστη της και τελικά αυτές μπορούν να μελετηθούν ή και να χρησιμοποιηθούν από άλλες εφαρμογές ή ανθρώπους.

Στην εργασία αυτή γίνεται ανάλυση των μεθόδων συλλογής δεδομένων (data gathering), της μοντελοποίησης των χρηστών της εφαρμογής (user modeling) καθώς και τους τρόπους με τους οποίους μπορούν να χρησιμοποιηθούν όλες αυτές οι πληροφορίες.

Abstract

The aim of the project is the study and application of profiling and analysis technologies for a dynamic application that comes according to the choices of each user on the application.

In this thesis there was designed and implemented a web application that lets users create their user profile and then the application analyzes and configures it considering the browsing history of each user. Finally, the application processes that data according to the user's shopping choices and displays them as logical clusters.

The Internet is flooded with a huge amount of information every day and every user has very different interests from another. It is therefore important that this information can be filtered and sorted according to their preferences. Thus the profiling systems exploit particularities and preferences of each user and finally they can be studied or used by other applications or humans.

This paper analyzes the methods for collecting data (data gathering), the modeling of the application users (user modeling) and the ways in which all this information can be used.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κύριο Κλεάνθη Γκατζιώλη καθηγητή του τμήματος Μηχανικών Πληροφορικής Τ.Ε., για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο.

Τέλος θα ήθελα να αφιερώσω αυτήν την εργασία στους γονείς μου, Κωνσταντίνο και Lise-Lotte, και στην αδερφή μου, Σοφία για όλη την συμπαράσταση που μου έδειξαν.

Πίνακας Περιεχομένων

1	Εισαγωγή	
1.1	Αντικείμενο της Πτυχιακής	10
1.2	Στόχοι της Πτυχιακής	11
1.3	Δομή της Πτυχιακής	11
2	Βασικό Υπόβαθρο	
2.1	Προφίλ Χρήστη	14
2.2	Μοντελοποίηση Χρήστη	15
2.3	Τύποι Δεδομένων στα Μοντέλα Χρηστών	16
2.3.1	Δημογραφικά Στοιχεία	16
2.3.2	Γνώση και Υπόβαθρο	16
2.3.3	Δεδομένα Ενδιαφερόντων και Προτιμήσεων	17
2.3.4	Μεμονωμένα Γνωρίσματα	17
2.4	Χρήσεις Δεδομένων των Μοντέλων Χρηστών	18
2.4.1	Προσαρμοστικά Υπερμέσα	18
2.4.2	Έμπειρα Συστήματα	19
2.4.3	Συστήματα Συστάσεων	19
2.4.4	Προσομοίωση Χρηστών	20
2.5	Εξόρυξη Γνώσης	21
3	Δημιουργία και Ανάλυση των Προφίλ των Χρηστών	
3.1	Δημιουργία Προφίλ	26
3.2	Δομή Προφίλ	27
3.3	Παρακολούθηση Προφίλ	28
3.4	Συλλογή Δεδομένων	29
3.5	Ανάλυση Δεδομένων	31
3.6	Οπτικοποίηση Δεδομένων	32
4	Παρόμοια Συστήματα	
4.1	Το Σύστημα WEST	35
4.2	Το Σύστημα Gumsaws	36
5	Χαρακτηριστικά Συστήματος	
5.1	Δυναμικές Ιστοσελίδες	38
5.2	Τεχνολογίες Συστήματος	39
5.2.1	HTML	39
5.2.2	CSS	41
5.2.3	Javascript	42
5.2.4	Το μοντέλο WAMP	43

5.2.4.1	Apache FTTP Server	44
5.2.4.2	MySQL	44
5.2.4.3	PHP	45
5.3	Προφίλ Χρηστών και Sessions	45
5.4	Η Ιστοσελίδα	47
5.5	Η Βάση Δεδομένων	48
5.6	Τεχνική Παρακολούθησης Χρηστών	50
5.7	Τεχνική Ανάλυσης και Εμφάνισης Δεδομένων	51
6	Ο κώδικας του Συστήματος	
6.1	Βασική Δομή	55
6.2	Η CSS του Συστήματος	58
6.3	Δημιουργία Χρήστη	62
6.4	Είσοδος και Έξοδος από το Σύστημα	64
6.5	Τα Προϊόντα	66
6.6	Το Καλάθι Αγορών	71
6.7	Ο Διαχειριστής	76
6.7.1	Προσθήκη Αντικειμένων	77
6.7.2	Στατιστικά Χρηστών	79
6.7.3	Στατιστικά Ιστοσελίδας	84
7	Συμπεράσματα και Μελλοντικές Επεκτάσεις	
7.1	Συμπεράσματα	87
7.2	Μελλοντικές Επεκτάσεις	87
	Βιβλιογραφία	89

Κεφάλαιο 1

Εισαγωγή

Εισαγωγή

Ο Παγκόσμιος Ιστός εξελίσσεται ραγδαία και το διαδίκτυο είναι πια προσβάσιμο από εκατομμύρια χρήστες, δίνοντας τους την δυνατότητα να έχουν πρόσβαση σε περισσότερες μορφές ειδήσεων, περισσότερους τρόπους αγοραπωλησιών ακόμα και περισσότερους τρόπους διασκέδασης.

Η πλοήγηση στο διαδίκτυο όμως είναι κατά κύριο λόγο απρόσωπη και ανώνυμη και λόγω του μεγάλου πληθυσμού που το χρησιμοποιεί είναι δύσκολο αυτοί οι άνθρωποι να ξεχωριστούν και να κατηγοριοποιηθούν αναλόγως με τις προτιμήσεις τους.

Μια λύση σε αυτό το πρόβλημα είναι η δημιουργία και ανάλυση των προφίλ των χρηστών μιας διαδικτυακής εφαρμογής η οποία ακολουθεί τις κινήσεις τους και τις καταγράφει αφήνοντας πίσω ένα προσωποποιημένο πια προφίλ για κάθε έναν χρήστη της.

Η προσωποποιημένη αυτή προσέγγιση έχει γίνει πια αναγκαία για πολλές ιστοσελίδες (ειδήσεων, καταστημάτων κ.α.) καθώς παρέχει όχι μόνο βασικές πληροφορίες (όνομα, email κ.α.) αλλά και δευτερεύουσες, εξίσου σημαντικές, όπως ενδιαφέροντα, επιθυμίες, και συμπεριφορές κάνοντας την επικοινωνία με τον χρήστη (αναγνώστη, πελάτη κ.α.) ακόμη πιο εύκολη. Με αυτόν τον τρόπο η πλατφόρμα θα μπορεί να για παράδειγμα να στείλει συγκεκριμένες πληροφορίες μόνο στους χρήστες που την ενδιαφέρουν.

Η μέθοδος με την οποία μαζεύονται όλες αυτές οι πληροφορίες για τους χρήστες και ξεχωρίζονται ανάλογα με τις προτιμήσεις τους λέγεται μοντελοποίηση χρηστών (user modeling) και υπάρχουν πολλές εφαρμογές που το κάνουν αυτό καθώς επίσης υπάρχουν και πολλά μοντέλα για να επιτευχθεί αυτό. Από τα πιο απλά που είναι τα στατικά με μικρή δυνατότητα ευελιξίας μέχρι και δυναμικά μοντέλα ή ακόμη και υβριδικά που συνδυάζουν τα παραπάνω.

Η δυνατότητα αυτή να μπορούμε να προσωποποιούμε τους χρήστες είναι πολύ χρήσιμη καθώς οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν από εφαρμογές και ιστοσελίδες ευρείας χρήσης. Παραδείγματα εφαρμογών που μπορεί να χρησιμοποιούν τέτοιες εφαρμογές είναι ιστοσελίδες που προβάλλουν περιεχόμενο αναλόγως τις προτιμήσεις των χρηστών, συστήματα συστάσεων ή και προσομοιωτές χρηστών.

1.1 Αντικείμενο της Πτυχιακής

Το αντικείμενο της παρούσας εργασίας είναι να αναδειχθεί η χρησιμότητα και τα πλεονεκτήματα που προσφέρουν οι τεχνολογίες ανάλυσης προφίλ των χρηστών καθώς και ανάλυση των μεθόδων που χρησιμοποιούνται και προβλημάτων των οποίων μπορεί να δημιουργούνται. Επιπλέον θα γίνει ανάλυση της μελλοντικής χρήσης της εφαρμογής καθώς και αξιοποίησή της από τρίτες εφαρμογές.

Η εφαρμογή που θα δημιουργηθεί θα πρέπει να είναι σε θέση να δέχεται πληροφορίες και να τις ερμηνεύει ως λογικές συστάδες οι οποίες θα είναι κατανοητές από χρήστες ή προγράμματα που θα τις παρακολουθούν.

Η εφαρμογή αναπαριστά ένα ηλεκτρονικό κατάστημα διαφόρων αθλητικών ειδών όπου οι χρήστες συνδέονται στο σύστημα και από εκεί θα κάνουν τις αγορές τους. Καθ' όλη τη διάρκεια το σύστημα παρακολουθεί τις κινήσεις των χρηστών και είναι σε θέση να δει αν, για παράδειγμα, του χρήστη του αρέσει το τένις και όχι το ποδόσφαιρο.

1.2 Στόχος της Πτυχιακής

Η εργασία αυτή δίνει βάση στο πεδίο της δημιουργίας και της ανάλυσης των προφίλ των χρηστών μιας εφαρμογής και κατά πόσο αυτή θα βελτιώσει ένα σύστημα ηλεκτρονικού καταστήματος που θα δείχνει τα ενδιαφέροντα των πελατών της.

Αρχικά ο χρήστης δημιουργεί το προφίλ του στην πλατφόρμα και από εκεί και πέρα η εφαρμογή χρησιμοποιεί τεχνολογίες ικανές να παρακολουθήσουν την κίνηση των χρηστών μέσα στην εφαρμογή.

Στόχος μας είναι ο διαχειριστής την πλατφόρμας να είναι σε θέση να μπορεί να αναζητήσει καθέναν χρήστη ξεχωριστά και η εφαρμογή να “υπολογίζει” τα ενδιαφέροντα τους. Το σύστημα δηλαδή αποτελείται από έναν μηχανισμό που συνδυάζει όλα τα παραπάνω στάδια: διεπαφή χρήστη με το σύστημα, ιστορικό του χρήστη, κανόνες εμφάνισης πληροφοριών.

1.3 Δομή της Πτυχιακής

Στο υπόλοιπο μέρος της εργασίας παρουσιάζεται το θεωρητικό υπόβαθρο πάνω στο οποίο βασίστηκε η ανάπτυξη της εφαρμογής.

Πιο συγκεκριμένα:

Στο κεφάλαιο 2 γίνεται αναφορά στο βασικό υπόβαθρο και χρήσεις της ανάλυσης των προφίλ των χρηστών.

Στο κεφάλαιο 3 γίνεται ανάλυση των συστημάτων δημιουργίας και ανάλυσης των προφίλ των χρηστών.

Στο κεφάλαιο 4 παρουσιάζονται παρόμοια συστήματα.

Στο κεφάλαιο 5 γίνεται περιγραφή των χαρακτηριστικών του συστήματος και παρουσίασή του.

Στο κεφάλαιο 6 γίνεται ανάλυση του κώδικα του συστήματος.

Στο κεφάλαιο 7 περιγράφονται συμπεράσματα και μελλοντικές χρήσεις και επεκτάσεις του συστήματος.

Κεφάλαιο 2

Βασικό Υπόβαθρο

2.1 Προφίλ Χρήστη

Το προφίλ ενός χρήστη είναι μια οπτική απεικόνιση των δεδομένων του προσωπικού χαρακτήρα που συνδέονται με ένα συγκεκριμένο χρήστη ή ένα προσαρμοσμένο περιβάλλον εργασίας. [1] Ένα προφίλ είναι δηλαδή η ψηφιακή αναπαράσταση της ταυτότητας ενός ατόμου. Μπορεί όμως να θεωρηθεί και ως η αναπαράσταση ενός μοντέλου χρήστη.

Ένα προφίλ μπορεί να χρησιμοποιηθεί για την αποθήκευση της περιγραφής και των χαρακτηριστικών του προσώπου που αντιπροσωπεύει. Η πληροφορία αυτή μπορεί να αξιοποιηθεί από διάφορα συστήματα που λαμβάνουν υπόψη τα χαρακτηριστικά και τις προτιμήσεις των ανθρώπων. Για αυτόν τον λόγο τα προφίλ είναι απαραίτητα για ένα σύγχρονο σύστημα καθώς μέσω αυτού γίνεται προσωποποίηση των πληροφοριών που βρίσκονται στο προφίλ και έτσι μας δίνεται η δυνατότητα να τα ξεχωρίσουμε και να τα ομαδοποιήσουμε.

Η πληροφορία που περιέχει ένα προφίλ μπορεί να είναι είτε δυναμική είτε στατική. Στην πρώτη περίπτωση το προφίλ ονομάζεται δυναμικό και αυτό σημαίνει ότι η πληροφορία μπορεί να αλλάξει με την πάροδο του χρόνου. Οι αλλαγές αυτές συνήθως γίνονται σύμφωνα με τις ενέργειες που πραγματοποιεί ο χρήστης στο σύστημα και συνήθως ο χρήστης δεν μπορεί να χρησιμοποιήσει ή να κάνει αλλαγές στις πληροφορίες αυτές. Αντίθετα στην δεύτερη περίπτωση που το προφίλ ονομάζεται στατικό, οι πληροφορίες στο προφίλ παραμένουν σταθερές για ένα μεγάλο χρονικό διάστημα και σπανίως αλλάζουν. Ένα τέτοιο προφίλ θα περιέχει κυρίως δημογραφικά στοιχεία για τον χρήστη, όπως ονοματεπώνυμο, ηλικία, ύψος κ.α. Σε πολλά συστήματα μπορεί να παρατηρηθεί ένας συνδυασμός των πλεονεκτημάτων των στατικών και δυναμικών καθιστώντας έτσι το προφίλ υβριδικό. Τα προφίλ μπορεί να βρεθούν σε λειτουργικά συστήματα, προγράμματα ηλεκτρονικών υπολογιστών, συστήματα συστάσεων, ηλεκτρονικά παιχνίδια κ.α.

2.2. Μοντελοποίηση Χρήστη

Μοντελοποίηση χρήστη είναι μια υποδιαίρεση της αλληλεπίδρασης ανθρώπου-υπολογιστή και περιγράφει τη διαδικασία δημιουργίας και τροποποίησης ενός μοντέλου χρήστη. [2]

Ο κύριος στόχος της μοντελοποίησης των χρηστών είναι η προσαρμογή των συστημάτων στις ιδιαίτερες ανάγκες του χρήστη. Το σύστημα πρέπει να φαίνεται ότι είναι φτιαγμένο για κάθε χρήστη ξεχωριστά, ενώ είναι φτιαγμένο για εκατομμύρια χρήστες. Θα πρέπει, δηλαδή να λέει "το σωστό πράγμα, την σωστή στιγμή, με τον σωστό τρόπο".[3]

Η μοντελοποίηση χρηστών αποτελείται από δύο κυρίως κατηγορίες. Το πρώτο είναι το μοντέλο του χρήστη που αποτελεί το σύνολο των πληροφοριών που αποτελούν το προφίλ του χρήστη και το δεύτερο είναι η συλλογή δεδομένων. Το σύνολο των πληροφοριών που αποτελούν το προφίλ είναι όλα τα δεδομένα που κάνουν το προφίλ ξεχωριστό από τα υπόλοιπα και περιγράφονται αναλυτικά στο κεφάλαιο 2.3. Η συλλογή δεδομένων είναι και αυτή ένα ξεχωριστό κεφάλαιο από μόνη της, όπως περιγράφεται στο κεφάλαιο 3.6, καθώς μέσω αυτής μπορούμε να επεκτείνουμε τις πληροφορίες που έχουμε για έναν χρήστη είτε ζητώντας από τον χρήστη να μας τις δώσει, είτε παρακολουθώντας τις ενέργειες του στο σύστημα. Το δεύτερο είναι εξαιρετικά σημαντικό για ένα σύστημα που μπορεί να προσαρμόζεται στις ανάγκες του χρήστη.

Ένα πολύ απλό παράδειγμα μοντελοποίησης χρηστών μπορεί να αποτελέσουν οι ιστοσελίδες ηλεκτρονικού εμπορίου που χρησιμοποιούν όλες τις πληροφορίες σχετικά με την πλοήγηση και τις αγορές ενός χρήστη και τις συνδυάζουν με πληροφορίες από άλλους χρήστες με σκοπό την καλύτερη κατανόηση των αγοραστικών τους προτιμήσεων. Έτσι το σύστημα μπορεί εύκολα να προτείνει στους χρήστες πιθανά προϊόντα που μπορεί να τους ενδιαφέρουν.

2.3 Τύποι Δεδομένων στα Μοντέλα Χρηστών

Τα δεδομένα των χρηστών περιλαμβάνουν δεδομένα σχετικά με την αλληλεπίδραση του στο σύστημα.[4] Έτσι κάθε χρήστης είναι φτιαγμένος σύμφωνα με τα δεδομένα αυτά και τον κάνει να ξεχωρίζει από τους υπόλοιπους. Παρακάτω παρουσιάζονται τύποι δεδομένων οι οποίοι μπορούν να ενσωματωθούν σε μοντέλα χρηστών.

2.3.1 Δημογραφικά Στοιχεία

Τα δημογραφικά δεδομένα έχουν στοιχεία σχετικά με το όνομα, το επώνυμο, την ηλικία, το ύψος, το βάρος, το φύλο, την υπηκοότητα, τον τόπο διαμονής κ.α. Τα στοιχεία αυτά μπορούν να επεκταθούν και να τροποποιηθούν σε τεράστιο βαθμό αναλόγως με τις απαιτήσεις της εφαρμογής. Συνήθως, αποτελούν το στατικό κομμάτι των προφίλ καθώς οι πληροφορίες αυτές αλλάζουν πολύ σπάνια έως ποτέ. Εξετάζοντας τα στοιχεία αυτά μπορούμε να ομαδοποιήσουμε τους χρήστες του συστήματος αναλόγως με το φύλλο τους και να εξετάσουμε τις ενέργειές τους ξεχωριστά. Κάτι τέτοιο πάλι θα μπορούσε να είναι χρήσιμο σε ένα σύστημα ηλεκτρονικού καταστήματος καθώς θα μπορούσαμε να εξετάσουμε τις αγοραστικές προτιμήσεις των δύο φύλλων ξεχωριστά.

2.3.2 Γνώση και Υπόβαθρο

Τα δεδομένα γνώσης ίσως είναι από τα σημαντικά στα μοντέλα χρηστών. Τα δεδομένα αυτά συνήθως δεν υπόκεινται σε συχνές αλλαγές και γίνονται βραχυπρόθεσμα αναγκάζοντας έτσι τα συστήματα να είναι δυναμικά. Αυτό σημαίνει πως το σύστημα θα πρέπει να

καταλαβαίνει τις αλλαγές στη γνώση που αποκτά ο χρήστης παρατηρώντας την κίνηση και τις επιλογές του χρήστη στο σύστημα και να αναπροσαρμόζει τα δεδομένα έτσι ώστε να είναι πιο χρήσιμα στον χρήστη. Τέτοια δεδομένα παρατηρούνται συνήθως σε προσαρμοστικά υπερμέσα, όπως παρουσιάζονται παρακάτω στο κεφάλαιο 2.4.1.

2.3.3 Δεδομένα Ενδιαφερόντων και Προτιμήσεων

Τα είδη δεδομένων αυτών αποτελούν τις σημαντικότερες πληροφορίες σε συστήματα που φιλτράρουν πληροφορίες, όπως είναι τα συστήματα συστάσεων, όπως παρουσιάζονται στο κεφάλαιο 2.4.4. Οι πληροφορίες όμως αυτές διαφέρουν συνήθως από τις δημογραφικές πληροφορίες καθώς δεν χρειάζεται να ερωτηθεί ο χρήστης για αυτές. Αντιθέτως, παρατηρώντας τα επαναλαμβανόμενα μοτίβα στις ενέργειες των χρηστών ένα ιδανικό σύστημα θα μπορούσε να συμπεράνει τα ενδιαφέροντα του χρήστη από μόνη της.

2.3.4 Μεμονωμένα Γνωρίσματα

Τα μεμονωμένα γνωρίσματα του χρήστη είναι το σύνολο των χαρακτηριστικών του (εξωστρεφής, αντιδραστικός κ.τ.λ.) που δεν υπόκεινται καμία αλλαγή ή αλλάζουν σε τεράστιο χρονικό διάστημα. Γι' αυτό πολλά τέτοια συστήματα με αυτού του είδους πληροφορίες μπορεί να είναι στατικά. Παραδείγματα τέτοιων συστημάτων είναι για ειδικά σχεδιασμένων ψυχολογικών τεστ. Οι πληροφορίες αυτές όπως και πριν διαφέρουν από τις δημογραφικές καθώς και εδώ είναι ιδιαίτερα σημαντικό να παρατηρούνται τα επαναλαμβανόμενα μοτίβα στις ενέργειες των χρηστών.

2.4 Χρήσεις Δεδομένων των Μοντέλων Χρηστών

Στα προηγούμενα κεφάλαια αναλύσαμε τα προφίλ και τις πληροφορίες που τις κατακλύζουν. Ένα σύγχρονο προφίλ θα πρέπει να έχει πληροφορίες τις οποίες έχει μαζέψει είτε δυναμικά, είτε στατικά και όλες μαζί σας σύνολο πρέπει να αποτελούν μια προσωποποίηση του χρήστη. Μόλις ένα σύστημα έχει συγκεντρώσει πληροφορίες για τους χρήστες μπορεί να αρχίσει να παρουσιάζει τα δεδομένα ή ακόμη και να τα χρησιμοποιήσει με το μέρος της. Η ανάλυση των προφίλ μπορεί να χρησιμοποιηθεί, έχοντας πολλά και σημαντικά οφέλη, σε εφαρμογές μερικές εκ τις οποίες παρουσιάζονται παρακάτω.

2.4.1 Προσαρμοστικά Υπερμέσα

Προσαρμοστικά υπερμέσα είναι η οικοδόμηση ενός μοντέλου με όλες τις προτιμήσεις, στόχους και γνώση για έναν μοναδικό χρήστη και χρήση του μοντέλου αυτού προκειμένου να προσαρμοστεί το σύστημα στις ανάγκες αυτού. [5] Σε αντίθεση, δηλαδή, με παραδοσιακά συστήματα όπου σε όλους τους χρήστες προσφέρονται τα ίδια υπερμέσα και σύνδεσμοι.

