

Τ.Ε.Ι. ΗΠΕΙΡΟΥ

Τ.Ε.Ι. OF EPIRUS



ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ & ΟΙΚΟΝΟΜΙΑΣ (Σ.Δ.Ο)
ΤΜΗΜΑ ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΟΙΚΗΣΗΣ

SCHOOL OF MANAGEMENT AND ECONOMICS
*DEPARTMENT OF COMMUNICATIONS,
INFORMATICS AND MANAGEMENT*

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΓΙΑ
ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΠΟΥ
ΑΦΟΡΟΥΝ ΤΗΝ ΑΠΟΔΟΣΗ ΤΑΞΙΔΙΩΝ ΜΙΑΣ
ΕΤΑΙΡΙΑΣ**

Αναγνωστοπούλου Ελένη

ΑΜ 3468

Υπεύθυνος Καθηγητής:
Τσιαντής Λεωνίδας

ΠΑΤΡΑ
ΜΑΡΤΙΟΣ 2006

Αφιερωμένη στον μοναδικό άνθρωπο
που με έμαθε να αγαπώ,
στον πατέρα μου

ΠΡΟΛΟΓΟΣ

Σκοπός αυτής της εργασίας είναι η παρουσίαση μιας αυτοματοποιημένης εφαρμογής απόδοσης ταξιδιών σε μια εταιρία μέσα από την οποία θα αναδείξουμε τις υπηρεσίες του σύγχρονου Internet και πώς αυτές αλληλεπιδρούν με τις τεχνολογίες του διαδικτύου.

Στην πορεία της εργασίας θα γίνει μια μικρή αναφορά στην ιστορία του Internet και τα χαρακτηριστικά του καθώς επίσης και στις τεχνολογίες που θα χρησιμοποιήσουμε για την διεκπεραίωση της εφαρμογής.

Κλείνοντας το σύντομο πρόλογο θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου Τσιαντή Λεωνίδα (Καθηγητή του τμήματος Τηλεπληροφορικής και Διοίκησης) για την επίβλεψη της εργασίας, την πολύτιμη βοήθειά του και τη συμπαράστασή του προκειμένου να ολοκληρωθεί με επιτυχία η διπλωματική εργασία.

Θα ήθελα επίσης να ευχαριστήσω τους καλούς μου φίλους Ηλέκτρα Κίτσου, Σοφία Αραβαντινού, Γιάννη Αντωνέλλη, Κωνσταντίνο Μπαρούνη για την συμπαράστασή τους και τη βοήθειά τους καθ' όλη τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας.

Αναγνωστοπούλου Ελένη
Πάτρα Μάρτιος 2006

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΕΧΟΜΕΝΑ	4
ΓΛΩΣΣΑΡΙΟ	6
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	7
ΛΙΣΤΑ ΕΙΚΟΝΩΝ	7
1 ΕΙΣΑΓΩΓΗ	10
1.1 INTERNET	12
1.2 Οι υπηρεσίες που προσφέρονται μέσω του Internet	13
1.3 Το Μέλλον του Διαδικτύου	14
1.4 Στατικές Ιστοσελίδες	15
1.5 Δυναμικές Ιστοσελίδες	16
1.6 HTML γλώσσα	20
1.7 Περιορισμοί HTML	20
1.8 Η Δομή μιας HTML Σελίδας	21
1.9 Δυναμική HTML	23
1.10 Hypertext Transfer Protocol(HTTP)	23
1.11 Apache	25
2 ΕΠΙΛΟΓΗ ΤΕΧΝΟΛΟΓΙΩΝ	27
2.1 Βάση Δεδομένων	27
2.1.1 ΓΙΑΤΙ MYSQL	27
2.1.2 ΕΠΙΛΕΓΟΝΤΑΣ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	28
2.2 Τεχνολογία Δημιουργίας	29
2.2.1 ΓΙΑΤΙ PHP	29
2.2.2 ΓΙΑΤΙ JSP	30
2.2.3 ΓΙΑΤΙ ASP	31
2.2.4 ΓΙΑΤΙ CFML	31
2.3 SESSIONS	31
2.4 ΤΙ ΕΙΝΑΙ ΤΟ DREAMWEAVER	32
2.5 Τελική επιλογή τεχνολογιών	33
2.6 Χρήση των επιλεγμένων τεχνολογιών για δημιουργία εφαρμογής διαδικτύου	34
2.6.1 ΔΗΜΟΣΙΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΤΗΣ MYSQL ΣΤΟ WEB	34
2.6.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ ΤΗΣ MYSQL ΜΕ ΤΗΝ PHP	35
2.6.3 Εισαγωγή Δεδομένων στη Βάση Δεδομένων	37

3	ΑΠΑΙΤΗΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	39
3.1	Απαιτήσεις της εταιρίας	39
3.1.1	Κατηγορίες χρηστών του συστήματος.....	39
3.1.2	Πληροφορίες για κάθε ταξίδι	40
3.1.3	Διαδικασία απόδοσης χρηματικής αμοιβής.....	40
4	Σχεδιασμός του συστήματος.....	43
4.1	Η ΑΝΑΓΚΗ ΓΙΑ ΕΝΑ DATABASE-DRIVEN WEB SITE	43
4.2	ΞΕΚΙΝΗΜΑ ΜΕ ΤΟ DREAMWEAVER MX	44
4.2.1	ΕΙΣΑΓΩΓΗ ΣΤΟ DREAMWEAVER MX.....	45
4.3	CSS(Cascade Style Sheet)	48
4.3.1	EMBEDDED STYLE SHEETS	48
4.3.2	CLASSES.....	50
4.4	Η βάση δεδομένων και ανάλυση του κώδικα	52
4.4.1	ΑΝΑΛΥΣΗ ΤΩΝ ΠΙΝΑΚΩΝ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	54
4.4	Περιγραφή του συστήματος	55
4.5.7	ΑΛΛΑΓΗ CONFIGURATION.....	62
4.6	Διασύνδεση βάσεων δεδομένων και Web	63
4.6.1	ΣΕΛΙΔΕΣ ΠΟΛΛΑΠΛΟΥ ΣΚΟΠΟΥ (MULTI-PURPOSE PAGES).....	64
5	ΘΕΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ	66
5.1	Αρχεία Εφαρμογής	66
5.2	Πηγαίος Κώδικας ενδεικτικών λειτουργιών	68
5.2.1	ΣΥΝΔΕΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	68
5.2.3	ΑΡΧΙΚΟΠΟΙΗΣΗ SESSION	69
5.2.4	ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΤΟΥ ΧΡΗΣΤΗ	69
5.2.5	ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΑΠΟ ΦΟΡΜΑ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	70
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	71
	ΠΑΡΑΡΤΗΜΑ.....	72

ΓΛΩΣΣΑΡΙΟ

ADSL	Asymeric Digital Subscriber Line
Browser	Φυλλομετρητής
Cable	Καλωδιακή
CSS	Cascade Style Sheet
Content	Περιεχόμενο
Cyberspace	Κυβερνοχώρος
e-mail	Ηλεκτρονικό Ταχυδρομείο
Format	Μορφοποίηση
HTML Editor	Πρόγραμμα με το οποίο μπορεί να γίνει επεξεργασία HTML γλώσσας
Interactive	Αλληλεπιδρών
Interperter	Διερμηνέας
ISDN	Integraned Services Digital Network/It Still Does Nothing
Link	Σύνδεσμος (αναφέρεται σε ιστοσελίδα)
Modem	Modulator/Demodulator
Multimedia	Πολυμέσα
MIME	Multipurpose Internet Mail Extensions
Software	Λογισμικό
Tag	Επικεφαλίδα. Ο όρος χρησιμοποιείται για τις δηλώσεις σε HTML γλώσσα
Template	Περίγραμμα
URL	Uniform Resource Locator
WWW	World Wide Web
Website	Κυβερνοχώρος

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

DBMS	DataBase Management System
HTML	HyperText Mark-up Language
PHP	PHP Hypertext Preprocessor
URL	Uniform Resource Locator
WWW	World Wide Web
ΒΔ	Βάση Δεδομένων
CSS	Cascade Style Sheet
GUI	Graphical User Interface
MIME	Multipurpose Internet Mail Extensions

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1	Αρχιτεκτονική πελάτη-εξυπηρετητή
Εικόνα 2	Οπτικό αποτέλεσμα κώδικα HTML
Εικόνα 3	Μοντέλο πελάτη- εξυπηρετητή
Εικόνα 4	Εξέλιξη της χρήσης της PHP σε παγκόσμιο επίπεδο
Εικόνα 5	Επιφάνεια εργασίας του Dreamweaver
Εικόνα 6	Σχεδιάγραμμα της Βάσης Δεδομένων
Εικόνα 7	Διάγραμμα ER

Εικόνα 8	Διάγραμμα ροής πληροφορίας
Εικόνα 9	Αρχική σελίδα εφαρμογής
Εικόνα 10	Σελίδα Διαχειριστή
Εικόνα 11	Σελίδα μελλοντικών ταξιδιών
Εικόνα 12	Εισαγωγή νέας εγγραφής
Εικόνα 13	Εκτυπώσιμη μορφή ταξιδιού
Εικόνα 14	Λίστα ταξιδιών
Εικόνα 15	Εισαγωγή νέου χρήστη
Εικόνα 16	Εμφάνιση χρηστών
Εικόνα 17	Δημιουργία configuration
Εικόνα 18	Εμφάνιση όλων των configuration

1 ΕΙΣΑΓΩΓΗ

Το Internet αποτελεί σήμερα ένα από τα ισχυρότερα εργαλεία πληροφόρησης και διάδοσης νέων τεχνολογιών, σε παγκόσμιο επίπεδο. Είναι μία αχανής βιβλιοθήκη συγκεντρωμένης γνώσης, προσπελάσιμη από τον καθένα. Το ευρύτερο και πιο πολύπλευρο μέσο επικοινωνίας και διάδοσης απόψεων και ιδεών...