Για να είναι προσαρμοζόμενο ένα σύστημα θα πρέπει να πλήρη τρία κριτήρια. Πρώτον, θα πρέπει να είναι ένα σύστημα υπερκειμένου ή υπερμέσων. Δεύτερον, θα πρέπει να έχει ένα μοντέλο χρήστη και τέλος θα πρέπει να είναι σε θέση να το προσαρμόσει χρησιμοποιώντας το μοντέλο αυτό. [6]

Από τα πιο σημαντικά προσαρμοστικά υπερμέσα είναι αυτά στον τομέα της εκπαίδευσης όπου σε αντίθεση με την παραδοσιακή ηλεκτρονική εκμάθηση, όπου δίνονται τα ίδια υπερμέσα σε όλους τους μαθητές, αυτά προσαρμόζονται στις ανάγκες, στόχους και ενδιαφέροντα του μαθητή και του παρέχουν υπερμέσα τα οποία χρειάζεται αυτός.

2.4.2 Έμπειρα Συστήματα

Τα έμπειρα συστήματα είναι συστήματα ηλεκτρονικών υπολογιστών τα οποία μπορούν να μιμηθούν την ικανότητα λήψης αποφάσεων του ανθρώπου προκειμένου να βοηθήσει να λυθεί ένα πρόβλημα σε έναν συγκεκριμένο τομέα. Τα συστήματα αυτά λειτουργούν κάνοντας ερωτήσεις βήμα προς βήμα για να εντοπίσουν το πρόβλημα και να βρουν μια λύση σε αυτό. Τα μοντέλα χρήστη μπορούν να χρησιμοποιηθούν για να προσαρμοστούν στην τρέχοντος γνώση του χρήστη και να τον διαφοροποιήσουν ανάμεσα σε έμπειρους και αρχάριους. Το σύστημα είναι ικανό να υποθέσει ότι έμπειροι χρήστες είναι σε καλύτερη θέση να καταλάβουν τα πιο πολύπλοκα ερωτήματα από κάποιον που είναι νέος στον τομέα. Έτσι προσαρμόζει το λεξιλόγιο του και τα ερωτήματα που χρησιμοποιεί για να βρεθεί μια λύση.

2.4.3. Συστήματα Συστάσεων

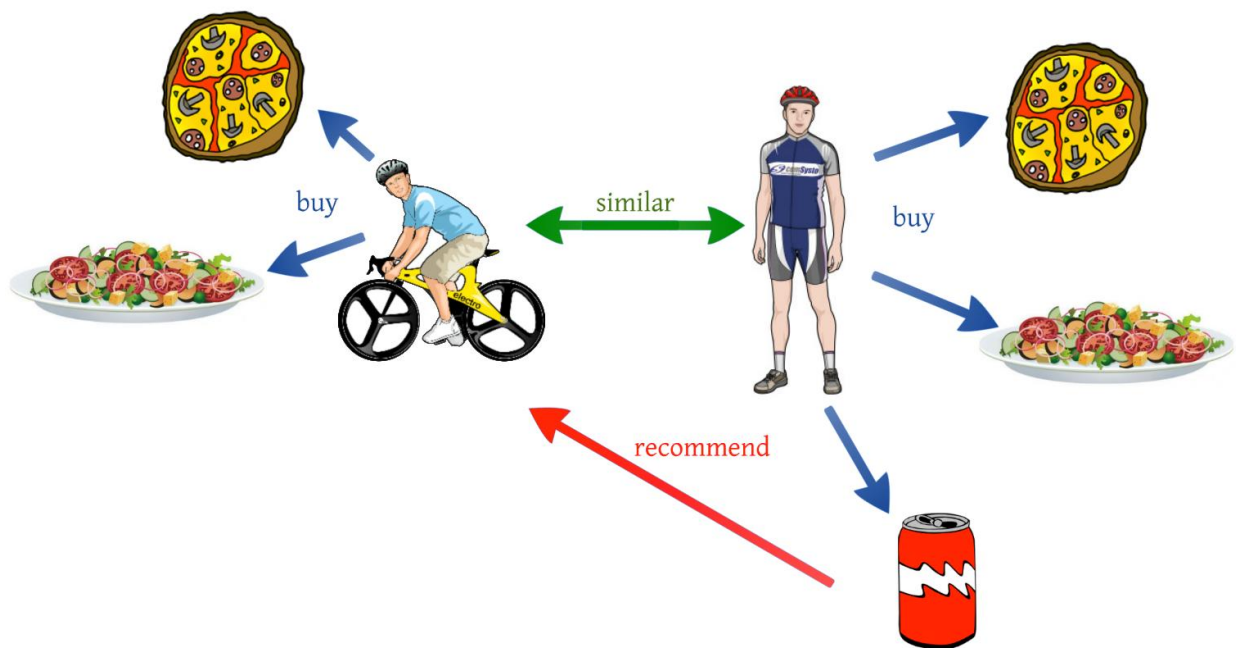
Τα συστήματα συστάσεων είναι εργαλεία εφαρμογών και τεχνικές οι οποίες παρέχουν συστάσεις για αντικείμενα που μπορεί να θέλει να χρησιμοποιήσει ένας χρήστης. Οι συστάσεις αυτές μπορεί να είναι αποφάσεις που θέλει να πάρει ο χρήστης, όπως τι να αγοράσει, τι μουσική να ακούσει ή τι νέα να διαβάσει. [7]

Η βασική ιδέα είναι να παρουσιάσει μια επιλογή στοιχείων που ταιριάζει καλύτερα στις ανάγκες του χρήστη ύστερα από ανάλυση που έγινε στο προφίλ του χρήστη κατά τη διάρκεια της δημιουργίας του προφίλ ή κατά την πλοήγηση του στην εφαρμογή.

Τα συστήματα συστάσεων έχουν γίνει εξαιρετικά κοινά τα τελευταία χρόνια και χρησιμοποιούνται ευρέως από τεράστια ποικιλία εφαρμογών. Οι πιο δημοφιλείς εφαρμογές είναι πιθανώς ταινίες, μουσική, ειδήσεις, βιβλία, ερευνητικά άρθρα, ερωτήματα αναζήτησης των μηχανών αναζήτησης, προϊόντα κτλ. Ένα πολύ κλασικό παράδειγμα συστήματος

συστάσεων είναι το σύστημα της ιστοσελίδας *www.stumbleupon.com* όπου χρησιμοποιεί τις αξιολογήσεις ιστοσελίδων που συγκεντρώθηκαν με ένα συνεργατικό σύστημα αξιολόγησης μπορεί να ταιριάξει τους χρήστες με ενδιαφέρουσες ιστοσελίδες βάση των προτιμήσεών τους.

Στην εικόνα 2.4.3 δείχνει δύο άτομα με ίδιες προτιμήσεις και πώς ένα σύστημα συστάσεων είναι ικανό να προτείνει στον πρώτο χρήστη κάτι που μπορεί να τον ενδιαφέρει.



Εικόνα 2.4.3 Παράδειγμα Συστήματος Συστάσεων

2.4.4 Προσομοίωση Χρηστών

Εφόσον η μοντελοποίηση ενός χρήστη επιτρέπει στο σύστημα να πραγματοποιήσει μια εσωτερική αναπαράσταση ενός συγκεκριμένου χρήστη, έτσι με την προσομοίωση χρηστών μας δίνεται η δυνατότητα να πραγματοποιήσουμε δοκιμές ευχρηστίας.

Οι δοκιμές αυτές είναι μια τεχνική που χρησιμοποιείται για την αξιολόγηση ενός προϊόντος δοκιμάζοντας την στους χρήστες αυτούς παρέχοντας την βασική ιδέα για το πως θα χρησιμοποιούσαν πραγματικοί χρήστες το σύστημα. [8]

Οι δοκιμές αυτές επικεντρώνονται στην μέτρηση της ικανότητας ενός προϊόντος να ικανοποιήσει έναν άνθρωπο. Παραδείγματα από προϊόντα που επωφελούνται από αυτές τις δοκιμές είναι τα τρόφιμα, καταναλωτικά προϊόντα, ιστοσελίδες, διεπαφές υπολογιστών, εγγράφων κ.α.

2.5 Εξόρυξη Γνώσης

Η εξόρυξη γνώσης, που ονομάζεται επίσης ως ανακάλυψη γνώσης σε βάσεις δεδομένων, στην επιστήμη των υπολογιστών είναι η διαδικασία της ανακάλυψης ενδιαφερόντων και χρήσιμων μοτίβων και σχέσεων σε μεγάλες ποσότητες δεδομένων. [9] Το πεδίο της εξόρυξης γνώσεων συνδυάζει εργαλεία και τεχνικές της τεχνητής νοημοσύνης με την διαχείριση βάσεων δεδομένων και χρησιμοποιείται ευρέως από επιχειρήσεις (ασφαλιστικές, τράπεζες κ.α.), στην έρευνα επιστημών (αστρονομία, ιατρική) και σε κυβερνητικά συστήματα ασφαλείας (ανίχνευση των εγκληματιών και τρομοκρατών). Έτσι με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης εξάγονται δεδομένα με σκοπό να βοηθήσουν τον άνθρωπο να πάρει τις κατάλληλες αποφάσεις.

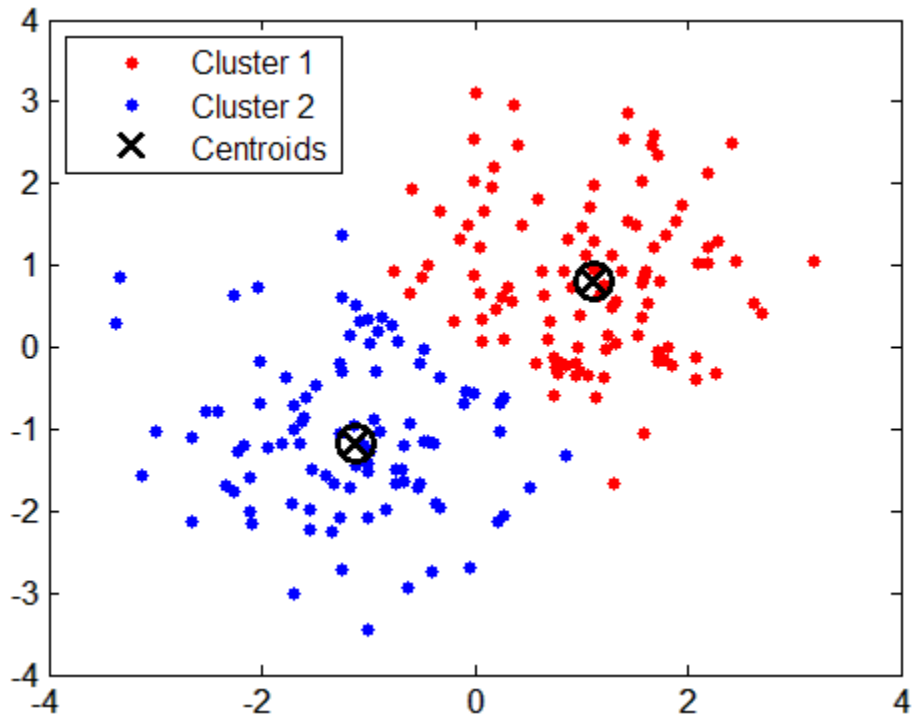
Καθώς οι δυνατότητες αποθήκευσης δεδομένων αυξήθηκαν σημαντικά κατά τη διάρκεια της δεκαετίας του 1980, όπου πολλές εταιρείες άρχισαν να αποθηκεύουν περισσότερα δεδομένα συναλλαγών, άρχισε να δημιουργείται ζήτηση για έμπειρα συστήματα ικανά να ανακαλύψουν την γνώση που βρίσκεται μέσα σε αυτά τα δεδομένα. Αυτό οδήγησε στην δημιουργία συνεδρίων περί εξόρυξης γνώσης με σκοπό της λύσης αυτού του προβλήματος. Μια από τις πρώτες επιτυχημένες εφαρμογές εξόρυξης δεδομένων ήταν η ανίχνευση για απάτες στις πιστωτικές

κάρτες. Το σύστημα μελετούσε την αγοραστική συμπεριφορά του καταναλωτή και εμφάνιζε ένα μοτίβο για αυτόν. Οποιαδήποτε αγορά γινόταν έξω από αυτό το μοτίβο οδηγούσε σε έρευνα.

Η πλήρης διαδικασία εξόρυξης δεδομένων περιλαμβάνει πολλαπλά στάδια τα οποία αποτελούν την συλλογή πληροφοριών, την προεπεξεργασία όπου πριν την εφαρμογή των αλγορίθμων εξόρυξης δεδομένων το ερευνώμενο σύνολο πληροφοριών συναρμολογείται. Ύστερα γίνεται μετασχηματισμός των δεδομένων η οποία επιτρέπει την εξόρυξη των δεδομένων και τελικά καταλήγει στην ερμηνεία των δεδομένων την βάση. Για την επίτευξη της προαναφερθέντας διαδικασίας υπάρχουν κάποιες τεχνικές οι οποίες αναλύονται παρακάτω.

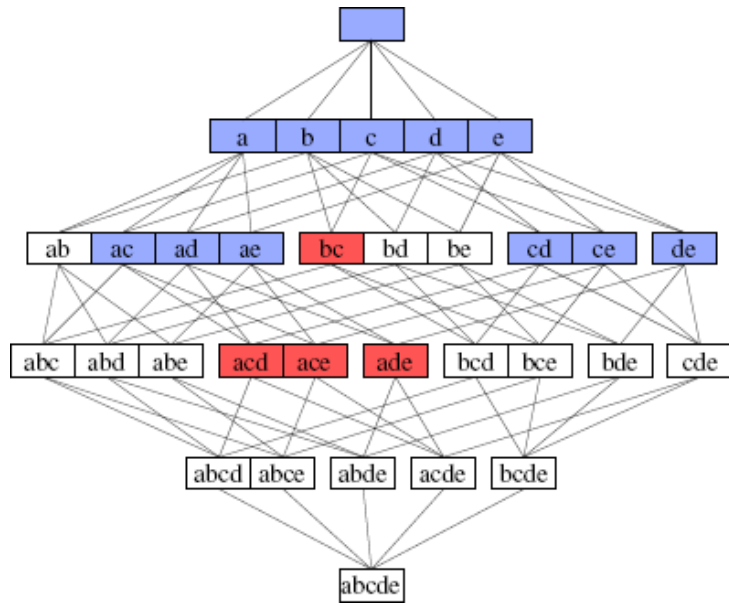
Η προγνωστική μοντελοποίηση χρησιμοποιείται όταν ο στόχος μας είναι να εκτιμηθεί η τιμή ενός συγκεκριμένου χαρακτηριστικού και μας είναι γνωστές μερικές από τις τιμές του χαρακτηριστικού. Ένα παράδειγμα είναι η ταξινόμηση δεδομένων η οποία λαμβάνει ένα σύνολο από δεδομένα που έχουν είδη χωριστεί σε προκαθορισμένες ομάδες και αναζητά μοτίβα στα δεδομένα που διαφοροποιούν αυτές τις ομάδες. Τα ανακαλυφθέντα μοτίβα αυτά μπορούν στην συνέχεια να ξαναχρησιμοποιηθούν για την ταξινόμηση άλλων δεδομένων όπου η ονομασία για το χαρακτηριστικό της ομάδας είναι άγνωστο. Για παράδειγμα ένας κατασκευαστής θα μπορούσε να αναπτύξει ένα προγνωστικό μοντέλο που διακρίνει τα ανταλλακτικά που αποτυγχάνουν σε ακραία ζεστές ή κρύες θερμοκρασίες.

Μια δεύτερη τεχνική είναι η περιγραφική μοντελοποίηση ή συσταδοποίηση η οποία χωρίζει επίσης τα δεδομένα της σε ομάδες. Με την συσταδοποίηση οι κατάλληλες ομάδες μπορεί να μην είναι γνωστές εκ των προτέρων αλλά ανακαλύπτονται ύστερα από ανάλυση των δεδομένων. Για παράδειγμα, ένας διαφημιστής μπορεί να αναλύσει ένα γενικό πληθυσμό προκειμένου να χαρακτηρίσει τους πιθανούς πελάτες σε διαφορετικές ομάδες και στην συνέχεια να αναπτύξει ξεχωριστές διαφημιστικές καμπάνιες. Στην εικόνα 2.5.1 φαίνεται ένα απλό παράδειγμα συσταδοποίησης.



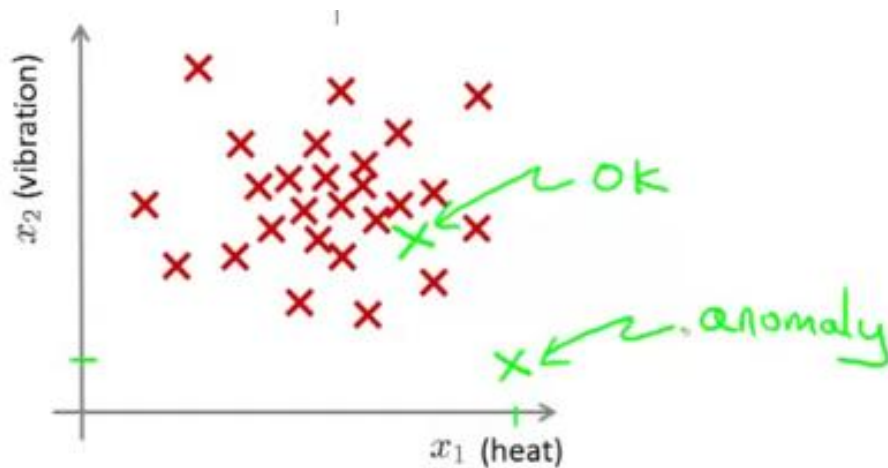
Εικόνα 2.5.1 Συσταδοποίηση

Η επόμενη τεχνική εξόρυξης δεδομένων που θα αναλύσουμε είναι η εξόρυξη μέσω μοτίβων. Αυτή η τεχνική επικεντρώνεται στον προσδιορισμό κανόνων που περιγράφουν συγκεκριμένα μοτίβα μέσα στα δεδομένα. Χρησιμοποιούνται συχνά σε καταστήματα που προσπαθούν να βρουν ποια προϊόντα αγοράζονται συνήθως μαζί με κάποια άλλα προϊόντα. Αν και οι δοκιμές σε τέτοιες γνώσεις είναι εφικτές και χωρίς την βοήθεια μιας εφαρμογής, η εξόρυξη δεδομένων επέτρεψε την ανακάλυψη ενώσεων σε λιγότερο εμφανή συνόλων δεδομένων. Στην εικόνα 2.5.2 παρουσιάζεται με απλό τρόπο πως η τεχνική εξόρυξης μοτίβων χρησιμοποιείται στα δεδομένα.



Εικόνα 2.5.2 Εξόρυξη Μοτίβων

Τέλος καταλήγουμε στην τεχνική εντοπισμού σφαλμάτων η οποία μπορεί να θεωρηθεί ως την άλλη όψη της συσταδοποίησης η οποία βρίσκει περιπτώσεις δεδομένων που είναι ασυνήθιστες και δεν ταιριάζουν σε κάποιο καθιερωμένο πρότυπο. Πιο πριν αναφέραμε την εξόρυξη δεδομένων ως έναν τρόπο για τον εντοπισμό απάτης σε πιστωτικές κάρτες. Ο εντοπισμός σφαλμάτων είναι ένας τρόπος με τον οποίον μπορεί να αντιμετωπιστεί το πρόβλημα αυτό. Στην εικόνα 2.5.3 φαίνεται πως η τεχνική αυτή είναι ικανή να ανιχνεύσει σφάλματα.



Εικόνα 2.5.3 Εντοπισμός Σφαλμάτων

Κεφάλαιο 3

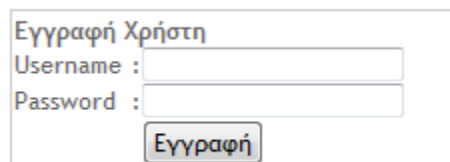
Δημιουργία και Ανάλυση των Προφίλ των Χρηστών

3.1 Δημιουργία Προφίλ

Η δημιουργία ενός προφίλ που πρόκειται να αναλυθεί αποτελείται από δύο βασικά βήματα. Το πρώτο βήμα είναι να μαζευτούν πληροφορίες από τον χρήστη κατά την εγγραφή του στην εφαρμογή όπου και καλείται να συμπληρώσει μια φόρμα με ένα σύνολο από προσωπικά στοιχεία. Το δεύτερο βήμα είναι πιο αυτοματοποιημένο και σκοπός είναι να βελτιώσει την δημιουργία του προφίλ. Αυτό μπορεί να γίνει μέσω της πλοήγησης του χρήστη στην εφαρμογή ή και μέσω εξωτερικών πηγών όπως για παράδειγμα είναι τα κοινωνικά δίκτυα (facebook, twitter κ.α.) όπου μπορεί ο χρήστης να δημοσιεύσει τα ενδιαφέροντά του.

Το πρόβλημα του χρήστη με την συλλογή πληροφοριών, με αυτόν τον τρόπο, είναι η έλλειψη ελέγχου αυτών των πληροφοριών στο προφίλ του και συχνά μπορεί οι χρήστες να μην είναι πρόθυμοι να συμβάλουν στην ενίσχυση του προφίλ τους. Για αυτόν τον λόγο, στα πλαίσια της πτυχιακής εργασίας, έπρεπε να δημιουργηθεί ένα σύστημα στο οποίο οι χρήστες να μην χρειάζεται να αλληλεπιδρούν με το προφίλ του χρήστη τους. Αντιθέτως το σύστημα αυτόματα το συμπληρώνει και το αναπτύσσει ώστε στο τέλος να υπάρξει ένα πλήρως ολοκληρωμένο προφίλ.

Για την δημιουργία του προφίλ απαιτείται μόνο από τον χρήστη να το δημιουργήσει και να εισέλθει στο σύστημα με το αναγνωριστικό του. Στην παρακάτω εικόνα, 3.1 φαίνεται η μοναδική διαδικασία που πρέπει να ακολουθήσει ο χρήστης για να δημιουργήσει το προφίλ του.



Εγγραφή Χρήστη
Username :
Password :
Εγγραφή

Εικόνα 3.1 Εγγραφή Χρήστη

3.2 Δομή Προφίλ

Σύμφωνα με την παραπάνω περιγραφή που αναφέρεται στα χαρακτηριστικά του προφίλ του χρήστη μπορούμε να χωρίσουμε το προφίλ σε υποκατηγορίες. [10]

Η εικονική ταυτότητα είναι το πρώτο πράγμα που επιλέγει ο χρήστης και αναφέρεται στο αναγνωριστικό του. Η ταυτότητα αυτή παραμένει μόνιμα και δεν αλλάζει και είναι επιλογή του χρήστη αν θέλει κάποιο ψευδώνυμο ή την πραγματική του ταυτότητα.

Το βασικό προφίλ είναι αυτό που περιέχει τις πολύ βασικές πληροφορίες του χρήστη (δημογραφικά δεδομένα) και συνήθως μπορούν να αλλάξουν, αν και σπανίως αλλάζουν, σύμφωνα με τις ανάγκες του χρήστη.

Το επεκταμένο προφίλ περιέχει πληροφορίες που αλλάζουν με την πάροδο του χρόνου και δεν καθορίζονται κατά τη δημιουργία του προφίλ. Οι πληροφορίες μπορούν να αλλάξουν ή και ακόμη να καταχωρηθούν νέες κάνοντας το προφίλ δυναμικό.

Αλληλεπίδραση με τρίτα προφίλ και πολιτικές που αφορούν τις ρυθμίσεις που σχετίζονται με την ασφάλεια των δεδομένων και την δηκτικότητα του χρήστη ως προς το ποιός μπορεί να χρησιμοποιήσει τις πληροφορίες αυτές.

Καθώς όλα αυτά τα χαρακτηριστικά αποτελούν την δομή ενός ολοκληρωμένου προφίλ υπάρχουν και διαφορετικά πρότυπα σχεδιασμού των προφίλ ή και συχνά ένα μείγμα από τα πρότυπα αυτά.

Τα στατικά μοντέλα είναι τα βασικά είδη προφίλ χρήστη. Σε αυτά μαζεύονται κυρίως δεδομένα τα οποία δεν πρόκειται να αλλάξουν πάλι, είναι δηλαδή στατικά. Οι μεταβολές στις επιλογές του χρήστη δεν καταχωρούνται στο σύστημα και δεν χρησιμοποιούνται αλγόριθμοι με στόχο να παραμετροποιήσουν το προφίλ.

Τα δυναμικά μοντέλα επιτρέπουν μια πιο επίκαιρη εκπροσώπηση των χρηστών. Γίνονται συχνά αλλαγές σε αυτά με την πάροδο του χρόνου και με την αλληλεπίδραση του χρήστη με το σύστημα. Τα προφίλ αυτά είναι ιδιαίτερα χρήσιμα στα προσαρμοστικά υπερμέσα καθώς ενημερώνονται για να λάβουν υπόψη τις τρέχουσες ανάγκες και στόχους του χρήστη.

Τα υβριδικά μοντέλα είναι αυτά που συνδυάζουν τα στατικά και τα δυναμικά μοντέλα σύμφωνα με τις ανάγκες του συστήματος.

3.3 Παρακολούθηση Προφίλ

Για να αναλυθεί ένα προφίλ, θα πρέπει πρώτα να γίνει εκτεταμένη παρακολούθηση του και καταγραφή όλων των ενεργειών του χρήστη με την πάροδο του χρόνου.

Η παρακολούθηση ενός προφίλ αποτελείται από τρεις διαδικασίες: [11]

- Άμεση παρακολούθηση της χρήσης της εφαρμογής κρατώντας ιστορικό του τρόπου χρήσης.
- Αποθήκευση του ιστορικού από το σύστημα για αποφυγή αποτυχιών.
- Άμεση ανατροφοδότηση σχετικά με την εκτέλεση της υπηρεσίας.

Φυσικά οι πληροφορίες αυτές είναι ιδιαίτερα πολύτιμες και ο κίνδυνος της παραβίασης της ιδιωτικής ζωής του χρήστη είναι υψηλή και γι' αυτό το θέμα αυτό εγείρει ηθικά και νομικά θέματα σχετικά με την επιτήρηση της προστασίας της ιδιωτικής ζωής.

3.4 Συλλογή Δεδομένων

Αφού δημιουργηθεί το προφίλ του χρήστη το επόμενο βήμα είναι να συγκεντρωθούν πληροφορίες για τον χρήστη έτσι ώστε τελικά να αναλυθούν. Υπάρχουν αρκετοί τρόποι να μαζευτούν πληροφορίες για τους χρήστες με μερικούς από αυτούς να αναλύονται παρακάτω. [12]

Ο πιο εύκολος και γρήγορος τρόπος συλλογής πληροφοριών είναι μέσω της άμεσης αλληλεπίδρασης του χρήστη με το σύστημα, όπου ο χρήστης καλείται να απαντήσει σε μια λίστα με ερωτήματα που θα βοηθήσουν το σύστημα να “μάθει” για τον χρήστη. Αυτή η διαδικασία πραγματοποιείται συνήθως κατά την εγγραφή του στο σύστημα όπου ο χρήστης καλείται να συμπληρώσει φόρμες ή άλλες διασυνδέσεις που εξυπηρετούν αυτόν τον σκοπό. Συνήθως, αυτού του είδους πληροφορίες είναι προαιρετικές καθώς οι χρήστες μπορεί να μην είναι διατεθειμένοι να συμπληρώσουν μακρά έντυπα καθώς επίσης οι πληροφορίες αυτές σπανίως αλλάζουν με το πέρασ του χρόνου. Σε γενικές γραμμές, οι πληροφορίες που συγκεντρώνονται με αυτόν τον τρόπο είναι δημογραφικές, όπως η ηλικία, το φύλλο ή την οικογενειακή κατάσταση του χρήστη.

Υπάρχουν όμως αρκετά προβλήματα με την συλλογή πληροφοριών με τον πρώτο τρόπο καθώς οι χρήστες μπορεί να μην θέλουν να παρέχουν πολλές πληροφορίες και αυτό οδήγησε στην δημιουργία ενός δεύτερου τρόπου ο οποίος μαθαίνει τις προτιμήσεις του χρήστη παρατηρώντας τον να αλληλεπιδρά με το σύστημα. Σε αυτή τη περίπτωση το σύστημα δεν ζητά άμεσα πληροφορίες, για τις προτιμήσεις, από τον χρήστη. Αντιθέτως οι πληροφορίες αυτές έρχονται κατά την πλοήγηση του στο σύστημα όπου υποσυνείδητα καλείται να πάρει κάποιες αποφάσεις και έτσι το σύστημα μαθαίνει δυναμικά από την παρατήρηση των αλληλεπιδράσεων τους. Για αυτόν τον λόγο για να μάθει το σύστημα για ένα προφίλ θα πρέπει η συμπεριφορά του χρήστη να είναι επαναλαμβανόμενη, δηλαδή οι ενέργειες του χρήστη θα πρέπει να εκτελούνται υπό παρόμοιες συνθήκες σε διαφορετικά χρονικά σημεία.

Υπάρχει επίσης και ένας τρίτος υβριδικός τρόπος ο οποίος αποτελεί έναν συνδυασμό των δύο παραπάνω. Δηλαδή συλλέγονται δεδομένα όχι μόνο καλώντας τον χρήστη να απαντήσει άμεσα σε ερωτήσεις, αλλά και κατά την διάρκεια της αλληλεπίδρασης του με το σύστημα. Αυτός ο τρόπος συνδυάζει τα πλεονεκτήματα των δύο προηγούμενων καθιστώντας τον έτσι και ιδανικό για τα περισσότερα συστήματα ανάλυσης προφίλ.

Κάθε μέθοδος έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Η πρώτη μέθοδος είναι συνήθως η καλύτερη όταν χρειάζεται να συλλεχθούν δεδομένα γρήγορα όμως υπάρχουν διάφορα προβλήματα. Πρώτον, της λείπει η ικανότητα να προσαρμόζεται στις αλλαγές και στις προτιμήσεις του χρήστη. Δεύτερον, εξαρτάται σε μεγάλο βαθμό από την διάθεση που έχει ο χρήστης να παρέχει τις πληροφορίες και είναι πιθανό να μην ισχύουν μετά από ένα διάστημα. Τρίτον, οι χρήστες μπορεί να μη γράψουν αληθή στοιχεία στις φόρμες και όσοι είναι πρόθυμοι να παρέχουν αληθή στοιχεία μπορεί να μην ξέρουν πως να εκφράσουν τα ενδιαφέροντά τους. Ωστόσο με αυτόν το τρόπο οι χρήστες έχουν τον πλήρη έλεγχο των πληροφοριών που συλλέγονται και είναι δική τους απόφαση για το τι θέλουν να μοιραστούν με το σύστημα.

Με την δεύτερη μέθοδο οι πληροφορίες μαζεύονται παρατηρώντας τις κινήσεις του χρήστη στο σύστημα έτσι παίρνει περισσότερο χρόνο να συγκεντρωθούν και οι πληροφορίες αυτές δεν είναι δυνατό να αλλαχθούν ή να τις δουν οι χρήστες. Επιπλέον αν δεν υπάρχει επανάληψη στις ενέργειες του χρήστη δεν θα μπορεί να ανακαλυφθεί το μοτίβο. Παρόλα αυτά οι πληροφορίες αυτές μπορούν εύκολα και αυτόματα να αλλάξουν έτσι ώστε το σύστημα να είναι πάντα ενήμερο και πιο ακριβής για τις προτιμήσεις του χρήστη.

Η υβριδική μέθοδος επιχειρεί να συνδυάσει τα πλεονεκτήματα και των δύο προηγούμενων μεθόδων, ζητώντας άμεσα από τους χρήστες να δώσουν όσο το δυνατόν περισσότερες πληροφορίες μπορούν και ύστερα το σύστημα, παρατηρώντας την αλληλεπίδραση τους,

αναπροσαρμόζει το προφίλ του χρήστη σύμφωνα με τις προτιμήσεις του.

3.5 Ανάλυση Δεδομένων

Η ανάλυση δεδομένων είναι μια διαδικασία για την επιθεώρηση, τον καθαρισμό, την μετατροπή και την μοντελοποίηση των δεδομένων με στόχο την ανακάλυψη πληροφοριών οι οποίες είναι χρήσιμες για την λήψη αποφάσεων από τους χρήστες. Η ανάλυση των δεδομένων μπορεί να διακριθεί σε διάφορες φάσεις όπως παρουσιάζονται παρακάτω. [13]

Η συλλογή δεδομένων όπως παρουσιάζεται και στο κεφάλαιο 3.4 εκ των οποίων οι απαιτήσεις καθορίζονται με βάση εκείνων που κατευθύνουν την ανάλυση δεδομένων.

Η επεξεργασία των δεδομένων είναι οι φάσεις όπου οι πρώτες πληροφορίες υποβάλλονται σε επεξεργασία και μετατρέπονται σε πληροφορίες οι οποίες είναι έτοιμες να αναλυθούν. Αυτό μπορεί να συνεπάγεται με την εισαγωγή των δεδομένων σε γραμμές και στήλες με μορφή πίνακα, όπως είναι ένα λογιστικό φύλλο ή μια βάση δεδομένων.

Μοντελοποίηση των δεδομένων είναι η διαδικασία όπου εφαρμόζονται μαθηματικοί τύποι ή αλγόριθμοι στα δεδομένα ώστε να εμφανιστούν οι σχέσεις μεταξύ των μεταβλητών έτσι ώστε τελικά οι πληροφορίες να μπορούν να οπτικοποιηθούν για να είναι κατανοητές από τον χρήστη.

Όλα τα παραπάνω όμως εξαρτώνται από την αρχική φάση της ανάλυσης των δεδομένων η οποία αποτελείται από τέσσερα ερωτήματα. Τα ερωτήματα αυτά έχουν να κάνουν με την ποιότητα των δεδομένων, την ποιότητα των μετρήσεων, την μετατροπή των δεδομένων και αν οι πληροφορίες που συλλέγονται πληρούν τις προϋποθέσεις του σχεδιασμού της έρευνας. [14]

3.6 Οπτικοποίηση Δεδομένων

Το τελικό στάδιο της ανάλυσης του προφίλ των χρηστών αποτελεί η οπτικοποίηση των δεδομένων. Η οπτικοποίηση των δεδομένων θεωρείται ως επιστημονικός κλάδος της οπτικής επικοινωνίας. Σκοπός της είναι η δημιουργία μιας οπτικής αναπαράστασης των δεδομένων ώστε να είναι κατανοητά στον χρήστη.

Υπάρχουν δύο τύποι δεδομένων των οποίων ο συνδυασμός αποτελεί μια ολοκληρωμένη οπτικοποίηση δεδομένων. [15] Το ένα είναι οι ετικέτες κειμένου που περιγράφουν την φύση των δεδομένων (ηλικία, όνομα κ.α.) και το δεύτερο είναι το αριθμητικό μέτρο που εκπροσωπεί την αντίστοιχη ετικέτα, όπως για παράδειγμα το '25' εκπροσωπεί την ηλικία σε έτη.

Όπως υπάρχουν δύο τύποι δεδομένων, έτσι υπάρχουν και δύο τρόποι αναπαράστασης των πληροφοριών για την οπτικοποίηση δεδομένων. [15] Αυτοί είναι μέσω πίνακα και μέσω γραφήματος. Ο πίνακας περιέχει συνήθως ποσοτικά δεδομένα τα οποία είναι οργανωμένα σε γραμμές και στήλες κατηγοριοποιημένα μέσω των ετικετών. Ένα απλό παράδειγμα παρουσιάζεται στην εικόνα 3.6.1.

	Weighting	Jupiter	Mars	Mercury	Neptune	Saturn	Venus
CURRENT OFFERING	50%	4.62	3.87	2.73	4.15	2.76	3.96
Metrics, dimensions, and correlations	15%	4.46	4.25	1.99	4.57	3.94	4.19
Data handling	10%	5.00	3.80	2.12	5.00	4.20	4.12
Reporting and analysis	30%	4.60	3.95	2.75	4.53	3.01	3.99
Integration	15%	4.00	3.80	3.20	2.40	1.40	3.60
Application usability and administration	20%	5.00	3.83	2.98	4.40	2.40	4.30
Service and support	10%	4.68	3.30	3.18	3.68	1.58	3.22
STRATEGY	50%	4.70	3.70	3.35	4.00	2.16	4.20
Corporate strategy	50%	4.40	3.70	3.00	4.40	3.60	4.30
Product strategy	15%	5.00	3.00	3.00	5.00	2.40	4.30
Quality of references	35%	5.00	4.00	4.00	3.00	0.00	4.00
MARKET PRESENCE	0%	4.55	2.25	3.58	4.25	1.60	3.30
Installed base	50%	4.70	1.70	3.30	4.30	1.60	3.40
Company Financials	25%	4.40	3.20	3.00	3.40	1.60	3.00
Employees	25%	4.40	2.40	4.70	5.00	1.60	3.40

All scores are based on a scale of 0 (weak) to 5 (strong).

Εικόνα 3.6.1 Οπτικοποίηση μέσω πίνακα

Αντιθέτως το γράφημα χρησιμοποιείται κυρίως για να δείξει τις σχέσεις μεταξύ των δεδομένων και τις απεικονίζει με οπτικά αντικείμενα (γραμμές, μπαρ, κουκίδες κ.α.) μέσα σε μια περιοχή που οριοθετείται από έναν ή περισσότερους άξονες. Διάφορα παραδείγματα απεικόνισης δεδομένων με γραφήματα παρουσιάζονται στην εικόνα 3.6.2



Εικόνα 3.6.2 Παραδείγματα Γραφημάτων

Κεφάλαιο 4

Παρόμοια Συστήματα

4.1 Το Σύστημα WEST

Αναλύοντας συστήματα ανάλυσης χρηστών είναι σημαντικό να αναφερθούμε σε πρώιμα συστήματα τα οποία έγιναν πρωτοπόρα στον τομέα τους. Ένα από αυτά που μελέτησα κατά την διάρκεια της έρευνας μου ήταν το σύστημα WEST. [16]

Το σύστημα WEST ήταν ένα παιχνίδι καθοδήγησης για ένα παιχνίδι που λεγόταν HowTheWestWasWon(Πώς η Δύση Κατακτήθηκε). Σε αυτό το παιχνίδι οι παίχτες περιστρέφουν τρεις σβούρες και πρέπει να δημιουργήσουν αριθμητικές εκφράσεις, με τους αριθμούς που έτυχαν, χρησιμοποιώντας +,-,*,/ και κατάλληλες παρενθέσεις ώστε να καθοριστεί ποια θα είναι η τελική τιμή. Έτσι αν, για παράδειγμα, ο παίχτης έφερε 2, 3 και 4 με τις σβούρες, θα μπορούσε να δημιουργήσει την αριθμητική έκφραση $(2+3)*4=20$ και να προχωρήσει 20 θέσεις. Αν κάποιος παίχτης φτάσει σε μία πόλη (κάθε 10 θέσεις δηλαδή) αυτόματα προχωράει στην επόμενη και αν προσγειωθεί σε έναν αντίπαλο τότε τον στέλνει πίσω δύο πόλεις. Καθιστώντας έτσι ως βέλτιστη στρατηγική να πρέπει να υπολογίσει ο χρήστης όλες τις πιθανές κινήσεις που τον βάζουν πιο μπροστά από τους αντιπάλους του. Αναλύοντας έτσι το σύστημα τις κινήσεις των παιχτών ανακάλυψε ότι η πιο δημοφιλής στρατηγική ήταν η προσθήκη των δύο μικρότερων αριθμών και ο πολλαπλασιασμός τους από τον μεγαλύτερο.

Ο WEST τελικά λειτουργούσε σαν προπονητής των παιχτών βοηθώντας να βρουν την βέλτιστη στρατηγική. Αρχικά βαθμολογούσε τις κινήσεις τους και παρακολουθούσε πόσες φορές χρησιμοποιούσαν την ίδια τακτική και πόσες ευκαιρίες να πέσουν οι παίχτες πάνω σε πόλη ή αντίπαλο πήγαν χαμένες. Τέλος ανίχνευε πότε ένα σχέδιο ήταν λάθος και παρέμβαινε στο σχέδιο του παίχτη.

Παρά το γεγονός ότι το σύστημα WEST εξερεύνησε λίγες από τις βασικές έννοιες της μοντελοποίησης χρηστών, λόγω του ότι τα αποτελέσματα ήταν περιορισμένα, λειτούργησε πολύ καλά αναλύοντας

τις συμπεριφορές των παιχτών ώστε να είναι κατανοητές από τους χρήστες.

4.2 Το Σύστημα Gumsaws

Για την υποστήριξη της κατασκευής προσαρμοστικών ιστοσελίδων δημιουργήθηκε το σύστημα Gumsaws. [17] Το σύστημα αυτό είναι σε θέση να ικανοποιήσει τις ανάγκες επεκτασιμότητας, δυνατότητας αντικατάστασης και προσαρμοστικότητας μιας ιστοσελίδας μοντελοποιώντας τους χρήστες. Αυτό το κάνει χρησιμοποιώντας τεχνικές εξόρυξης γνώσης για να μάθει το ιστορικό πλοήγησης του χρήστη.

Το σύστημα Gumsaws διαθέτει λειτουργίες δημιουργίας προφίλ ή ομάδας προφίλ και την αποθήκευση, την ανάκτηση, την ενημέρωση και την διαγραφή καταχωρήσεων. Οι λειτουργίες αυτές εκτελούνται από το σύστημα με την χρήση διαφόρων πηγών πληροφόρησης όπως παρουσιάζονται παρακάτω. Οι άμεσες πληροφορίες οι οποίες προέρχονται κατευθείαν από τους χρήστες, οι ομαδικές πληροφορίες που προέρχονται από το ιστορικό πλοήγησης των χρηστών και οι συσχετίσεις μεταξύ τους είναι μερικές από τις πηγές πληροφοριών. Έτσι το σύστημα μπορεί να χρησιμοποιηθεί από ειδησεογραφικά συστήματα και να εξυπηρετεί τους χρήστες του σύμφωνα με τις προτιμήσεις τους.

Κεφάλαιο 5

Χαρακτηριστικά Συστήματος

5.1 Δυναμικές Ιστοσελίδες

Δυναμικές ονομάζονται οι ιστοσελίδες οι οποίες δημιουργούνται σε πραγματικό χρόνο. Αυτό σημαίνει ότι οι ιστοσελίδες αυτές έχουν μέσα κώδικα, όπως η PHP, ο οποίος ελέγχει και προσαρμόζει τα δεδομένα που εμφανίζονται στον περιηγητή του χρήστη.

Είναι δύσκολο να καθοριστεί το πότε ξεκίνησαν να δημιουργούνται δυναμικές ιστοσελίδες αλλά σίγουρα συνέβη κατά τη δεκαετία του '90 όπου παρατηρήθηκε ραγδαία εξέλιξη των ιστοσελίδων. Οι πρώτες δυναμικές ιστοσελίδες μπορεί να ήταν απλές, δηλαδή μόνο με ένα μενού επιλογών αλλά με την εξέλιξη της τεχνολογίας βρήκαν δρόμο στην επαγγελματική αγορά προς τα τέλη του 1990.

Οι δυναμικές ιστοσελίδες διαφέρουν από τις στατικές σε πολλούς τομείς. Οι στατικές ιστοσελίδες δεν αλλάζουν το περιεχόμενό τους και εμφανίζουν το ίδιο περιεχόμενο σε όλους τους χρήστες. Το περιεχόμενο αυτό συνήθως είναι απλό κείμενο, σύνδεσμοι ή ακόμη και βίντεο. Τα μόνα πλεονεκτήματα που έχουν οι σελίδες αυτές είναι ότι ο χρόνος παραγωγής, το κόστος δημιουργίας και το κόστος συντήρησης είναι μικρά. Αντίθετα οι δυναμικές ιστοσελίδες συνήθως είναι πιο αργές και πιο ακριβές στην δημιουργία και συντήρηση αλλά τα πλεονεκτηματά τους είναι πολλά. Το πιο σημαντικό είναι ότι προσφέρουν ένα πιο αποδοτικό σύστημα το οποίο είναι πιο εύκολο να αναβαθμιστεί ενώ ταυτόχρονα προσφέρει την μέγιστη χρησιμότητα τον χρήστη καθώς μπορεί όχι μόνο να αναβαθμίζει το περιεχόμενό του αλλά και να το αναπροσαρμόζει σύμφωνα με τις ανάγκες κάθε χρήστη ξεχωριστά.

Για να επιτευχθεί αυτή η διαδραστικότητα στην εμπειρία του χρήστη υπάρχουν δύο τρόποι. Ο ένας είναι με κωδικοποίηση από την πλευρά του χρήστη, που σημαίνει ότι είναι συνήθως γραμμένη σε HTML και μπορεί να έχει και κώδικα όπως η Javascript τα οποία "τρέχουν" μέσα από την ιστοσελίδα. Το περιεχόμενο των σελίδων αυτών αλλάζει σύμφωνα με τις επιλογές εισόδου του χρήστη, όπως είναι η ώρα, η μέρα ή και το πάτημα ενός κουμπιού στην ιστοσελίδα. Ο δεύτερος

τρόπος είναι με κωδικοποίηση από την πλευρά του διακομιστή. Αυτός ο τρόπος χρησιμοποιείται για να δώσουν μια πιο προσωποποιημένη διεπαφή στον χρήστη καθώς μπορεί να συγκεντρώσει πληροφορίες για τα χαρακτηριστικά του. Θεωρείται επίσης και πιο ασφαλής τρόπος καθώς ο κώδικας δεν είναι φανερός στον χρήστη σε αντίθεση με τον πρώτο τρόπο. Το αρνητικό του είναι ότι είναι συνήθως πιο αργός και αυτό εξαρτάται από τα αιτήματα που δέχεται ο διακομιστής καθώς πολλές φορές μπορεί να αποτρέψει και την σύνδεση του χρήστη στο σύστημα. Η κωδικοποίηση από πλευράς διακομιστή χρησιμοποιεί γλώσσες όπως για παράδειγμα η PHP, Perl, Ruby κ.τ.λ. αλλά συνήθως τις χρησιμοποιεί ταυτόχρονα με κώδικες από την πλευρά του χρήστη.

5.2 Τεχνολογίες Συστήματος

Η ζήτηση δυναμικών ιστοσελίδων έγινε αιτία της ανάπτυξης τεχνολογιών ικανές να τις κάνουν πραγματικότητα. Το σύστημα που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας αυτής χρησιμοποιεί τεχνολογίες που είναι ικανές να δημιουργούν ένα προφίλ για τον χρήστη και ύστερα από ανάλυση να εμφανίζονται οι προτιμήσεις του. Χρησιμοποιήθηκε ένα μεγάλο εύρος τεχνολογιών (CSS, PHP, Javascript κ.τ.λ.) που σε συνδυασμό μεταξύ τους δημιουργούν μια πλήρης δυναμική εφαρμογή. Στα παρακάτω υποκεφάλαια παρουσιάζονται αναλυτικά οι τεχνολογίες που χρησιμοποιήθηκαν για την επίτευξη αυτού του έργου.

5.2.1 HTML

Η HTML είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της αποτελούν τα βασικά δομικά στοιχεία των ιστοσελίδων. Ο περιηγητής με την σειρά του είναι ικανός να διαβάσει την γλώσσα αυτή και να την ερμηνεύσει έτσι ώστε ο χρήστης να μπορεί να την διαβάσει.

Η HTML δημιουργήθηκε από τεχνικούς της CERN και έκανε την πρώτη της εμφάνιση στις αρχές της δεκαετίας του '90 όπου γινόταν περιγραφή των στοιχείων που την αποτελούσαν και έτσι το 1995 δημιουργήθηκε μια πιο ολοκληρωμένη έκδοση πάνω στην οποία βασίστηκαν οι μεταγενέστερες.

Η γλώσσα σήμανσης αυτή από μόνη της δεν είναι ικανή να δημιουργήσει πολύπλοκες και διαδραστικές ιστοσελίδες παρά μόνο στατικές. Για αυτόν τον σκοπό χρησιμοποιείται η δυναμική HTML η οποία είναι ουσιαστικά η ίδια η HTML στην οποία έχουν προστεθεί γλώσσες ελέγχου εμφάνισης της γλώσσας σήμανσης, όπως η CSS και κώδικας από την πλευρά του χρήστη, όπως η Javascript, για να μπορεί να επικοινωνήσει με τον χρήστη.

Το σύστημα της πτυχιακής εργασίας, που είναι ένα ηλεκτρονικό κατάστημα χρησιμοποιεί την HTML δυναμικά (με CSS και Javascript) και κύριος στόχος της είναι να εμφανίζει το στατικό περιεχόμενο της ιστοσελίδας, δηλαδή τα κείμενα, τις εικόνες καθώς και την αρχική δομή της ιστοσελίδας προτού υποστεί επεξεργασία ως προς την εμφάνιση. Παράδειγμα χρήσης HTML μπορεί να είναι οι λίστες των υπερσυνδέσμων πριν επεξεργαστούν από την CSS.

```
<ul>
```

```
<li><a href="index.php" class="active">Κεντρική Σελίδα</a></li>
```

```
<li><a href="other/support.php">Υποστήριξη</a></li>
```

```
<li><a href="other/thestore.php">Το Κατάστημα</a></li>
```

```
<li><a href="other/contact.php">Επικοινωνία</a></li>
```

```
</ul>
```


5.2.2 CSS

Η CSS είναι μια γλώσσα φύλλων στυλ η οποία χρησιμοποιείται για τον έλεγχο της εμφάνισης μιας ιστοσελίδας που έχει γραφτεί με μια γλώσσα σήμανσης. Εμφανίστηκε πρώτη φορά το 1994, με την εξέλιξη της HTML, από τεχνικούς της CERN λόγω της ζήτησης περισσότερων στιλιστικών επιλογών για τις ιστοσελίδες.

Το ηλεκτρονικό κατάστημα που δημιουργήθηκε στα πλαίσια της αυτής της πτυχιακής κάνει χρήση της CSS για να κάνει την εμπειρία του χρήστη στην ιστοσελίδα πιο ευχάριστη και άνετη. Σκοπός της είναι να κάνει την ιστοσελίδα πιο εύκολη στο μάτι δίνοντας χρώματα και τονίζοντας τις λέξεις που χρειάζεται, κάνει το μενού πιο κατανοητό καθώς λέει στον χρήστη που βρίσκεται καθ' όλη τη διάρκεια της πλοήγησης και δίνει το κατάλληλο γραφικό περιβάλλον σε όλα τα πεδία της. Συνεχίζοντας το παράδειγμα από το προηγούμενο υποκεφάλαιο, ο κώδικας CSS για τις λίστες των υπερσυνδέσμων ώστε το μενού να φαίνεται πιο όμορφο και γραφικό.

```
#navigation { float:right; white-space:nowrap;}
```

```
#navigation ul{ list-style-type: none; height:64px; font-weight: bold; float:left;}
```

```
#navigation ul li{ float:left; display:inline; }
```

```
#navigation ul li a{ float:left; height:64px; line-height:64px; text-decoration: none; color:#fff; padding:0 15px;}
```

```
#navigation ul li a.active,
```

```
#navigation ul li a:hover{ background:#fff; color:#8b0000; }
```

5.2.3 Javascript

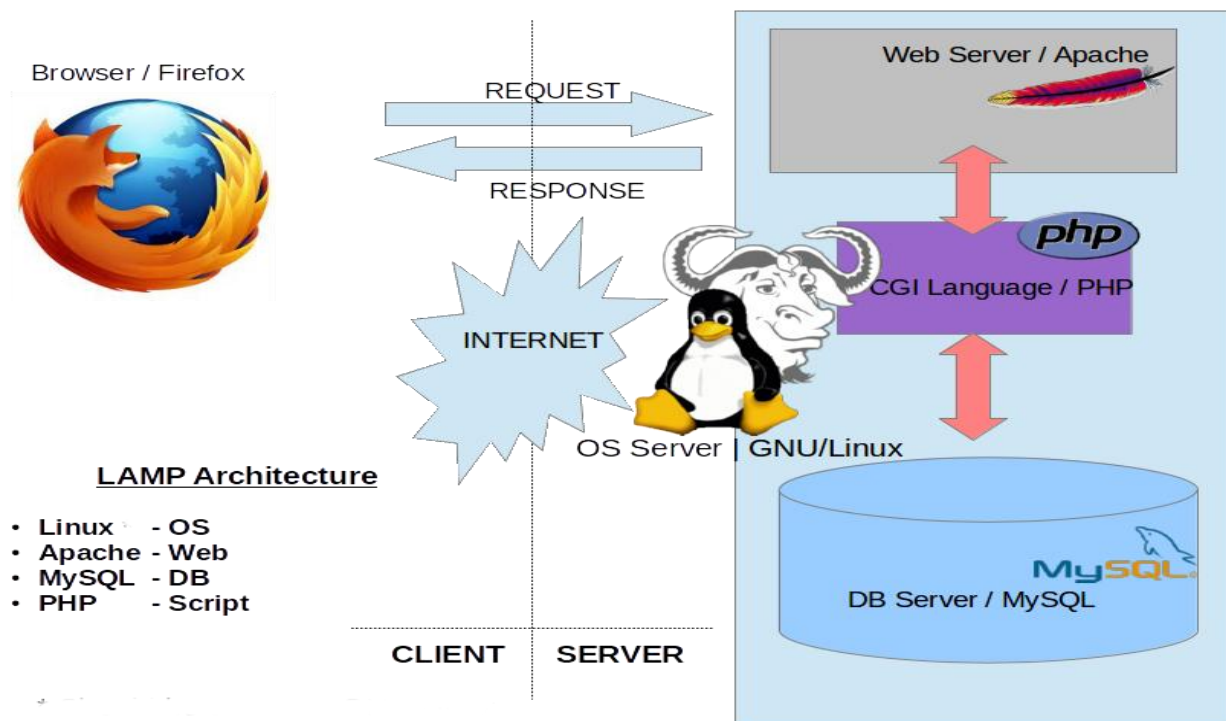
Η Javascript είναι μια διερμηνευμένη γλώσσα προγραμματισμού και χρησιμοποιείται από δυναμικές ιστοσελίδες που θέλουν να έχουν κωδικοποίηση από την πλευρά του χρήστη για να μπορούν να επικοινωνούν με αυτόν. Δημιουργήθηκε από το Brendan Eich το 1995 με την επωνυμία Mocha που ύστερα μετονομάστηκε σε LiveScript και τέλος σε Javascript. Είναι σημαντικό να ειπωθεί ότι η Javascript δεν πρέπει να συγχέεται με την Java καθώς αυτή είναι μια εντελώς διαφορετική γλώσσα προγραμματισμού με εντελώς διαφορετικές εφαρμογές. Η Javascript έχει δεχθεί όμως σημαντικές επιρροές από την C, ειδικά στην σύνταξη κάνοντας την και αυτή μια αντικειμενοστραφής γλώσσα.

Στο ηλεκτρονικό κατάστημα της πτυχιακής χρησιμοποιείται για να δώσει μια μικρή μορφή επικοινωνίας μεταξύ της ιστοσελίδας και του χρήστη. Χαρακτηριστικό παράδειγμα είναι ο κώδικας ο οποίος αποτρέπει στον χρήστη να αποστείλει την φόρμα αγορών κενή ή ελλιπή όπως για παράδειγμα, άμα ο χρήστης δεν έχει συμπληρωμένο το ονοματεπώνυμο, την διεύθυνση, το τηλέφωνο κτλ. του εμφανίζεται ένα μήνυμα στην οθόνη.

```
<script type="text/javascript">
function checkform(form) {
    var inputs = form.getElementsByTagName('input');
    for (var i = 0; i < inputs.length; i++) {
        if(inputs[i].hasAttribute("required")){
            if(inputs[i].value == ""){
                alert("Please fill all required fields");
                return false; } } }
    return true; }
</script>
```

5.2.4 Το Μοντέλο WAMP

Το μοντέλο WAMP ή LAMP είναι μια λίστα από τέσσερα προγράμματα ανοιχτού κώδικα όπου το αρκτικόλεξό τους αποτελεί το μοντέλο. Τα προγράμματα αυτά είναι: το λειτουργικό σύστημα Linux ή Windows, ο Apache HTTP Server, η βάση δεδομένων MySQL και η γλώσσα προγραμματισμού PHP. Το μοντέλο WAMP είναι κατάλληλο για την δημιουργία πολύπλοκων δυναμικών ιστοσελίδων αλλά και διαδικτυακών εφαρμογών. Η χρήση των συγκεκριμένων προγραμμάτων δεν είναι αναγκαία καθώς μπορούν να αντικατασταθούν από άλλα παρόμοια, όπως η MongoDB αντί της MySQL και η Perl αντί της PHP. Στα παρακάτω υποκεφάλαια περιγράφονται αναλυτικά τα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση του ηλεκτρονικού καταστήματος (Apache HTTP Server, MySQL και PHP). Στην παρακάτω εικόνα 5.2.4 φαίνεται ένα αρχιτεκτονικό σύστημα LAMP και τις συνδέσεις μεταξύ τους.



Εικόνα 5.2.4 Το Αρχιτεκτονικό Σύστημα LAMP

5.2.4.1 Apache FTTP Server

Ο Apache FTTP Server είναι το πλέον πιο διαδεδομένο λογισμικό διακομιστή διαδικτύου. Ο Apache είναι το πρόγραμμα που 'τρέχει' σε έναν υπολογιστή που είναι συνδεδεμένος στο διαδίκτυο ο οποίος με την σειρά του 'φιλοξενεί' μια ιστοσελίδα. Σκοπός του είναι να απαντάει σε αιτήσεις που δέχεται από προγράμματα περιήγησης του διαδικτύου και να αποκρίνεται στέλνοντας πίσω τα απαραίτητα στοιχεία. Αυτή η αλληλεπίδραση μεταξύ πελάτη-διακομιστή είναι η διαδικασία με την οποία λειτουργεί το διαδίκτυο. [18]

Το σχέδιο του προγράμματος Apache βγήκε στην επιφάνεια το 1995 ύστερα από μια συλλογική προσπάθεια να δημιουργηθεί ένα λογισμικό ανοιχτού κώδικα, εμπορικής ποιότητας, ικανό να λειτουργεί ως διακομιστής δικτύου. Μέχρι και σήμερα το σχέδιο για την ανάπτυξη του Apache αποτελεί έργο του Apache Software Foundation και εκατοντάδες χρήστες συνεισφέρουν με ιδέες, κώδικα και την τεκμηρίωση του έργου.

5.2.4.2 MySQL

Η MySQL είναι το πιο διαδεδομένο σύστημα διαχείρισης βάσεων δεδομένων και ανήκει στα συστήματα ανοιχτού κώδικα μιας και διατίθεται δωρεάν από την Oracle. Η MySQL έκανε την πρώτη της εμφάνιση το 1995 από μια Σουηδική εταιρεία, την MySQL AB και από εκεί και μέχρι και σήμερα συνεχίζει και αναβαθμίζει το αρχικό μοντέλο της.

Βάση δεδομένων είναι μια συλλογή από δεδομένα και ψηφιακών πληροφοριών των οποίων η ανάκτηση γίνεται μέσω ερωτημάτων (queries). Οι βάσεις δεδομένων χρησιμοποιούνται από αμέτρητες δυναμικές ιστοσελίδες καθώς μπορούν να κρατήσουν πληροφορίες σχετικά με τους χρήστες ή τα προϊόντα της. Χαρακτηριστικά παραδείγματα θα περιλαμβάνουν το ηλεκτρονικό σύστημα μιας βιβλιοθήκης ή ένα σύστημα κρατήσεων σε κάποιο αεροδρόμιο.

Στο ηλεκτρονικό κατάστημα της πτυχιακής η βάση δεδομένων παίζει καθοριστικό ρόλο καθώς κρατάει τα αρχεία όλων των χρηστών της ιστοσελίδας, όλων των προϊόντων και επίσης κρατάει σημειώσεις για την κινητικότητα των χρηστών στην. Από εκεί και πέρα, μέσω ερωτημάτων (queries), είναι σε θέση να εμφανίσει πληροφορίες σχετικά με τις προτιμήσεις και τα ενδιαφέροντα των χρηστών σαν σύνολο αλλά και για κάθε χρήστη ξεχωριστά.

5.2.4.3 PHP

Η PHP είναι μια γλώσσα προγραμματισμού από τη μεριά του διακομιστή η οποία είναι ειδικά φτιαγμένη για την δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Μπορεί πολύ εύκολα να συνδυαστεί με HTML και συνήθως αποτελεί ένα από τα κομμάτια των μοντέλων WAMP (ή LAMP). Η PHP αρχικά δημιουργήθηκε από έναν φοιτητή, τον Rasmus Lerdorf, το 1994 για να διατηρεί μια προσωπική του ιστοσελίδα και από εκεί και πέρα συνέχιζε να προσθέτει κώδικα μέχρι που τελικά κατέληξε να έχει δημιουργήσει μια καινούρια γλώσσα προγραμματισμού.

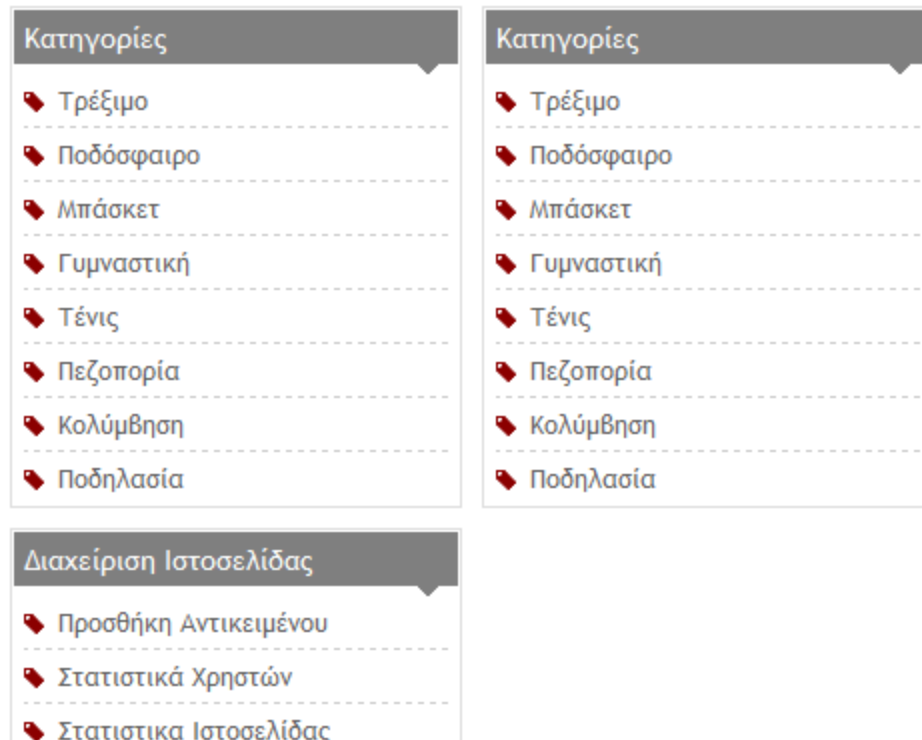
Η χρήση της γλώσσας PHP στο σύστημα του ηλεκτρονικού καταστήματος της πτυχιακής είναι το σημαντικότερο κομμάτι του συστήματος. Μέσω αυτής το σύστημα είναι σε θέση να εκτελέσει τις σημαντικότερες λειτουργίες του όπως να δημιουργεί νέους χρήστες, να προσθέτει νέα προϊόντα, να εκτελεί τα κατάλληλα ερωτήματα (queries) για την εμφάνιση των σωστών προϊόντων, να 'παρακολουθεί' την κινητικότητα των χρηστών και να την εμφανίζει είναι μερικά από τα σημαντικότερα.

5.3 Προφίλ Χρηστών και Sessions

Το πρώτο πράγμα που θα κάνει ο χρήστης μόλις ανοίξει την ιστοσελίδα του ηλεκτρονικού καταστήματος είναι να δημιουργήσει το προφίλ του χρήστη του. Τα δύο πράγματα που θα του ζητηθούν θα είναι ένα αναγνωριστικό όνομα (username) και ένας κωδικός πρόσβασης

(password) τα οποία θα είναι μοναδικά, αν δηλαδή κάποιος δεύτερος χρήστης προσπαθήσει να δημιουργήσει έναν λογαριασμό με κάποιο όνομα που χρησιμοποιείται είδη, θα του βγάλει ένα μήνυμα σφάλματος. Παράλληλα με το προφίλ του χρήστη δημιουργείται και ένα δεύτερο προφίλ στο οποίο δεν έχει άμεση πρόσβαση ο χρήστης το οποίο ανανεώνεται κάθε φορά που ο χρήστης επισκέπτεται την ιστοσελίδα, βλέπει τα προϊόντα ή πραγματοποιεί κάποια αγορά. Το δεύτερο προφίλ αυτό περιέχει στοιχεία τα οποία βοηθούν για να υπολογίσει το σύστημα αν ο χρήστης είναι άντρας ή γυναίκα, αν έχει παιδιά, με ποιο είδος άθλησης ασχολείται καθώς και πόσο συχνά επισκέπτεται και πραγματοποιεί αγορές. Τα στοιχεία αυτά είναι δυναμικά, το οποίο σημαίνει ότι αλλάζουν με το πέρασ του χρόνου. Επίσης, το σύστημα έχει σχεδιαστεί έτσι ώστε να μπορεί να ξεχωρίσει τον διαχειριστή (admin) από τους απλούς χρήστες. Όταν καταλάβει ότι ο χρήστης είναι διαχειριστής τότε του εμφανίζει παραπάνω επιλογές διαχείρισης ιστοσελίδας όπως δείχνει η εικόνα 5.3.

Για να μπορεί ο χρήστης να είναι συνεχώς στο σύστημα θα πρέπει η σύνδεσή του να διατηρείται. Αυτό γίνεται με sessions. Τα sessions επιτρέπουν την συνεχής επικοινωνία μεταξύ δύο ή περισσότερων συσκευών. Αυτή η επικοινωνία ξεκινάει από τη στιγμή που ο χρήστης πληκτρολογήσει το αναγνωριστικό του και συνδεθεί στο σύστημα, μέχρι τελικά να αποσυνδεθεί. Τα sessions συνδέονται άμεσα με τα cookies αλλά έχουν σημαντικές διαφορές. Τα cookies σε αντίθεση με τα sessions αποθηκεύουν τις πληροφορίες στον περιηγητή του χρήστη αντί για τον διακομιστή καθιστώντας τις έτσι λιγότερο ασφαλείς. Παρ' όλα αυτά οι πληροφορίες μένουν για περισσότερο καιρό ενεργές σε αντίθεση με τα sessions που χάνονται όταν ο χρήστης κλείσει τον περιηγητή του.




Εικόνα 5.3 Διαφορά απλού χρήστη με διαχειριστή

5.4 Η ιστοσελίδα

Έχοντας, λοιπόν, ο χρήστης εισέλθει πια στο σύστημα είναι έτοιμος να ξεκινήσει τις αγορές του. Η πλοήγηση του στην ιστοσελίδα ξεκινάει από την κεντρική σελίδα όπου παρατηρεί δύο μενού (τρία αν ο χρήστης είναι ο διαχειριστής). Το πρώτο είναι το μενού του καταστήματος όπου ο χρήστης μπορεί να βρει βασικές πληροφορίες και πώς να επικοινωνήσει με αυτό και το δεύτερο περιέχει όλες τις κατηγορίες με τα προϊόντα (τρέξιμο, ποδόσφαιρο κτλ.). Στην περίπτωση που έχει συνδεθεί ο διαχειριστής εμφανίζει άλλο ένα μενού το οποίο του δίνει την δυνατότητα να προσθέσει νέα αντικείμενα στην βάση δεδομένων και να δει διάφορες πληροφορίες για κάθε χρήστη ξεχωριστά ή σαν σύνολο. Τέλος υπάρχει το καλάθι αγορών όπου τα αντικείμενα που

θέλει να αγοράσει ο χρήστης κατευθύνονται αρχικά. Επιλέγοντας σε μια από τις κατηγορίες (π.χ. ποδόσφαιρο) ο χρήστης ξεκινά την αναζήτηση των προϊόντων. Επιλέγοντας τις κατηγορίες, ουσιαστικά, ο χρήστης θέτει ερωτήματα (queries) στην βάση δεδομένων και αυτή με την σειρά του εμφανίζει τα αντικείμενα που της ζητήθηκαν. Αν υποθέσουμε ότι ο χρήστης είναι άντρας που θέλει να αγοράσει παπούτσια σύμφωνα με τις επιλογές του θα εμφανιστεί το περιεχόμενο της εικόνας 5.4. Αφού επιλέξει ο χρήστης το παπούτσι που τον ενδιαφέρει το στέλνει στο καλάθι και από εκεί συμπληρώνοντας τα στοιχεία αποστολής ολοκληρώνει την πλοήγησή του στο σύστημα.

	
CLASSIC 30 TF	ABSOLADO INSTINCT FG
011000 Ανδρικό παπούτσι ποδοσφαίρου από την PRO TOUCH.	011001 Ανδρικό παπούτσι ποδοσφαίρου με τεχνολογία predator.
19.99 €	48.99 €
<input type="text" value="1"/> Add to cart	<input type="text" value="1"/> Add to cart

Εικόνα 5.4 Το αποτέλεσμα της αναζήτησης ανδρικών παπουτσιών

5.5 Η βάση δεδομένων

Όπως αναφερθήκαμε και σε προηγούμενο κεφάλαιο η βάση δεδομένων χρησιμοποιείται για τα στατικά δεδομένα των χρηστών που εισάγονται

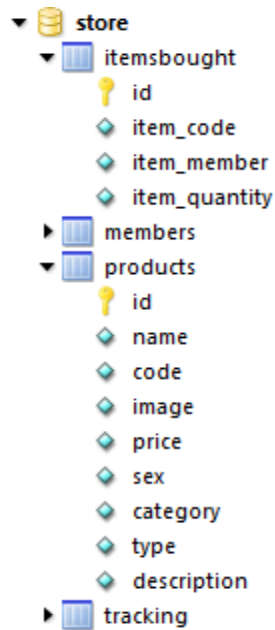
κατά την εγγραφή τους, τα δυναμικά δεδομένα όπου εισάγονται κατά την πλοήγησή τους, τα προϊόντα και την λίστα με όλες τις αγορές που έχουν πραγματοποιήσει οι χρήστες. Η βάση αποτελείται από τέσσερις πίνακες, τους members (χρήστες), products (προϊόντα), tracking (παρακολούθηση) και itemsbought (αγορές).

Ο πίνακας των χρηστών αποτελείται από τρία μόνο στοιχεία, το όνομα χρήστη, τον κωδικό πρόσβασης και ένα αναγνωριστικό (id) για κάθε χρήστη. Το αναγνωριστικό αυτό είναι μοναδικό για κάθε χρήστη και είναι το κλειδί που ενώνει αυτόν τον πίνακα με τον πίνακα παρακολούθησης.

Ο πίνακας παρακολούθησης περιέχει δεδομένα τα οποία προσπαθούν να προσδιορίσουν το αν ο χρήστης είναι άντρας ή γυναίκα, αν έχει παιδιά και ποιά είναι τα χόμπι του. Επίσης κρατάει αρχείο για το πότε συνδέθηκε τελευταία φορά, πόσες φορές έχει ψωνίσει κάτι από το κατάστημα, πόσα λεφτά έχει ξοδέψει και εφόσον έχει, μερικά προσωπικά του στοιχεία.

Ο πίνακας με τα προϊόντα περιέχει ένα προς ένα τις πληροφορίες και τις εικόνες αυτών καθώς και πληροφορίες που βοηθάνε το σύστημα να κατηγοριοποιήσει τα προϊόντα και να απαντήσει στα ερωτήματα (queries) που δέχεται από τον χρήστη κατά την διάρκεια των αγορών.

Τέλος ο πίνακας των αγορών (itemsbought) περιέχει πληροφορίες σχετικά με τις αγορές προϊόντων που έχει πραγματοποιήσει ο κάθε χρήστης ξεχωριστά. Στην εικόνα 5.5 φαίνονται οι πίνακες και μερικά από τα στοιχεία και κλειδιά που αποτελούν την βάση δεδομένων του συστήματος.



Εικόνα 5.5 Η Βάση Δεδομένων

5.6 Τεχνική παρακολούθησης χρηστών

Η διαδικασία της παρακολούθησης των χρηστών είναι και το σημείο όπου γίνεται και το 'χτίσιμο' των προφίλ δυναμικά. Κάθε φορά που κάποιος χρήστης πραγματοποιήσει ένα ερώτημα στην βάση δεδομένων αυτή εμφανίζει τα κατάλληλα προϊόντα και ταυτόχρονα σημειώνει, επεξεργάζοντας το προφίλ του χρήστη, τις κατηγορίες που τον ενδιαφέρουν.

Η επικοινωνία μεταξύ χρήστη και διακομιστή γίνεται με χρήση της γλώσσας προγραμματισμού PHP και μέσω αυτής στέλνεται το μήνυμα στην βάση δεδομένων για να πραγματοποιηθούν οι κατάλληλες εντολές. Η δυναμική αυτή επεξεργασία του προφίλ δεν είναι φανερή στον απλό χρήστη παρά μόνο στον διαχειριστή της ιστοσελίδας και δεν μπορεί να υποστεί καμία επεξεργασία παρά μόνο αν 'πειράξει' κανείς τις πληροφορίες στην βάση δεδομένων.

Αναφέραμε σε προηγούμενο κεφάλαιο τους τρόπους με τους οποίους είναι δυνατή η παρακολούθηση των προφίλ. Σε αυτή την εφαρμογή ο ιδανικός τρόπος είναι ο δεύτερος, δηλαδή, μέσω παρακολούθησης μέσα από τις ενέργειες του χρήστη. Με αυτόν τον τρόπο, παρατηρώντας το επαναλαμβανόμενο μοτίβο των χρηστών, μπορεί το σύστημα να προσαρμόζεται στις μεταβολές στα ενδιαφέροντα, τις προτιμήσεις, τις συνήθειες και τους στόχους του χρήστη. Το μόνο αρνητικό είναι ότι για να "χτιστεί" ένα ολοκληρωμένο προφίλ μπορεί να πάρει αρκετό χρόνο και εάν δεν του δοθεί αρκετός χρόνος ώστε να δημιουργηθούν κάποια επαναλαμβανόμενα μοτίβα από τον χρήστη τα στοιχεία μπορεί να φαίνονται ελλιπή.

Πιο αναλυτικά, ο τρόπος με τον οποίο παρακολουθείται ένα προφίλ έχει να κάνει με τις σελίδες που επισκέπτεται στην εφαρμογή. Εάν δηλαδή ο χρήστης επισκεφτεί ανδρικά προϊόντα το σύστημα θα το ξέρει και θα ανεβάσει τις φορές που έχει επισκεφτεί τα ανδρικά προϊόντα κατά ένα. Παρατηρώντας έτσι τον χρήστη για κάποια ώρα το σύστημα θα έχει αρκετές πληροφορίες για αυτόν έτσι ώστε να μπορεί ο διαχειριστής να τον ξεχωρίσει από τους υπόλοιπους.

5.7 Τεχνική ανάλυσης και εμφάνισης δεδομένων

Το τελικό στάδιο της εργασίας είναι να υπολογίζει και να εμφανίζει στατιστικά στοιχεία ανάλογα με τις προτιμήσεις του κάθε χρήστη ξεχωριστά. Η επιλογή αυτή είναι ορατή μόνο στον διαχειριστή της εφαρμογής και επιτρέπει σε αυτόν να αναζητήσει κάποιον χρήστη. Η εφαρμογή με την σειρά της αναζητά τον χρήστη στην βάση δεδομένων και όλα τα δεδομένα τα οποία τον αποτελούν. Ύστερα υπολογίζει τα δεδομένα και τα εμφανίζει έτσι ώστε να είναι κατανοητά από τον διαχειριστή.

Η ανάλυση είναι η διαδικασία όπου το σύστημα παίρνει τις πληροφορίες όπου ο χρήστης έβλεπε αντρικά ή γυναικεία προϊόντα και τις κατηγορίες τους και τα υπολογίζει ως ποσοστά σύμφωνα με τις επιλογές του. Η

εμφάνιση των δεδομένων γίνεται μέσω πινάκων όπου εμφανίζονται όλες οι κατηγορίες και ο διαχειριστής μπορεί να δει ξεκάθαρα δημογραφικά στοιχεία και τα ενδιαφέροντα του χρήστη.

Τα περισσότερα δεδομένα εμφανίζονται ως έχουν, για παράδειγμα το σημείο όπου λέει πότε δημιουργήθηκε ο χρήστης δεν αλλάζει και το πότε έγινε η τελευταία σύνδεση απλά περνάει τα δεδομένα από πάνω. Για τα υπόλοιπα στατιστικά όμως που απαιτείται να βγάλει κάποιο στατιστικό ο υπολογισμός γίνεται βάση μερικών μαθηματικών πράξεων πάνω στα δεδομένα της βάσης δεδομένων. Αν για παράδειγμα ο χρήστης έχει επισκεφτεί την κατηγορία "τρέξιμο" 4 φορές και καμία φορά τις υπόλοιπες, η εφαρμογή θα υπολογίσει ότι τα ενδιαφέροντά του είναι 100% "τρέξιμο" και 0% τα υπόλοιπα. Αν επισκεφτεί κάποια άλλη κατηγορία αργότερα το ποσοστό θα πέσει, για παράδειγμα 80% "τρέξιμο" και 20% "τένις".

Στην εικόνα 5.7 παρουσιάζεται ένα παράδειγμα όπου ο χρήστης (test2) πραγματοποίησε αγορές στο ηλεκτρονικό κατάστημα. Σύμφωνα με τα στοιχεία που βλέπουμε στην εικόνα παρατηρούμε ότι ο χρήστης είναι γυναίκα και από τον χρόνο που πέρασε στην ιστοσελίδα μπορούμε να υποθέσουμε ότι τα χόμπι του είναι το τρέξιμο. Επίσης βλέπουμε ότι είναι παλιός χρήστης, έχει πραγματοποιήσει αγορά του ύψους 131,98 € μια λίστα με τα προϊόντα που έχει αγοράσει καθώς και μερικά προσωπικά στοιχεία που εισήγαγε κατά τη διάρκεια των αγορών του.

Όνομα Χρήστη: test2

Ο χρήστης είναι άντρας: 0.0000 %

Ο χρήστης είναι γυναίκα: 100.0000 %

Ο χρήστης δεν έχει παιδί.

Σύμφωνα με τις επιλογές του χρήστη, το ποσοστό επισκεψιμότητας κάθε δραστηριότητας έχει ως εξής:

Τρέξιμο: 100.0000 %

Ποδόσφαιρο: 0.0000 %

Μπάσκετ: 0.0000 %

Γυμναστική: 0.0000 %

Τένις: 0.0000 %

Πεζοπορία: 0.0000 %

Κολύμβηση: 0.0000 %

Ποδηλασία: 0.0000 %

Ο χρήστης δημιουργήθηκε στις: 2015-07-17

Η τελευταία σύνδεση έγινε στις: 2015-11-25

Ο χρήστης έχει συνδεθεί: 2 φορές από την στιγμή που δημιούργησε τον χρήστη του.

Ο χρήστης έχει ψωνίσει απο το e-shop: 1 φορές.

Ο χρήστης έχει ξοδέψει: 131.98 € σε αγορές στο e-shop.

Προσωπικά στοιχεία χρήστη:

Πόλη: test

Διεύθυνση: test

E-mail: test@test.com

Τηλέφωνο: 1234567890

Ο χρήστης έχει αγοράσει τα παρακάτω προϊόντα:

Όνομα Προϊόντος	Κωδικός	Φύλο	Κατηγορία	Τύπος	Τιμή	Ποσότητα	Τελική Τιμή
8" FILAMENT SHORT	001001	women	running	clothe	27.99	1	27.99
CHOST 7 W	004000	women	running	shoe	103.99	1	103.99

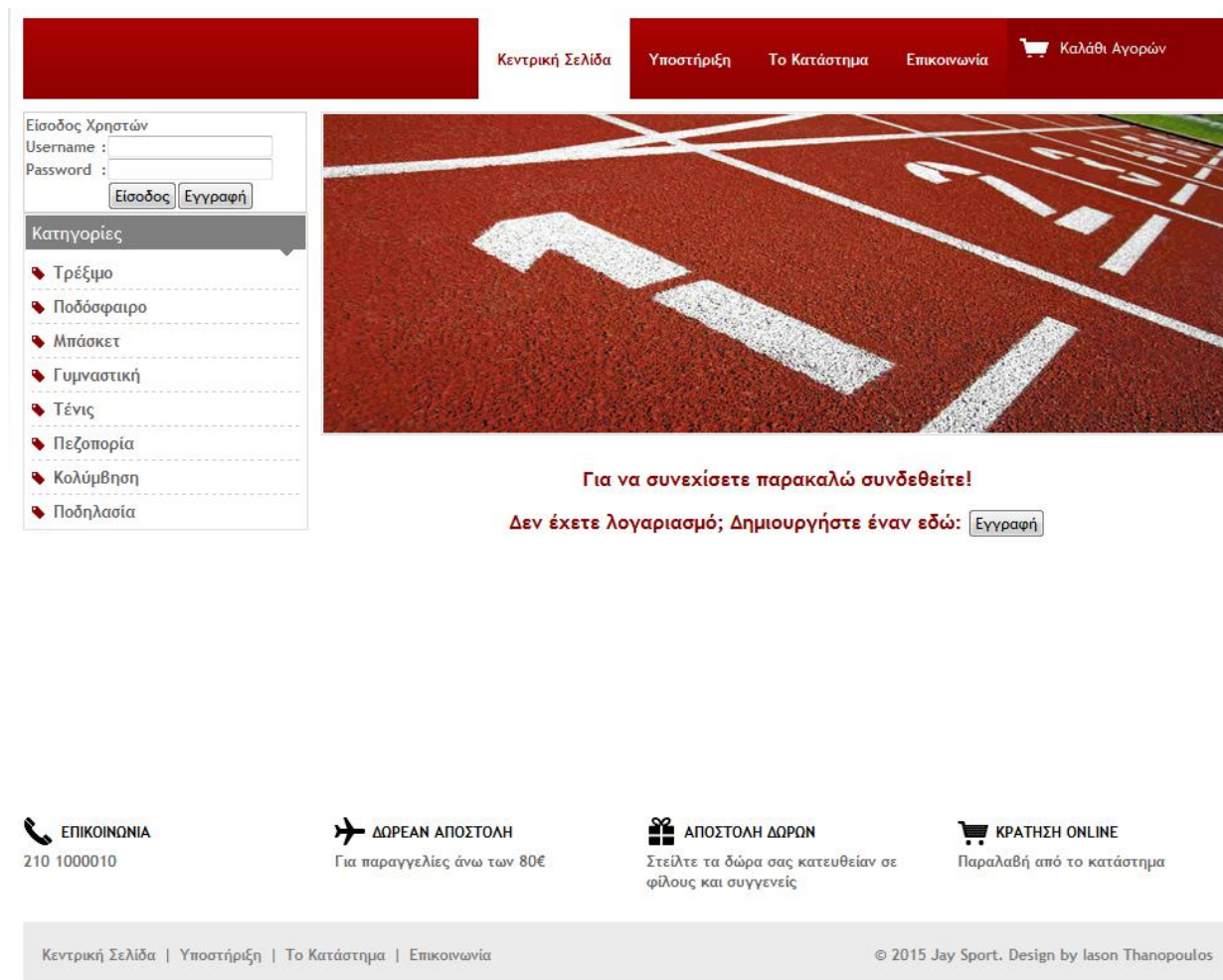
Εικόνα 5.7 Πληροφορίες του χρήστη test2

Κεφάλαιο 6

Ο Κώδικας του συστήματος

6.1 Βασική δομή

Ο παρακάτω κώδικας (index.php) αποτελεί τον "σκελετό" της εργασίας ο οποίος μαζί με την CSS βάζει τα σωστά κείμενα στην σωστή θέση τους και με τον κατάλληλο τρόπο. Κομμάτια αυτού του κώδικα θα βρίσκονται σε όλα τα σημεία της εφαρμογής. Τα κυριότερα στοιχεία του είναι τα μενού επιλογών και το σύστημα εισόδου.



Εικόνα 6.1 Η Κεντρική Σελίδα

Στο σημείο αυτό το σύστημα τσεκάρει εάν ο χρήστης είναι συνδεδεμένος στο σύστημα και αν είναι τον ανακατευθύνει στο σημείο που πρέπει να βρίσκεται ο χρήστης. Επειδή θέλουμε να

παρακολουθούμε τις κινήσεις των χρηστών μας, εάν ο χρήστης δεν είναι συνδεδεμένος δεν θα μπορεί να συνεχίσει την πλοήγησή του στην εφαρμογή. Για τον λόγο αυτόν υπάρχει ένας αντίστοιχος κώδικας σε πολλά σημεία της εφαρμογής που τσεκάρει το αντίστροφο, δηλαδή αν ο χρήστης δεν είναι συνδεδεμένος και τον ανακατευθύνει αναλόγως.

```
<?php
session_start();
if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
    header("location:main_login.php");
}
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Jay Sport</title>
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
</head>
<body>
<div class="shell">
    <div id="header">
        <h1 id="logo"><a href="index.php">Jay Sport</a></h1>
        <div id="cart"> <a href="cart/cart.php" class="cart-link">Καλάθι
Αγορών</a>
        <div class="cl">&nbsp;</div>
    </div>
```

Εδώ βρίσκεται το κεντρικό μενού επιλογών όπου ο χρήστης μπορεί να μάθει περισσότερα για το κατάστημα.

```
<div id="navigation">
    <ul>
        <li><a href="index.php" class="active">Κεντρική Σελίδα</a></li>
        <li><a href="other/support.php">Υποστήριξη</a></li>
        <li><a href="other/thestore.php">Το Κατάστημα</a></li>
        <li><a href="other/contact.php">Επικοινωνία</a></li>
    </ul>
</div>
<div id="main">
    <div class="cl">&nbsp;</div>
    <div id="content">
        <div id="storepic" class="box">
            
        </div>
        <br><b><center><font size="3" color="darkred">Για να συνεχίσετε
παρακαλώ συνδεθείτε!</font></center></b><br>
        <b><center><font size="3" color="darkred">Δεν έχετε λογαριασμό;
Δημιουργήστε έναν εδώ: <td><input type="button" value="Εγγραφή"
onClick="parent.location='register.php'"></font></td><b><center>
    </div>
    <div id="sidebar">
```


Παρακάτω βρίσκεται η φόρμα εισόδου στο σύστημα. Ο χρήστης εισάγει το αναγνωριστικό και τον κωδικό πρόσβασής του και πατώντας το κουμπί εισόδου το σύστημα τον ανακατευθύνει και ελέγχει τα στοιχεία του. Αυτά που μπορεί να συμβούν είναι δύο πράγματα. Πρώτον, ο χρήστης θα εισέλθει επιτυχώς στο σύστημα. Δεύτερον, το σύστημα θα τον ανακατευθύνει στην σελίδα σφάλματος για να ξαναπροσπαθήσει.

```

<div class="login">
  <table width="225" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
    <tr>
      <form name="form1" method="post" action="checklogin.php">
        <td>
          <table width="100%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
            <tr>
              <td colspan="3"><strong>Είσοδος Χρηστών </strong></td>
            </tr>
            <tr>
              <td width="78">Username</td>
              <td width="6">:</td>
              <td width="294"><input name="myusername" type="text"
id="myusername"></td>
            </tr>
            <tr>
              <td>Password</td>
              <td>:</td>
              <td><input name="mypassword" type="password" id="mypassword"></td>
            </tr>
            <tr>
              <td>&nbsp;</td>
              <td>&nbsp;</td>
              <td><input type="submit" name="Submit" value="Είσοδος"><input
type="button" value="Εγγραφή" onClick="parent.location='register.php'"></td>
            </tr>
          </table>
        </td>
      </form>
    </tr>
  </table>
</div>

```

Εδώ βρίσκεται το πλαινό μενού του συστήματος όπου ο χρήστης μπορεί να δει τις κατηγορίες των προϊόντων του καταστήματος και να διαλέξει όποιο επιθυμεί.

```

<div class="box categories">
  <h2>Κατηγορίες <span></span></h2>
  <div class="box-content">
    <ul>
      <li><a href="shop/running.php">Τρέξιμο</a></li>
      <li><a href="shop/football.php">Ποδόσφαιρο</a></li>
      <li><a href="shop/basketball.php">Μπάσκετ</a></li>
    </ul>
  </div>
</div>

```

```

        <li><a href="shop/fitness.php">Γυμναστική</a></li>
        <li><a href="shop/tenis.php">Τένις</a></li>
        <li><a href="shop/hiking.php">Πεζοπορία</a></li>
        <li><a href="shop/swimming.php">Κολύμβηση</a></li>
        <li class="last"><a href="shop/biking.php">Ποδηλασία</a></li>
    </ul>
</div>
</div>
</div>
<div class="cl">&nbsp;</div>
</div>
<div class="side-full">
    <div class="cols">
        <div class="cl">&nbsp;</div>
        <div class="col">
            <h3 class="ico ico2">Επικοινωνία</h3>
            <p>210 1000010</p>
        </div>
        <div class="col">
            <h3 class="ico ico1">Δωρεαν Αποστολη</h3>
            <p>Για παραγγελίες άνω των 80€</p>
        </div>
        <div class="col">
            <h3 class="ico ico3">Αποστολη Δωρων</h3>
            <p>Στείλτε τα δώρα σας κατευθείαν σε φίλους και συγγενείς</p>
        </div>
        <div class="col col-last">
            <h3 class="ico ico4">Κρατηση Online</h3>
            <p>Παραλαβή από το κατάστημα</p>
        </div>
        <div class="cl">&nbsp;</div>
    </div>
<div id="footer">
    <p class="left"> <a href="index.php">Κεντρική Σελίδα</a> <span>|</span>
<a href="other/support.php">Υποστήριξη</a> <span>|</span> <a
href="other/thestore.php">Το Κατάστημα</a> <span>|</span> <a
href="other/contact.php">Επικοινωνία</a> </p>
    <p class="right"> &copy; 2015 Jay Sport. Design by Iason Thanopoulos
</p>
</div>
</div>
</div>
</body>
</html>

```

6.2 Η CSS του συστήματος

Παρακάτω ακολουθεί ολόκληρος ο κώδικας με την CSS ο οποίος είναι υπεύθυνος για την στυλιστική εμφάνιση του συστήματος. Δίνει χρώματα

στα γράμματα, τα βάζει στο σωστό σημείο, φτιάχνει τα μενού επιλογών ομοιόμορφα κ.α.

```
* { padding:0; margin:0; outline:0; }
body {
    background:#fff url(images/body.gif) repeat-x 0 0;
    font-family: "Trebuchet MS", Arial, sans-serif;
    font-size:12px;
    line-height:16px;
    color:#636363;
    height:100%;
}
input, textarea, select { font-family: "Trebuchet MS", Arial, sans-serif; font-size:12px; }
.field { background:#ebebcb; border:solid 1px #dedede; padding:2px;}
a img { border:0; }
a { color:#8b0000; text-decoration: underline; cursor:pointer; }
a:hover { color:#666; text-decoration: none; }
.left, .alignleft { float:left; display:inline; }
.right, .alignright { float:right; display:inline; }
.cl { font-size:0; line-height:0; clear:both; display:block; height:0; }
.al { text-align: left; }
.ar { text-align: right; }
.ac { text-align: center; }
h2 { font-size:14px; line-height:16px; }
h1#logo { font-size:0; line-height:0; width:156px; height:64px; float:left; }
.shell { width:960px; margin:0 auto; padding:10px; background:#fff;}
#header { height:64px; background:url(images/header.gif); position:relative;}
#navigation { float:right; white-space:nowrap;}
#navigation ul { list-style-type: none; height:64px; font-weight: bold; float:left;}
#navigation ul li { float:left; display:inline; }
#navigation ul li a { float:left; height:64px; line-height:64px; text-decoration: none; color:#fff; padding:0 15px;}
#navigation ul li a.active,
#navigation ul li a:hover { background:#fff; color:#8b0000; }
```

```

#cart { float:right; width:160px; top:0; right:0; height:51px;
background:#8b0000; color:#fff; padding:13px 10px 0 10px; white-space:nowrap;
line-height:20px;}
a.cart-link { color:#fff; background:url(images/cart-link.gif) no-repeat 0 0;
padding:0 0 0 32px; text-decoration: none;}
a.cart-link:hover { text-decoration: underline;}

#sidebar { float:left; width:226px; }
#content { float:right; width:724px; }
#content h2 { font-size:14px; line-height:16px; position:relative; top:50px;
left:250px; color:darkred;}
#content h3 {font-size:14px; line-height:16px;}
#content h1 {margin-bottom:50px; margin-top:50px; color:darkred;}

#main { padding:10px 0 0 0; margin-bottom:200px; }

.box { padding:1px; border:solid 1px #dedede; margin-bottom:10px;}
.box h2 { background:#7f7f7f; color:#fff; font-weight: normal; padding:0 5px;
position:relative; height:27px; line-height:27px; }
.box h2 span { position:absolute; width:10px; height:5px;
background:url(images/h2-arr.gif); top:27px; right:10px; font-size:0; line-
height:0;}
.box-content { padding:5px;}
.box-content .active {color:darkred;}

.categories { min-height:100px; }
.categories ul { list-style-type: none; font-size:13px;}
.categories ul li { border-bottom:dashed 1px #ccc; padding:5px 0;}
.categories ul li.last { border-bottom:0; padding-bottom:0;}
.categories ul li a { color:#5f5f5f; text-decoration: none;
background:url(images/cat.gif) no-repeat 0 4px; padding-left:17px;}
.categories ul li a:hover { color:#8b0000; }

#storepic { height:252px; position:relative; overflow:hidden; }

.products {}
.products ul { list-style-type: none; }
.products ul li { position:relative; padding:1px; border:solid 1px #dedede;
float:left; width:226px; margin-right:9px; height:383px; overflow:hidden;
margin-bottom:10px;}
.products ul li.last { margin-right:0; }
.products h3 { font-size:12px; line-height:14px; text-transform: uppercase;
color:#000; }
.products h4 { font-size:12px; line-height:14px; text-transform: uppercase;
color:#000; }

.product-info { position:absolute; width:153px; top:194px; left:0;}

```

```

.product-info h3{ background:#8b0000; color:#fff; padding:6px 10px; }
.product-info h4{ font-weight: normal;}
.product-info p{ font-size:16px; line-height:18px; color:#000; padding:5px 0
7px 0; color:#5f5f5f; text-decoration: none; font-size:11px;}
.product-info .product-desc{ padding:10px; background:url(images/info.png);
width:133px;}
.product-info form{position:absolute; width:153px; left:100px; top:152px;}

.price { display:block; font-size:21px; color:#8b0000; line-height:23px; }

.more-products { border:solid 1px #dedede; position:relative; height:114px;
overflow:hidden; }
.more-products ul{ list-style-type: none; height:94px; position:relative;
overflow:hidden; width:805px;}
.more-products ul li{ float:left; width:94px; height:94px; border-right:dashed
1px #ccc; padding:0 10px;}
.more-products ul li.last{ border-right:0; }

.more-products-holder { width:804px; height:94px; position:relative;
overflow:hidden; top:10px; left:70px;}
.more-products-holder .jcarousel-clip{ width:804px; height:94px;
position:relative; overflow:hidden; }

.register {position:relative; height:100px; top:50px; left:250px;}

.side-full {position:absolute;}

.cols { padding:15px 0; }
.col { float:left; display:inline; width:217px; margin-right:30px;}
.col-last { margin-right:0;}

.cols h3 { font-size:12px; line-height:14px; text-transform: uppercase;
color:#000; }
.cols h4 { font-size:12px; line-height:14px; text-transform: uppercase;
color:#000; }

h3.ico { background-repeat:no-repeat; background-position:0 2px; padding:6px 0
8px 30px;}
h3.icol { background-image:url(images/icol.gif);}
h3.ico2 { background-image:url(images/ico2.gif);}
h3.ico3 { background-image:url(images/ico3.gif);}
h3.ico4 { background-image:url(images/ico4.gif);}

#footer { height:51px; background:#ebebeb; white-space:nowrap; line-
height:50px; padding:0 15px; color:#7b7b7b; margin-top:10px; }
#footer a{ color:#7b7b7b; text-decoration: none;}
#footer a:hover{ color:#000;}
#footer span{ padding:0 2px;}

```

```

.txt-heading{padding: 5px 10px;font-size:1.1em;font-weight:bold;color:#999;}

.btnRemoveAction{color:#D60202;border:0;padding:2px 10px;font-size:0.9em;}

#btnEmpty{background-color:#D60202;border:0;padding:1px 10px;color:#FFF;
font-size:0.8em;font-weight:normal;float:right;text-decoration:none;}

.btnAddAction{background-color:#79b946;border:0;padding:3px
10px;color:#FFF;margin-left:1px;}

#shopping-cart {border-top: #79b946 2px solid;margin-bottom:30px;}
#shopping-cart .txt-heading{background-color: #D3F5B8;}
#shopping-cart table{width:100%;background-color:#F0F0F0;}
#shopping-cart table td{background-color:#FFFFFF;}

.cart-item {border-bottom: #79b946 1px dotted;padding: 10px;}

#product-grid {border-top: #F08426 2px solid;margin-bottom:30px;}
#product-grid .txt-heading{background-color: #FFD0A6;}

.product-item { float:left; background:#F0F0F0; margin:15px; padding:5px;}
.product-item div{text-align:center; margin:10px;}
.product-price {color:#F08426;}
.product-image {height:100px;background-color:#FFF;}

```

6.3 Δημιουργία χρήστη

Πατώντας το κουμπί εγγραφής στο σύστημα ο χρήστης καλείται να συμπληρώσει μια φόρμα με δύο στοιχεία (αναγνωριστικό και κωδικός πρόσβασης) Αφού τα συμπληρώσει, επιβεβαιώνει την εγγραφή του και το σύστημα τον ανακατευθύνει αυτόματα.

```

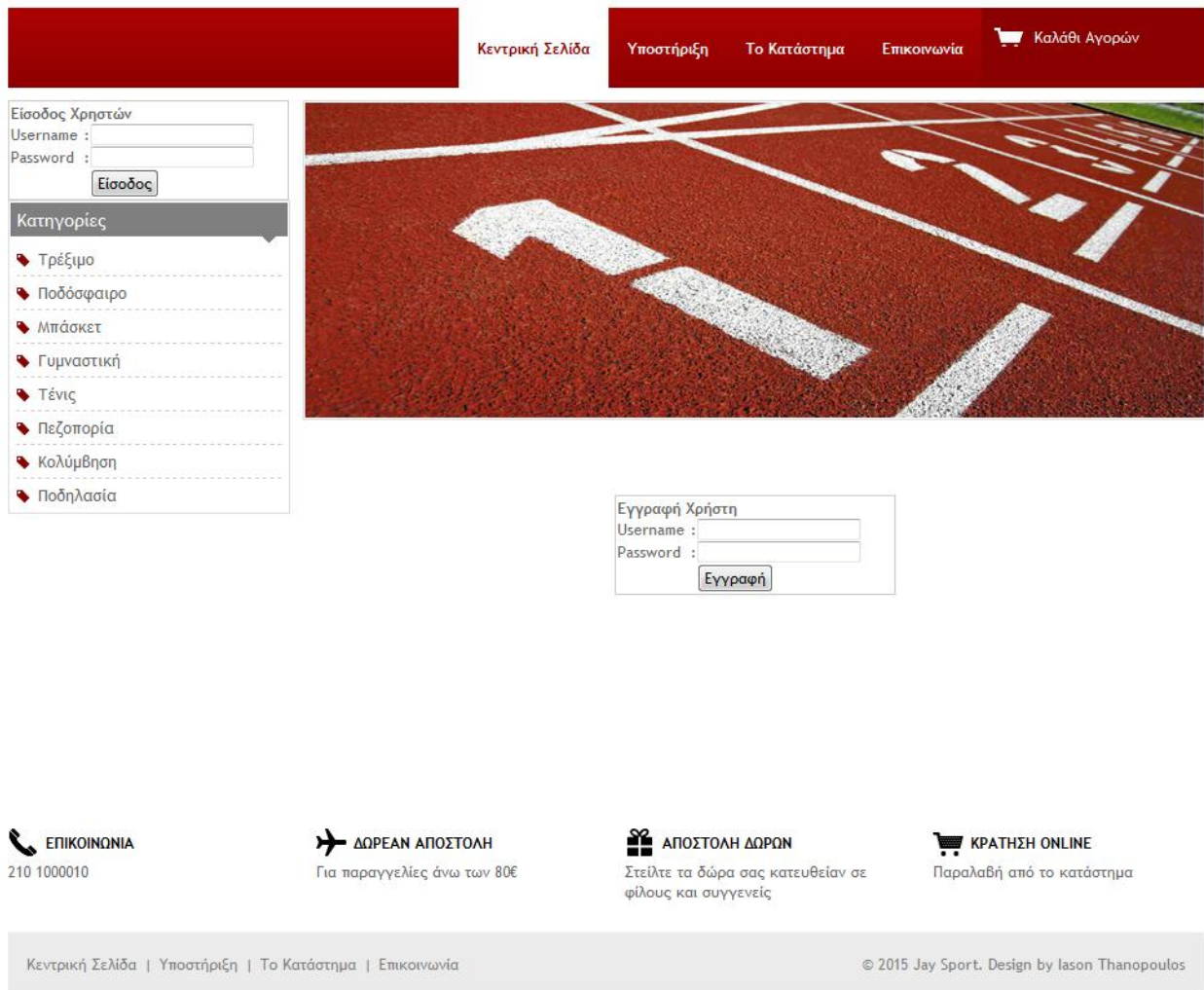
<div class="register">
  <table width="225" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
  <tr>
  <form name="form1" method="post" action="registeracc.php">
  <td>
  <table width="100%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
  <tr>
  <td colspan="3"><strong>Εγγραφή Χρήστη </strong></td>
  </tr>
  <tr>
  <td width="78">Username</td>
  <td width="6">:</td>

```

```

        <td width="294"><input name="myusername" type="text"
id="myusername"></td>
    </tr>
    <tr>
    <td>Password</td>
    <td>:</td>
    <td><input name="mypassword" type="password" id="mypassword"></td>
    </tr>
    <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td><input type="submit" name="Submit" value="Εγγραφή"></td>
    </tr>
</table>
</td>
</form>
</tr>
</table>
</div>

```



Εικόνα 6.2 Σελίδα Εγγραφής Χρήστη

Το επόμενο βήμα της εφαρμογής είναι να τσεκάρει εάν μπορεί να δημιουργήσει τον χρήστη ή εάν υπάρχει είδη στο σύστημα.

```
<?php
```

```
ob_start();
```

Εδώ το σύστημα προσπαθεί να δει εάν μπορεί να συνδεθεί στην βάση δεδομένων.

```
$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123", "u282837298_store") or die("cannot connect");
```

Υστερα το σύστημα δημιουργεί ένα ερώτημα στην βάση δεδομένων και της λέει εάν υπάρχει το αναγνωριστικό που εισήγαγε ο χρήστης ξανά στην βάση.

```
$myusername=$_POST['myusername'];  
$mypassword=$_POST['mypassword'];
```

```
$sql="SELECT * FROM members WHERE username='$myusername';"  
$result=mysqli_query($link, $sql);  
$count=mysqli_num_rows($result);
```

Τελικά μπορούν να συμβούν δύο πράγματα. Πρώτον, το αναγνωριστικό δεν βρέθηκε στην βάση δεδομένων επομένως το σύστημα το εισάγει στην βάση και ανακατευθύνει τον χρήστη στην κεντρική σελίδα. Δεύτερον, το αναγνωριστικό βρέθηκε ξανά στην βάση δεδομένων οπότε το σύστημα ανακατευθύνει τον χρήστη στην σελίδα όπου τον ενημερώνει αναλόγως.

```
if($count!=1){  
$query="INSERT INTO members (username, password) VALUES  
( '$_POST[myusername]', '$_POST[mypassword]' )";  
$query2="INSERT INTO tracking (accrated) VALUES (now())";  
mysqli_query($link, $query);  
mysqli_query($link, $query2);  
header("location:index.php");  
}  
else {  
header("location:falseregister.php");  
}  
ob_end_flush();  
?>
```

6.4 Είσοδος και έξοδος από το σύστημα

Αφού ο χρήστης πληκτρολογήσει τα στοιχεία πρόσβασης του στην φόρμα εισόδου το σύστημα θα προσπαθήσει να τον εισάγει στο σύστημα. Με τον ίδιο τρόπο όπως και στην δημιουργία χρήστη το σύστημα θα δημιουργήσει ένα ερώτημα στην βάση δεδομένων για το αν υπάρχει ο χρήστης είδη. Εάν υπάρχει και τα στοιχεία είναι σωστά τότε το σύστημα δημιουργεί μια σειρά από sessions τα οποία κρατάνε ανοιχτή την σύνδεση του χρήστη με τον διακομιστή, αλλιώς τον ανακατευθύνει στην σελίδα σφάλματος.

```
<?php
ob_start();

$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");

$username=$_POST['myusername'];
$password=$_POST['mypassword'];

$sql="SELECT * FROM members WHERE username='$username' and
password='$password'";
$result=mysqli_query($link, $sql);
$count=mysqli_num_rows($result);

if($count==1){
    session_start();
    $_SESSION['username']=$username;
    $_SESSION['password']=$password;
    $_SESSION['loggedin'] = true;
}
```

Ο παρακάτω SQL κώδικας σημειώνει στην βάση δεδομένων την τωρινή ημερομηνία και την αντικαθιστά με την προηγούμενη και αυξάνει της φορές που έχει συνδεθεί στο σύστημα ο χρήστης κατά ένα.

```
$query="UPDATE members m, tracking t SET t.lastlogin=now(),
t.timesloggedin=t.timesloggedin+1 WHERE m.username='$username' and
m.password='$password' and m.id=t.id";
mysqli_query($link, $query);
header("location:main_login.php");
}
else {
header("location:wronguser.php");
}

ob_end_flush();
?>
```

Ο χρήστης θα ξέρει άμεσα ότι έχει εισέλθει στο σύστημα καθώς στην θέση του πίνακας εισαγωγής των στοιχείων του θα βρίσκεται ένα μήνυμα που τον καλωσορίζει στο κατάστημα.

```

        <table width="225" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
        <?php
            echo "&nbsp;<br />";
            echo "Καλωσήρθες, " . $_SESSION['username'] . ". <a
href='../logout.php'><span>Εξοδος</span></a><br />";
            echo "&nbsp;<br />";
            echo "&nbsp;";
        ?>
    </table>

```

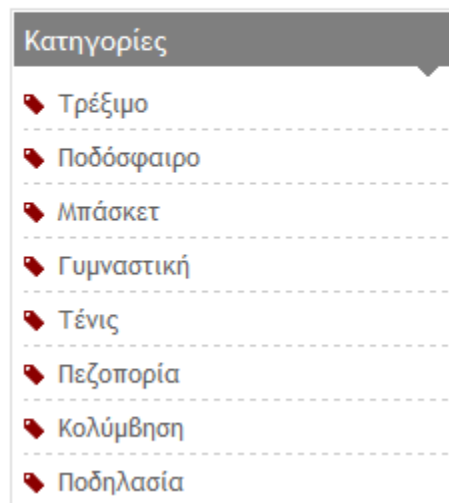
Τέλος, εάν ο χρήστης επιθυμεί να αποσυνδεθεί από το σύστημα, πατώντας τον υπερσύνδεσμο εξόδου από το σύστημα αυτό καταστρέφει όλα τα ανοιχτά sessions και τον ανακατευθύνει στην αρχική σελίδα.

```

<?php
session_start();
session_destroy();
header("location:index.php");
?>

```

Καλωσήρθες, test. [Εξοδος](#)



Εικόνα 6.4 Μήνυμα Καλωσορίσματος και Κουμπί Εξόδου

6.5 Τα προϊόντα

Η εφαρμογή διαθέτει ένα τεράστιο όγκο από προϊόντα και σκοπός της είναι να τα εμφανίζει ανάλογα με τις επιλογές του χρήστη. Εάν υποθέσουμε ότι ο χρήστης επέλεξε την κατηγορία "τρέξιμο" από το μενού επιλογών το σύστημα θα το παροτρύνει σε μια ακόμα επιλογή. Στην προκειμένη περίπτωση θα πρέπει να διαλέξει ανάμεσα σε ρούχα ή παπούτσια. Κάθε επιλογή είναι ένας υπερσύνδεσμος που τον οδηγεί στο επόμενο βήμα.

```
<div id="content">
  <div id="storepic" class="box">
    
  </div>
  <div class="products">
    <div class="cl">&nbsp;</div>
    <ul>
      <li> <a href="runningclothes.php"></a>
      <div class="product-info">
        <h3>Ρούχα</h3>
        <div class="product-desc">
          <p>Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν
εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά
σημαντική! <br />
          </div></div>
        </li>
      <li class="last"> <a href="runningshoes.php"></a>
      <div class="product-info">
        <h3>Παπούτσια</h3>
        <div class="product-desc">
          <p>Τα παπούτσια running είναι αναμφισβήτητα το σημαντικότερο
κομμάτι εξοπλισμού για έναν δρομέα.<br />
          </div></div>
        </li>
      </ul>
    <div class="cl">&nbsp;</div>
  </div>
</div>
```

Καλωσήρθες, test. [Εξοδος](#)

- Κατηγορίες
- 🔴 Τρέξιμο
 - 🔴 Ποδοσφαιρο
 - 🔴 Μπάσκετ
 - 🔴 Γυμναστική
 - 🔴 Τένις
 - 🔴 Πεζοπορία
 - 🔴 Κολύμβηση
 - 🔴 Ποδηλασία



ΡΟΥΧΑ

Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά σημαντική!

ΠΑΠΟΥΤΣΙΑ

Τα παπούτσια running είναι αναμφισβήτητα το σημαντικότερο κομμάτι εξοπλισμού για έναν δρομέα.

Εικόνα 6.5.1 Η κατηγορία "τρέξιμο"

Αν υποθέσουμε πάλι, ότι ο χρήστης διάλεξε την υποκατηγορία με τα ρούχα το σύστημα θα βάλει τον χρήστη να κάνει μια τελευταία επιλογή. Σε αυτήν την περίπτωση θα διαλέξει ανάμεσα σε ανδρικά, γυναικεία ή παιδικά ρούχα.

```

<div class="products">
  <div class="cl">&nbsp;</div>
  <ul>
    <li> <a href="products/runningclothesmen.php"></a>
      <div class="product-info">
        <h3>Ανδρικά</h3>
        <div class="product-desc">
          <p>Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν
εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά
σημαντική! <br />
          </p></div>
        </div></li>
    <li> <a href="products/runningclotheswomen.php"></a>
      <div class="product-info">
        <h3>Γυναικεία</h3>

```

```

<div class="product-desc">
  <p>Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν
εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά
σημαντική! <br />
</div></div>
</li>
<li class="last"> <a href="products/runningclotheskids.php"></a>
<div class="product-info">
  <h3>Παιδικα</h3>
  <div class="product-desc">
    <p>Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν
εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά
σημαντική! <br />
  </div></div>
</li>
</ul>
<div class="cl">&nbsp;</div>
</div>

```

Κεντρική Σελίδα Υποστήριξη Το Κατάστημα Επικοινωνία Καλάθι Αγορών

Καλωσήρθες, test. [Εξοδος](#)

Κατηγορίες

- Τρέξιμο
- Ποδόσφαιρο
- Μπάσκετ
- Γυμναστική
- Τένις
- Πεζοπορία
- Κολύμβηση
- Ποδηλασία

ΑΝΔΡΙΚΑ

Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά σημαντική!

ΓΥΝΑΙΚΕΙΑ

Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά σημαντική!

ΠΑΙΔΙΚΑ

Η επιλογή των κατάλληλων ρούχων, τα οποία διαθέτουν εξειδικευμένες τεχνολογίες που διευκολύνουν τον δρομέα, είναι εξαιρετικά σημαντική!

Εικόνα 6.5.2 Ρούχα από την κατηγορία "τρέξιμο"

Στην τελευταία επιλογή υποθέτουμε πάλι ότι ο χρήστης διάλεξε την κατηγορία με τα ανδρικά ρούχα. Το σύστημα είναι επιτέλους ικανό να δημιουργήσει ένα ερώτημα στην βάση δεδομένων για να του εμφανίσει τα κατάλληλα προϊόντα. Από εκεί και πέρα ο χρήστης μπορεί να δει την λίστα με τα προϊόντα και να διαλέξει αυτό που θέλει να αγοράσει. Εφόσον κάνει την επιλογή του, το σύστημα στέλνει το αντικείμενο στο καλάθι αγορών.


```
<div class="products">
  <?php
    $product_array = $db_handle->runQuery("SELECT * FROM products WHERE
category='running' AND type='clothe' AND sex='men'");
    if (!empty($product_array)) {
      foreach($product_array as $key=>$value){
        ?>
        <ul>
          <li> <a href="#">" alt="" /></a>
            <div class="product-info">
              <h3><?php echo $product_array[$key]["name"]; ?></h3>
              <div class="product-desc">
                <h4><?php echo $product_array[$key]["code"]; ?></h4>
                <p><?php echo $product_array[$key]["description"]; ?><br />
                <strong class="price"><?php echo
$product_array[$key]["price"]; ?> €</strong> </div>
                <form method="post"
action=".../.../cart/cart.php?action=add&code=<?php echo
$product_array[$key]["code"]; ?>">
                  <div><input type="text" name="quantity" value="1" size="2"
/><input type="submit" value="Add to cart" class="btnAddAction" /></div>
                </form>
              </div>
            </li>
          </ul>
        <?php
        }
      }
    }
  </div>
```



Κεντρική Σελίδα Υποστήριξη Το Κατάστημα Επικοινωνία Καλάθι Αγορών

Καλωσήρθες, test. Έξοδος

Κατηγορίες

- Τρέξιμο
- Ποδόσφαιρο
- Μπάσκετ
- Γυμναστική
- Τένις
- Πεζοπορία
- Κολύμβηση
- Ποδηλασία






4\"

000001

Ανδρικό σορτσάκι running με τεχνολογία Dri-FIT.

23.99 €

1 Add to cart




5\"

000002

Ανδρικό σορτσάκι running με τεχνολογία Dri-FIT.

35.99 €

1 Add to cart



EDDY UX RUNNING PRO H/Z

000003

Ανδρικό t-shirt running της PRO TOUCH.

19.99 €

1 Add to cart

Εικόνα 6.5.3 Ανδρικά Ρούχα από την κατηγορία "τρέξιμο"

Τέλος το σύστημα ειδοποιεί την βάση δεδομένων ότι ο χρήστης βρίσκεται στην κατηγορία με τα ανδρικά προϊόντα τρεξίματος και ανανεώνει το προφίλ του χρήστη του με νέα στοιχεία.

```
$tmysql = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
$query="UPDATE members m, tracking t SET t.man=t.man+1,
t.running=t.running+1 WHERE m.username='".$SESSION['username']."' and
m.password='".$SESSION['password']."' and m.id=t.id";
mysqli_query($tmysql, $query);
```

6.6 Το καλάθι αγορών

Για την εξυπηρέτηση των χρηστών της εφαρμογής το σύστημα διαθέτει ένα πλήρες λειτουργικό καλάθι αγορών. Αυτό σημαίνει ότι τα

αντικείμενα που επιθυμεί να αγοράσει ο χρήστης κατευθύνονται στο καλάθι μέχρι ο χρήστης να συμπληρώσει την φόρμα αγορών.

Ο κώδικας του καλαθιού αποτελείται από τρεις "switch cases" και κάθε μια αντιστοιχεί σε μια ενέργεια που πραγματοποιεί το καλάθι. Η πρώτη ενέργεια δίνει την δυνατότητα στο καλάθι να φτιάχνει τις λίστες με τα προϊόντα εφόσον το καλάθι δεν είναι άδειο. Η δεύτερη ενέργεια δίνει στον χρήστη την δυνατότητα να αφαιρέσει μεμονωμένα αντικείμενα από το καλάθι αφαιρώντας το συγκεκριμένο session. Τέλος, η τρίτη επιλογή δίνει στον χρήστη την δυνατότητα να αφαιρέσει όλα τα αντικείμενα από το καλάθι καταστρέφοντας όλα τα ενεργά sessions για το καλάθι.

```
<?php
session_start();
if (!isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == false) {
    header("location:../index.php");
}
require_once("dbcontroller.php");
$db_handle = new DBController();
if(!empty($_GET["action"])) {
switch($_GET["action"]) {
    case "add":
        if(!empty($_POST["quantity"])) {
            $productByCode = $db_handle->runQuery("SELECT * FROM products
WHERE code='" . $_GET["code"] . "'");
            $itemArray =
array($_productByCode[0]["code"]=>array('name'=>$productByCode[0]["name"],
'code'=>$productByCode[0]["code"], 'quantity'=>$_POST["quantity"],
'price'=>$productByCode[0]["price"]));

            if(!empty($_SESSION["cart_item"])) {

if(in_array($_productByCode[0]["code"],$_SESSION["cart_item"])) {
                foreach($_SESSION["cart_item"] as $k => $v) {
                    if($_productByCode[0]["code"] == $k)
                        $_SESSION["cart_item"][$k]["quantity"] =
$_POST["quantity"];
                }
            } else {
                $_SESSION["cart_item"] =
array_merge($_SESSION["cart_item"],$itemArray);
            }
            } else {
                $_SESSION["cart_item"] = $itemArray;
            }
        }
        break;
    case "remove":
        if(!empty($_SESSION["cart_item"])) {
```



```

        foreach($_SESSION["cart_item"] as $k => $v) {
            if($_GET["code"] == $k)
                unset($_SESSION["cart_item"][$k]);
            if(empty($_SESSION["cart_item"]))
                unset($_SESSION["cart_item"]);
        }
    }
    break;
    case "empty":
        unset($_SESSION["cart_item"]);
    break;
}
}
?>

```

Για να εμφανιστούν τα στοιχεία του πίνακα του καλαθιού, το σύστημα διαθέτει έναν πίνακα για τον σκοπό αυτό.

```

<?php
if(isset($_SESSION["cart_item"])){
    $item_total = 0;
?>
<table cellpadding="10" cellspacing="1">
<tbody>
<tr>
<th><strong>Όνομα Προϊόντος</strong></th>
<th><strong>Κωδικός</strong></th>
<th><strong>Ποσότητα</strong></th>
<th><strong>Τιμή</strong></th>
<th><strong>Αφαίρεση</strong></th>
</tr>
<?php
    foreach ($_SESSION["cart_item"] as $item){
        ?>
            <tr>
                <td><strong><?php echo $item["name"]; ?></strong></td>
                <td><?php echo $item["code"]; ?></td>
                <td><?php echo $item["quantity"]; ?></td>
                <td align="right"><?php echo "€".$item["price"]; ?></td>
                <td><a href="cart.php?action=remove&code=<?php echo
$item["code"]; ?>" class="btnRemoveAction">Αφαίρεση Αντικειμένου</a></td>
            </tr>
            <?php
                $item_total += ($item["price"]*$item["quantity"]);
            }
        ?>
    <tr>
        <td colspan="5" align="right"><strong>Σύνολο:</strong> <?php echo
"€".$item_total; ?></td>
    </tr>
</tbody>
</table>
<?php
}

```

```
?>
```

Το τελικό στάδιο της αγοράς από τον χρήστη είναι η συμπλήρωση της φόρμας αγορών. Η φόρμα αυτή είναι ορατή μόνο αν υπάρχει κάποιο αντικείμενο στο καλάθι και αφού συμπληρωθεί σωστά ανακατευθύνει τον χρήστη στην αρχική σελίδα και η αγορά έχει ολοκληρωθεί.

```
<?php
if(!empty($_SESSION["cart_item"]))
{ ?>
<form name="checker" action="buy.php" method="post" onSubmit="return
checkform(this)">
  <h3>Όνοματεπώνυμο: <input type="text" name="name" size="45"
required><br><br>
  Πόλη: <input type="text" name="city" size="45" required><br><br>
  Διεύθυνση <input type="text" name="adress" size="45" required><br><br>
  ΤΚ <input type="text" name="tk" size="45" pattern="[0-9]{5}"
required><br><br>
  e-mail <input type="text" name="mail" size="45" pattern="[a-z0-9._%+-
]+@[a-z0-9.-]+\.[a-z]{2,4}$" required><br><br>
  Τηλέφωνο <input type="text" name="phone" size="45" pattern="[0-9]{10}"
required><br><br></h3>
  <input type='hidden' name='cost' value='<?php echo $item_total?>' />
  <input type="submit" value="Ολοκλήρωση Παραγγελίας">
</form>
<?php } ?>
```

Επιπλέον, ο κώδικας περιέχει ένα μικρό κομμάτι Javascript ο οποίος ελέγχει εάν ο χρήστης έχει συμπληρώσει όλα τα στοιχεία του στην φόρμα και αν είναι σωστά. Για παράδειγμα, το σύστημα δεν θα επιτρέψει την συνέχιση της παραγγελίας εάν το e-mail δεν είναι της μορφής test@test.com ή το τηλέφωνο δεν είναι δεκαψήφιο.


```
<script type="text/javascript">
function checkform(form) {
  var inputs = form.getElementsByTagName('input');
  for (var i = 0; i < inputs.length; i++) {
    if(inputs[i].hasAttribute("required")){
      if(inputs[i].value == ""){
        alert("Please fill all required fields");
        return false;
      }
    }
  }
}
</script><br>
```

[Κεντρική Σελίδα](#)
[Υποστήριξη](#)
[Το Κατάστημα](#)
[Επικοινωνία](#)
[🛒 Καλάθι Αγορών](#)

Καλωσήρθες, test. [Έξοδος](#)

Κατηγορίες

- 🔴 Τρέξιμο
- 🔴 Ποδοσφαιρο
- 🔴 Μπάσκετ
- 🔴 Γυμναστική
- 🔴 Τένις
- 🔴 Πεζοπορία
- 🔴 Κολύμβηση
- 🔴 Ποδηλασία



Καλάθι Αγορών Αδειασμα

Όνομα Προϊόντος	Κωδικός	Ποσότητα	Τιμή	Αφαίρεση
4" RACER SHORT	000001	1	€23.99	Αφαίρεση Αντικειμένου
				Σύνολο: €23.99

Όνοματεπώνυμο:

Πόλη:

Διεύθυνση:

ΤΚ:

e-mail:

Τηλέφωνο:

Ολοκλήρωση Παραγγελίας

Εικόνα 6.6 Το καλάθι αγορών

Κατά την διάρκεια της ανακατεύθυνσης η εφαρμογή εκτελεί μια σειρά από εντολές οι οποίες ολοκληρώνουν την παραγγελία και ανανεώνουν το προφίλ του χρήστη.

```
<?php
session_start();
ob_start();
```

```
$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");
```

Εδώ το σύστημα στέλνει μια εντολή στην βάση δεδομένων ώστε να αλλάξει μερικά στοιχεία στο προφίλ του χρήστη. Αρχικά αυξάνει τον αριθμό των αγορών που έχει κάνει ο χρήστης κατά ένα, ύστερα ανανεώνει το συνολικό ποσό χρημάτων που έχει ξοδέψει σε αγορές και τέλος σημειώνει μερικά προσωπικά στοιχεία για τον χρήστη σύμφωνα με αυτά που συμπλήρωσε στην φόρμα παραγγελιών.

```
$query="UPDATE members m, tracking t SET t.timesbought=t.timesbought+1,
t.moneyspent=t.moneyspent+'$_POST[cost]', t.city='$_POST[city]',
```

```
t.adress='$_POST[address]', t.mail='$_POST[mail]', t.phone='$_POST[phone]'
WHERE m.username="' .$_SESSION['username'] .'" AND
m.password="' .$_SESSION['password'] .'" AND m.id=t.id";
```

```
mysqli_query($link, $query);
```

```
$membername = $_SESSION['username'];
```

Το επόμενο στάδιο είναι να στείλει σε έναν ξεχωριστό πίνακα, που συνδέεται με αυτόν του προφίλ του χρήστη, την λίστα με τα αντικείμενα που αγόρασε.

```
if(!empty($_SESSION["cart_item"])) {
    foreach($_SESSION["cart_item"] as $k => $v) {
        $quan = $_SESSION["cart_item"][$k]["quantity"];
        $query2 = "INSERT INTO itemsbought (item_code, item_member,
item_quantity) VALUES ('$k', '$membername', '$quan')";
        mysqli_query($link, $query2);
    }
}
```

Τέλος, καταστρέφει τα sessions του καλαθιού και τον ανακατευθύνει στην επόμενη σελίδα της εφαρμογής.

```
unset($_SESSION["cart_item"]);
```

```
header("location:../thankyou.php");
?>
```

6.7 Ο διαχειριστής

Όπως αναφέραμε σε προηγούμενο κεφάλαιο, το σύστημα ελέγχει αν ο χρήστης που εισέρχεται στο σύστημα είναι ο διαχειριστής της ιστοσελίδας. Στην περίπτωση που είναι τότε το σύστημα του εμφανίζει ένα επιπλέον μενού επιλογών με τρία στοιχεία (προσθήκη αντικειμένου, στατιστικά χρηστών, στατιστικά ιστοσελίδας) στην οποία έχει πρόσβαση μόνο αυτός.

```
<?php
if ($_SESSION['username'] == "admin")
```

```

{ ?>
<div class="box categories">
  <h2>Διαχείριση Ιστοσελίδας <span></span></h2>
  <div class="box-content">
    <ul>
      <li><a href="admin/additem.php">Προσθήκη Αντικειμένου</a></li>
      <li><a href="admin/userstats.php">Στατιστικά Χρηστών</a></li>
      <li class="last"><a href="admin/sitestats.php">Στατιστικά
Ιστοσελίδας</a></li>
    </ul>
  </div>
</div>
<?php } ?>

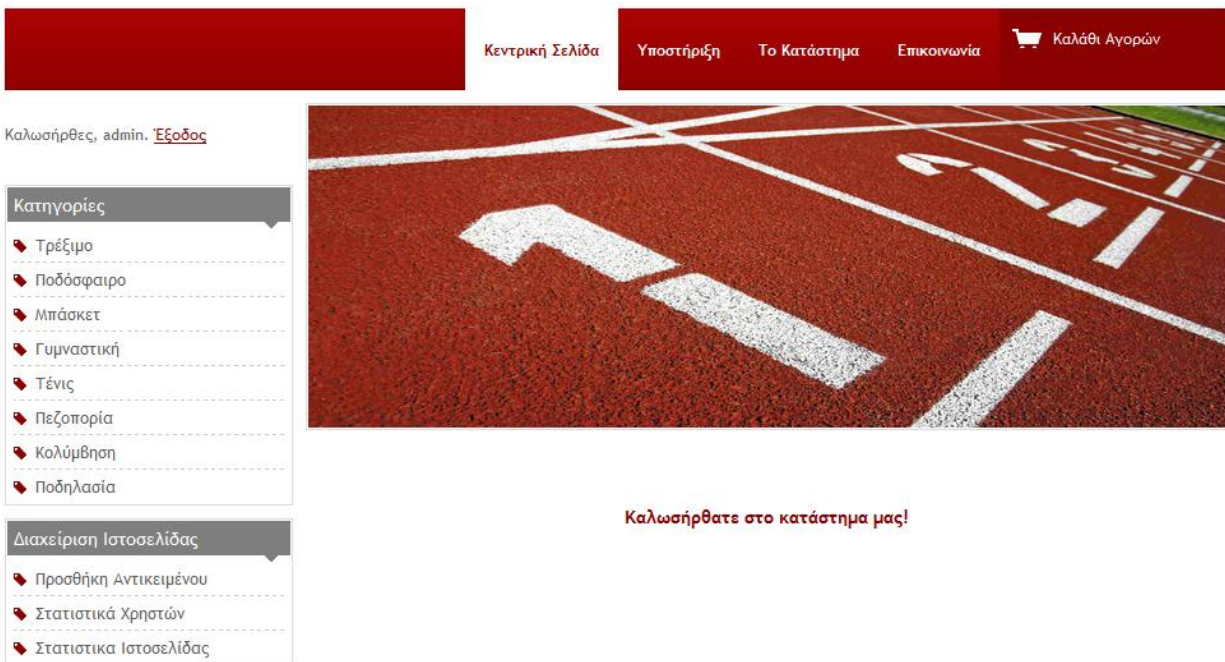
```

Για να σιγουρευτούμε ότι πρόσβαση έχει μόνο ο διαχειριστής σε αυτό το κομμάτι της εφαρμογής, το σύστημα ανακατευθύνει τους απλούς χρήστες στην αρχική σελίδα.

```

<?php
session_start();
if (!isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == false) {
    header("location:../index.php");
}
if ($_SESSION['username'] != "admin") {
    header("location:../index.php");
}
?>

```



Εικόνα 6.7 Η κεντρική σελίδα του διαχειριστή

6.7.1 Προσθήκη αντικειμένων

Το σύστημα δίνει την δυνατότητα στον διαχειριστή της ιστοσελίδας να προσθέσει καινούρια προϊόντα στην βάση δεδομένων. Αυτό γίνεται μέσω της συμπλήρωσης μιας φόρμας με συγκεκριμένο τρόπο. Αναλυτικές οδηγίες υπάρχουν για κάθε πεδίο.

```
<form action="addtophp.php" method="post">
  <h3>Παρακαλώ εισάγετε το όνομα του προϊόντος το οποίο
<u>πρέπει</u> να είναι μοναδικό:<br><br>
  Όνομα: <input type="text" name="name" size="45"><br><br>
  Παρακαλώ εισάγετε τον κωδικό του προϊόντος:<br><br>
  Κωδικός: <input type="text" name="code" size="45"><br><br>
  Παρακαλώ εισάγετε την τιμή του προϊόντος με την μορφή
<u>20.20</u>:<br><br>
  Τιμή: <input type="text" name="price"><br><br>
  Παρακαλώ εισάγετε την τοποθεσία της εικόνας του προϊόντος με
την μορφή <u>product-images/</u>όνομα εικονας<u>.jpg</u>:<br><br>
  Εικόνα: <input type="text" name="image"><br><br>
  Παρακαλώ εισάγετε μια περιγραφή για το προϊόν:<br><br>
  Πληροφορίες: <input type="text" name="info"
size="100"><br><br>
  Παρακαλώ διαλέξτε την κατηγορία του προϊόντος:
  <select name="category">
  <option value="running">Τρέξιμο</option>
  <option value="football">Ποδόσφαιρο</option>
  <option value="basketball">Μπάσκετ</option>
  <option value="fitness">Γυμναστική</option>
  <option value="tenis">Τένις</option>
  <option value="hiking">Πεζοπορία</option>
  <option value="swimming">Κολύμβηση</option>
  <option value="biking">Ποδηλασία</option>
  </select><br><br>
  Παρακαλώ διαλέξτε την υποκατηγορία του προϊόντος:
  <select name="type">
  <option value="clothe">Ρούχο</option>
  <option value="shoe">Παπούτσι</option>
  <option value="ball">Μπάλα</option>
  <option value="accessory">Αξεσουάρ</option>
  <option value="organ">Όργανο Γυμναστικής</option>
  <option value="racquet">Ρακέτα</option>
  <option value="backpack">Σακίδιο Πλάτης</option>
  <option value="tent">Σκηνή</option>
  <option value="swimsuit">Μαγιό</option>
  <option value="helmet">Κράνος</option>
  <option value="bike">Ποδήλατο</option>
  </select><br><br>
  Αν πρόκειται για ρούχο ή παπούτσι διαλέχτε <u>Ανδρας,Γυναίκα ή
Παιδί</u> αλλιώς <u>Ουδέτερο</u>:
  <select name="sex">
  <option value="men">Ανδρας</option>
  <option value="women">Γυναίκα</option>
```

```
<option value="children">Παιδί</option>
<option value="unisex">Ουδέτερο</option>
</select><br><br>
<input type="submit" value="Εισαγωγή στην Βάση">
<input type="reset" value="Καθαρισμός" /></h3>
</form>
```

Εφόσον η φόρμα συμπληρωθεί και αποσταλεί, το σύστημα ενημερώνει την βάση δεδομένων και προσθέτει το νέο προϊόν.

```
<?php
ob_start();

$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");

$sql="INSERT INTO products (name, code, price, image, description, category,
type, sex) VALUES
('$__POST[name]', '$__POST[code]', '$__POST[price]', '$__POST[image]', '$__POST[info]'
, '$__POST[category]', '$__POST[type]', '$__POST[sex]')";









mysqli_query($link,$sql);

header("location:additem.php");




ob_end_flush();
?>
```


Καλωσήρθες, admin. [Έξοδος](#)

Κατηγορίες

-  Τρέξιμο
-  Ποδοσφαίρο
-  Μπάσκετ
-  Γυμναστική
-  Τένις
-  Πεζοπορία
-  Κολύμβηση
-  Ποδηλασία

Διαχείριση Ιστοσελίδας

-  Προσθήκη Αντικειμένων
-  Στατιστικά Χρηστών
-  Στατιστικά Ιστοσελίδας



Παρακαλώ εισάγετε το όνομα του προϊόντος το οποίο πρέπει να είναι μοναδικό:

Όνομα:

Παρακαλώ εισάγετε τον κωδικό του προϊόντος:

Κωδικός:

Παρακαλώ εισάγετε την τιμή του προϊόντος με την μορφή 20.20:

Τιμή:

Παρακαλώ εισάγετε την τοποθεσία της εικόνας του προϊόντος με την μορφή product-images/όνομα εικονας.jpg:

Εικόνα:

Παρακαλώ εισάγετε μια περιγραφή για το προϊόν:

Πληροφορίες:

Παρακαλώ διαλέξτε την κατηγορία του προϊόντος: Τρέξιμο

Παρακαλώ διαλέξτε την υποκατηγορία του προϊόντος: Ρούχο

Αν πρόκειται για ρούχο ή παπούτσι διαλέχτε Άνδρας, Γυναίκα ή Παιδί αλλιώς Ουδέτερο: Άνδρας

Εικόνα 6.7.1 Η σελίδα προσθήκης αντικειμένων

6.7.2 Στατιστικά Χρηστών

Το κυριότερο κομμάτι της εφαρμογής είναι ότι ο διαχειριστής έχει την δυνατότητα να αναζητήσει και να δει όλες τις πληροφορίες που αντιπροσωπεύουν κάθε χρήστη ξεχωριστά. Αρχικά του δίνονται δύο επιλογές. Η πρώτη είναι ένα πεδίο αναζήτησης χρηστών και η δεύτερη είναι ένα κουμπί που του δίνει μια λίστα με όλους τους χρήστες του συστήματος.

```
<div class="add-content">
```



```

        <form action="search.php" method="post">
            <p>Αναζήτηση Χρήστη: <input name="user" type="text" id="user"
/><br /><br />
                <input type="submit" name="submit" value="Αναζήτηση" /></p><p>
</form><br /><br /> <br /><br />
        <form action="list.php" method="post">
            <p>Πλήρης λίστα με χρήστες: <input type="submit" name="submit"
value="Εμφάνιση" /></p><p>
</form>
</div>

```

Εάν ο χρήστης επιλέξει να δει την λίστα με όλους τους χρήστες τότε το σύστημα στέλνει μήνυμα στην βάση δεδομένων για να τους εμφανίσει.

```

<?php
$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");

$list=mysqli_query($link,"SELECT username FROM members");

$storeArray = Array();
while ($row = mysqli_fetch_array($list)) {
    $storeArray[] = $row['username'];
    echo $row['username'] . "<br/>";
}
?>

```

Στην περίπτωση που ο διαχειριστής αποφασίσει να αναζητήσει το αναγνωριστικό κάποιου χρήστη τότε το σύστημα στέλνει μήνυμα στην βάση δεδομένων ώστε να του φέρει όλα τα στοιχεία του προφίλ του. Αυτή η διαδικασία αποτελεί μια σειρά από ερωτήματα (queries) στην βάση και μαθηματικές πράξεις ώστε τα στοιχεία να είναι κατανοητά στον διαχειριστή.

```

$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");

$search=$_POST['user'];

$sql="SELECT * FROM members WHERE username='$search'";
$result=mysqli_query($link, $sql);
$count=mysqli_num_rows($result);
$resultman=mysqli_query($link,"SELECT ((t.man*100)/(t.man+t.woman)) AS Score
FROM tracking t, members m WHERE m.username='" . $search . "' AND
m.id=t.id");
$resultman1=mysqli_fetch_assoc($resultman);
$resultwoman=mysqli_query($link,"SELECT ((t.woman*100)/(t.man+t.woman)) AS
Score FROM tracking t, members m WHERE m.username='" . $search . "' AND
m.id=t.id");
$resultwoman1=mysqli_fetch_assoc($resultwoman);
$resultchild=mysqli_query($link,"SELECT child FROM tracking t, members m
WHERE m.username='" . $search . "' AND m.id=t.id");

```

```

$resultchild1=mysqli_fetch_assoc($resultchild);
$resultrunning=mysqli_query($link,"SELECT
((t.running*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS running FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultfootball=mysqli_query($link,"SELECT
((t.football*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS football FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultbasketball=mysqli_query($link,"SELECT
((t.basketball*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS basketball FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultfitness=mysqli_query($link,"SELECT
((t.fitness*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS fitness FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultttenis=mysqli_query($link,"SELECT
((t.tenis*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS tenis FROM tracking t, members m WHERE m.username='".$
. $search . "' AND m.id=t.id");
$resulthiking=mysqli_query($link,"SELECT
((t.hiking*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS hiking FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultswimming=mysqli_query($link,"SELECT
((t.swimming*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS swimming FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultbiking=mysqli_query($link,"SELECT
((t.biking*100)/(t.running+t.football+t.basketball+t.fitness+t.tenis+t.hiking+t.swimming+t.biking)) AS biking FROM tracking t, members m WHERE
m.username='".$ $search . "' AND m.id=t.id");
$resultrunning1=mysqli_fetch_assoc($resultrunning);
$resultfootball1=mysqli_fetch_assoc($resultfootball);
$resultbasketball1=mysqli_fetch_assoc($resultbasketball);
$resultfitness1=mysqli_fetch_assoc($resultfitness);
$resultttenis1=mysqli_fetch_assoc($resultttenis);
$resulthiking1=mysqli_fetch_assoc($resulthiking);
$resultswimming1=mysqli_fetch_assoc($resultswimming);
$resultbiking1=mysqli_fetch_assoc($resultbiking);
$resultacccreated=mysqli_query($link,"SELECT accreated FROM tracking t,
members m WHERE m.username='".$ $search . "' AND m.id=t.id");
$resultacccreated1=mysqli_fetch_assoc($resultacccreated);
$resultlastlogin=mysqli_query($link,"SELECT lastlogin FROM tracking t,
members m WHERE m.username='".$ $search . "' AND m.id=t.id");
$resultlastlogin1=mysqli_fetch_assoc($resultlastlogin);
$resulttimesloggedin=mysqli_query($link,"SELECT timesloggedin FROM tracking
t, members m WHERE m.username='".$ $search . "' AND m.id=t.id");
$resulttimesloggedin1=mysqli_fetch_assoc($resulttimesloggedin);
$resulttimesbought=mysqli_query($link,"SELECT timesbought FROM tracking t,
members m WHERE m.username='".$ $search . "' AND m.id=t.id");
$resulttimesbought1=mysqli_fetch_assoc($resulttimesbought);
$resultmoneyspent=mysqli_query($link,"SELECT moneyspent FROM tracking t,
members m WHERE m.username='".$ $search . "' AND m.id=t.id");
$resultmoneyspent1=mysqli_fetch_assoc($resultmoneyspent);

```

```

$resultcity=mysqli_query($link,"SELECT city FROM tracking t, members m WHERE
m.username='".$ . $search . "' AND m.id=t.id");
$resultcity1=mysqli_fetch_assoc($resultcity);
$resultaddress=mysqli_query($link,"SELECT adress FROM tracking t, members m
WHERE m.username='".$ . $search . "' AND m.id=t.id");
$resultadress1=mysqli_fetch_assoc($resultaddress);
$resultmail=mysqli_query($link,"SELECT mail FROM tracking t, members m WHERE
m.username='".$ . $search . "' AND m.id=t.id");
$resultmaill=mysqli_fetch_assoc($resultmail);
$resultphone=mysqli_query($link,"SELECT phone FROM tracking t, members m
WHERE m.username='".$ . $search . "' AND m.id=t.id");
$resultphonel=mysqli_fetch_assoc($resultphone);

```

Τέλος εμφανίζει όλα αυτά τα δεδομένα ή στέλνει το μήνυμα λάθους εάν ο χρήστης δεν υπάρχει.

```

<?php
    if($scount==1){
        echo "<b>Όνομα Χρήστη:</b> " . $_POST['user'] .
" <br/><br/>";
        echo "<b>Ο χρήστης είναι άντρας: </b>" .
$resultman1['Score'] . " %<br/>";
        echo "<b>Ο χρήστης είναι γυναίκα: </b>" .
$resultwoman1['Score'] . " %<br/><br/>";
        if ($resultchild1['child']>=3){
            echo "<b>Ο χρήστης σχεδόν σίγουρα έχει παιδί.
<br/><br/>";
        }
        else if ($resultchild1['child']==1 ||
$resultchild1['child']==2){
            echo "<b>Ο χρήστης ίσως έχει παιδί. <br/><br/>";
        }
        else{
            echo "<b>Ο χρήστης δεν έχει παιδί. <br/><br/>";
        }
        echo "<b>Σύμφωνα με τις επιλογές του χρήστη, το ποσοστό
επισκεψιμότητας κάθε δραστηριότητας έχει ως εξής:</b><br/>";
        echo "<b>Τρέξιμο: </b>" . $resultrunning1['running'] . "
%<br/>";
        echo "<b>Ποδόσφαιρο: </b>" . $resultfootball1['football']
. " %<br/>";
        echo "<b>Μπάσκετ: </b>" .
$resultbasketball1['basketball'] . " %<br/>";
        echo "<b>Γυμναστική: </b>" . $resultfitness1['fitness'] .
" %<br/>";
        echo "<b>Τένις: </b>" . $resulttenis1['tenis'] . "
%<br/>";
        echo "<b>Πεζοπορία: </b>" . $resulthiking1['hiking'] . "
%<br/>";
        echo "<b>Κολύμβηση: </b>" . $resultswimming1['swimming']
. " %<br/>";
        echo "<b>Ποδηλασία: </b>" . $resultbiking1['biking'] . "
%<br/><br/>";
        echo "<b>Ο χρήστης δημιουργήθηκε στις: </b>" .
$resultacccreated1['acccreated'] . " <br/><br/>";

```

```

        echo "<b>Η τελευταία σύνδεση έγινε στις: </b>" .
$resultlastlogin1['lastlogin'] . "<br/><br/>";
        echo "<b>0 χρήστης έχει συνδεθεί: </b>" .
$resulttimesloggedin1['timesloggedin'] . " φορές από την στιγμή που
δημιούργησε τον χρήστη του.<br/><br/>";
        echo "<b>0 χρήστης έχει ψωνίσει απο το e-shop: </b>" .
$resulttimesbought1['timesbought'] . " φορές.<br/><br/>";
        echo "<b>0 χρήστης έχει ξοδέψει: </b>" .
$resultmoneyspent1['moneyspent'] . " € σε αγορές στο e-shop.<br/><br/>";
        if ($resulttimesbought1['timesbought']>=1){
            echo "Προσωπικά στοιχεία χρήστη:<br/>";
            echo "<b>Πόλη: </b>" . $resultcity1['city'] .
" <br/>";
            echo "<b>Διεύθυνση: </b>" . $resultadress1['adress']
. " <br/>";
            echo "<b>E-mail: </b>" . $resultmail1['mail'] .
" <br/>";
            echo "<b>Τηλέφωνο: </b>" . $resultphone1['phone'] .
" <br/><br/>";
        }
        echo "<b>0 χρήστης έχει αγοράσει τα παρακάτω προϊόντα:
</b><br/><br/>";
        $list=mysqli_query($link,"SELECT p.name, p.code, p.sex,
p.category, p.type, p.price, i.item_quantity, (i.item_quantity*p.price) AS
Total_Price FROM itemsbought i INNER JOIN products p ON p.code=i.item_code
WHERE i.item_member='" . $search . "'");
        echo "<table border='1'>";
        echo "<tr> <th>Όνομα Προϊόντος</th> <th>Κωδικός</th>
<th>Φύλο</th> <th>Κατηγορία</th> <th>Τύπος</th> <th>Τιμή</th>
<th>Ποσότητα</th> <th>Τελική Τιμή</th></tr>";
        while ($row = mysqli_fetch_array($list)) {
            echo "<tr><td>";
            echo $row['name'];
            echo "</td><td>";
            echo $row['code'];
            echo "</td><td>";
            echo $row['sex'];
            echo "</td><td>";
            echo $row['category'];
            echo "</td><td>";
            echo $row['type'];
            echo "</td><td>";
            echo $row['price'];
            echo "</td><td>";
            echo $row['item_quantity'];
            echo "</td><td>";
            echo $row['Total_Price'];
            echo "</td></tr>";
        }
        echo "</table>";
    }
    else {
        header("location:falseuser.php");
    }
}
?>

```

6.7.3 Στατιστικά ιστοσελίδας

Το τελευταίο πράγμα που πραγματοποιεί η εφαρμογή είναι να εμφανίζει τα στατιστικά της ιστοσελίδας το οποίο σημαίνει ότι συγκεντρώνει όλες τις πληροφορίες από όλους τους χρήστες και τους βλέπει σαν σύνολο. Όπως και πριν, αρχικά, εκτελεί μια σειρά από ερωτήματα (queries) στην βάση δεδομένων.

```
$link = mysqli_connect("mysql.hostinger.gr", "u282837298_root", "tei@123",
"u282837298_store") or die("cannot connect");
$sql1="SELECT * FROM members";
$result1=mysqli_query($link, $sql1);
$count1=mysqli_num_rows($result1);
$resultrunning=mysqli_query($link,"SELECT sum(running) AS running FROM
tracking");
$resultfootball=mysqli_query($link,"SELECT sum(football) AS football FROM
tracking");
$resultbasketball=mysqli_query($link,"SELECT sum(basketball) AS basketball
FROM tracking");
$resultfitness=mysqli_query($link,"SELECT sum(fitness) AS fitness FROM
tracking");
$resulttenis=mysqli_query($link,"SELECT sum(tenis) AS tenis FROM tracking");
$resulthiking=mysqli_query($link,"SELECT sum(hiking) AS hiking FROM
tracking");
$resultswimming=mysqli_query($link,"SELECT sum(swimming) AS swimming FROM
tracking");
$resultbiking=mysqli_query($link,"SELECT sum(biking) AS biking FROM
tracking");
$resultrunning1=mysqli_fetch_assoc($resultrunning);
$resultfootball1=mysqli_fetch_assoc($resultfootball);
$resultbasketball1=mysqli_fetch_assoc($resultbasketball);
$resultfitness1=mysqli_fetch_assoc($resultfitness);
$resulttenis1=mysqli_fetch_assoc($resulttenis);
$resulthiking1=mysqli_fetch_assoc($resulthiking);
$resultswimming1=mysqli_fetch_assoc($resultswimming);
$resultbiking1=mysqli_fetch_assoc($resultbiking);
$resultsum=mysqli_query($link,"SELECT
sum(running+football+basketball+fitness+tenis+hiking+swimming+biking) AS sum
FROM tracking");
$resultsum1=mysqli_fetch_assoc($resultsum);
$resulttimesbought=mysqli_query($link,"SELECT sum(timesbought) as timesbought
FROM tracking");
$resulttimesbought1=mysqli_fetch_assoc($resulttimesbought);
$resultmoneyspent=mysqli_query($link,"SELECT sum(moneyspent) as moneyspent
FROM tracking");
$resultmoneyspent1=mysqli_fetch_assoc($resultmoneyspent);
```

Και τελικά καταλήγει και τα εμφανίζει όλα μαζί έτσι ώστε να είναι κατανοητά από τον διαχειριστή.

```

<div class="add-content">
    <?php
        echo "<b>Σύνολο χρηστών: </b>" . $count1 . "<br/><br/>";
        echo "<b>Σύνολικες προβολές ανα κατηγορία: </b><br/>";
        echo "<b>Τρέξιμο: </b>" . $resultrunning1['running'] . "
προβολές.<br/>";
        echo "<b>Ποδόσφαιρο: </b>" . $resultfootball1['football'] . "
προβολές.<br/>";
        echo "<b>Μπάσκετ: </b>" . $resultbasketball1['basketball'] . "
" προβολές.<br/>";
        echo "<b>Γυμναστική: </b>" . $resultfitness1['fitness'] . "
προβολές.<br/>";
        echo "<b>Τένις: </b>" . $resulttenis1['tenis'] . "
προβολές.<br/>";
        echo "<b>Πεζοπορία: </b>" . $resulthiking1['hiking'] . "
προβολές.<br/>";
        echo "<b>Κολύμβηση: </b>" . $resultswimming1['swimming'] . "
προβολές.<br/>";
        echo "<b>Ποδηλασία: </b>" . $resultbiking1['biking'] . "
προβολές.<br/><br/>";
        echo "<b>Σύνολο: </b>" . $resultsum1['sum'] . "
προβολές.<br/><br/>";
        echo "<b>Σύνολο αγορών: </b>" .
$resulttimesbought1['timesbought'] . "<br/><br/>";
        echo "<b>Συνολικό κέρδος από αγορές e-shop: </b>" .
$resultmoneyspent1['moneyspent'] . " €<br/><br/>";
    ?>
</div>

```

Κεντρική Σελίδα Υποστήριξη Το Κατάστημα Επικοινωνία Καλάθι Αγορών

Καλωσήρθες, admin. [Έξοδος](#)

Κατηγορίες

- Τρέξιμο
- Ποδόσφαιρο
- Μπάσκετ
- Γυμναστική
- Τένις
- Πεζοπορία
- Κολύμβηση
- Ποδηλασία

Διαχείριση Ιστοσελίδας

- Προσθήκη Αντικειμένου
- Στατιστικά Χρηστών
- Στατιστικά Ιστοσελίδας

Σύνολο χρηστών: 6

Σύνολικες προβολές ανα κατηγορία:

- Τρέξιμο: 79 προβολές.
- Ποδόσφαιρο: 25 προβολές.
- Μπάσκετ: 26 προβολές.
- Γυμναστική: 18 προβολές.
- Τένις: 16 προβολές.
- Πεζοπορία: 7 προβολές.
- Κολύμβηση: 5 προβολές.
- Ποδηλασία: 8 προβολές.

Σύνολο: 184 προβολές.

Σύνολο αγορών: 35

Συνολικό κέρδος από αγορές e-shop: 1400.48 €

Εικόνα 6.7.3 Η σελίδα στατιστικών

Κεφάλαιο 7

Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Καθώς οι χρήστες του διαδικτύου και οι πληροφορίες που σχετίζονται με αυτούς αυξάνονται γίνεται όλο και μεγαλύτερη ανάγκη να μπορούμε να ξεχωρίζουμε τους χρήστες ανάλογα με τις προτιμήσεις και τα ενδιαφέροντά τους. Έτσι οι πληροφορίες αυτές να μπορούν να αξιοποιούνται από άλλα πρόσωπα ή εφαρμογές προς όφελος του χρήστη.

Στα πλαίσια της πτυχιακής εργασίας δημιουργήθηκε ένα δυναμικό σύστημα το οποίο είναι ικανό να 'χτίσει' καθένα προφίλ χρήστη, που δημιουργήθηκε, σύμφωνα με τις επιλογές που κάνει στο σύστημα και ύστερα από ανάλυση ο χρήστης παίρνει μια μορφή πιο προσωποποιημένη.

Με λίγα λόγια μπορούμε να πούμε ότι συστήματα ανάλυσης των προφίλ των χρηστών σαν και αυτό κάνουν δυνατή την διαφοροποίηση των χρηστών και την εξαγωγή αυτής της γνώσης σε άλλους ανθρώπους ή εφαρμογές. Έτσι συμπεράνουμε ότι μια τέτοια εφαρμογή είναι σε θέση να κάνει τις πληροφορίες που κατακλύζουν το διαδίκτυο περισσότερο κατανοητές και χρήσιμες.

7.2 Μελλοντικές Επεκτάσεις

Η πτυχιακή εργασία αυτή μπορεί να αναπτυχθεί περισσότερο χρησιμοποιώντας τις πληροφορίες που μάζευε για τους χρήστες. Μερικές από τις εφαρμογές που μπορούν να προστεθούν στο σύστημα είναι:

Ένα σύστημα συστάσεων θα ήταν η ιδανική επιλογή για μια εφαρμογή εμπορικού καταστήματος καθώς έτσι το σύστημα θα μπορεί να προτείνει στους χρήστες προϊόντα που μπορεί να τους ενδιαφέρουν σύμφωνα με τα ενδιαφέροντά τους, κάτι που θα ήταν προς όφελος του

ηλεκτρονικού καταστήματος και συνεπώς να έφερνε περισσότερες πωλήσεις.

Επίσης ένα σύστημα προσαρμοστικών διαφημίσεων οι οποίες θα στοχεύουν τους χρήστες που, τυχών, ενδιαφέρονται από το διαφημιζόμενο προϊόν. Με αυτόν τον τρόπο, οι χρήστες, θα γλυτώνουν από άσκοπες διαφημίσεις μειώνοντας έτσι τα κόστη των εταιριών που θέλουν να διαφημίσουν ένα προϊόν.

Τέλος, το σύστημα θα μπορούσε να αναβαθμιστεί σημαντικά εάν μπορούσε δημιουργήσει ομάδες με τους χρήστες ανάλογα με τα ενδιαφέροντά τους. Ουσιαστικά θα είναι μια μορφή εξόρυξης δεδομένων η οποία θα επιτρέπει να δημιουργούνται σχέσεις μεταξύ των χρηστών της εφαρμογής προς όφελος του ηλεκτρονικού καταστήματος.

Βιβλιογραφία

- [1] *User Profile*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/User_profile
Wikipedia, the free encyclopedia. (2015). Retrieved 2015, from User profile.
- [2] *User Modeling*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/User_modeling
Wikipedia, the free encyclopedia. (2015). Retrieved 2015, from User modeling.
- [3] Fischer, Gerhard (2001), "*User Modeling in Human-Computer Interaction*", *User Modeling and User-Adapted Interaction* 11: 65
- [4] Brusilovsky, Peter (2001). "*Adaptive hypermedia*". *User Modeling and User-Adapted Interaction* 11 (1-2): 95-97
- [5] Brusilovsky, Peter (2001). "*Adaptive Hypermedia*". *User Modeling and User-Adapted Interaction* 11 (1-2): 87-88
- [6] Brusilovsky, Peter (1996). "*Methods and Techniques of Adaptive Hypermedia*". *User Modeling and User-Adapted Interaction* 6 (2-3): 87–129
- [7] Francesco Ricci and Lior Rokach and Bracha Shapira, *Introduction to Recommender Systems Handbook*, *Recommender Systems Handbook*, Springer, 2011, p 1-3
- [8] Nielsen, J. (1994). *Usability Engineering*, Academic Press Inc, p 165
- [9] Clifton, Christopher (2010). "*Encyclopedia Britannica: Definition of Data Mining*". Retrieved from <http://www.britannica.com/technology/data-mining>. Retrieved 2015.
- [10] Kyriazanos, D., & Olesen, H. (2007). *Specification of user profile, identity and role management for PNs and integration to the PN platform*.
- [11] WWRF. (2009). *User Profiles Personalization and Privacy*, p 22-23.
- [12] Johnson, Addie; Taatgen, Niels (2005), "*User Modeling*", *Handbook of human factors in Web design*, Lawrence Erlbaum Associates, p 424–439

- [13] O'Neil, Cathy and, Schutt, Rachel (2014). *Chapter 2: Statistical Inference, Exploratory Data Analysis, and the Data Science Process, Doing Data Science*, p 17-50
- [14] Adèr, H.J. (2008). Chapter 14: Phases and initial steps in data analysis. In H.J. Adèr & G.J. Mellenbergh (Eds.) (with contributions by D.J. Hand), *Advising on Research Methods: A consultant's companion*, p 333–356.
- [15] *Data Visualization*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Data_visualization
Wikipedia, the free encyclopedia. (2015). Retrieved 2015, from Data visualization.
- [16] Fischer, Gerhard (2001), "*User Modeling in Human-Computer Interaction*", *User Modeling and User-Adapted Interaction*, p 6-7.
- [17] Zhang, Jie (2003), "*GUMSAWS: A Generic User Modeling Server for Adaptive Web Systems*", BCS, University of New Brunswick, p 3-6.
- [18] *What is web server?* (2010, 11 23). Retrieved 2015, from Web Developer Notes: http://www.webdevelopersnotes.com/basics/what_is_web_server.php