...είναι πολλοί οι τρόποι με τους οποίους συνήθως περιγράφεται το internet. Υπάρχει όμως μία εκδοχή που δεν ακούγεται συχνά, είναι όμως η πιο αντιπροσωπευτική περιγραφή του σύγχρονου ρόλου του internet.

Το internet είναι το πιο οικονομικό μέσο για τη δυναμική και απεριόριστη επέκταση των δυνατοτήτων ενός απλού υπολογιστή.

Σήμερα, το να έχει κανείς στη διάθεση του **ισχυρά** και **έξυπνα** εργαλεία που θα χρησιμοποιήσει στη δουλειά του, **δεν** σημαίνει απαραίτητα την επένδυση σε ισχυρά υπολογιστικά συστήματα υψηλών προδιαγραφών, και όγκους δεδομένων αποθηκευμένους σε σκληρούς δίσκους υψηλής χωρητικότητας.

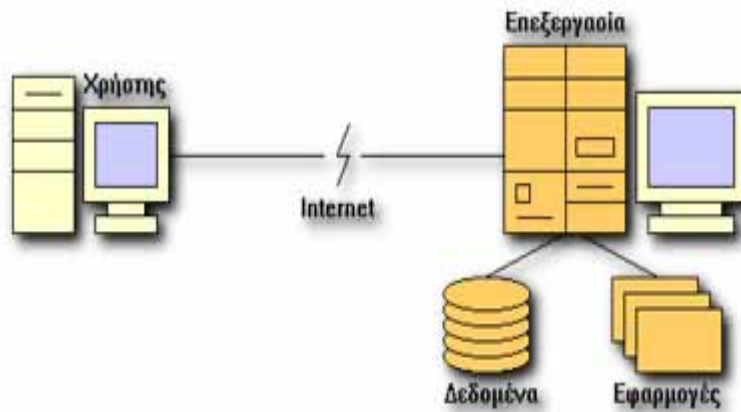
Ο χρήστης του internet μπορεί πλέον να εκμεταλλεύεται τον ισχυρό εξοπλισμό ενός εξειδικευμένου φορέα, ο οποίος διαθέτει τόσο τις εφαρμογές, όσο και τα δεδομένα. Και όλα αυτά, με **ένα απλό PC**, και τη σύνδεση με το internet.

1.1 INTERNET

Γνωστό και ως world wide web ή WWW. Ο παγκόσμιος ιστός είναι ο χαρακτηρισμός που προσδίδεται στο γραφικό περιβάλλον που διέπει το Internet. Αποτελεί δε, μια συλλογή από έγγραφα online, εγκατεστημένα σε υπολογιστές server παντού στον κόσμο, δημιουργώντας κατά αυτόν τον τρόπο, το πλέον ορατό και εύκολα προσβάσιμο κομμάτι του internet. Η πρόσβαση πραγματοποιείται με τη συνδρομή του web browser. Οι σελίδες στο internet συνδυάζουν συνήθως, κείμενο, γραφικά και φωτογραφίες, ακόμα και video. Κάθε σελίδα ή site, έχει τη δική της ξεχωριστή διεύθυνση (URL), η οποία προσημαίνεται από τα γράμματα www.

Κάποτε το Internet ήταν λιγότερο εντυπωσιακό απ' ό,τι σήμερα. Οι σελίδες του Web είχαν αποκλειστικά κείμενο και μικρές εικόνες και κατέβαιναν βασανιστικά αργά με τα πρώτα μόντεμ, τα οποία δούλευαν με την... εκπληκτική ταχύτητα των 9,6 Kbps... Από τότε όμως όλα έχουν αλλάξει -το Internet έγινε multimedia, «στολίστηκε» με φωτογραφίες, ήχους, μουσική, βίντεο και άρχισε να απαιτεί περισσότερη μνήμη, ταχύτερους επεξεργαστές και κάρτες γραφικών και πάνω από όλα ταχύτερη σύνδεση.

Έτσι από τις απλές αναλογικές συνδέσεις και τα 33άρια ή 56άρια modem πήγαμε στην ISDN, κάποιοι -λίγοι- τυχεροί στο T1, το T3 και την καλωδιακή (Cable) σύνδεση. Η έλευση του ADSL στην Ελλάδα υπόσχεται σαφώς μεγαλύτερες ταχύτητες, αλλά πολλοί δεν αρκούνται σε αυτές τις συνδέσεις. Έτσι επιλέγουν το δορυφορικό Internet, με ταχύτητες που θεωρητικά μπορούν να φθάσουν ακόμα και τα 8,8 GB το δευτερόλεπτο! Ειδικά μετά τη λειτουργία του ελληνικού δορυφόρου Hellas sat, τα πράγματα για το «ελληνικό» δορυφορικό Internet αναμένεται να αλλάξουν δραματικά.



Εικόνα 1. Αρχιτεκτονική πελάτη-εξυπηρετητή

Η εκρηκτική άφιξη του Internet είναι τυπικά συνδεδεμένη με την ανάπτυξη του World Wide Web (Παγκόσμιος Ιστός), που έκανε ένα μεγάλο και αναπτυσσόμενο μερίδιο των μέσων του Internet εύκολα προσβάσιμο. Η κυριαρχία του Web με τη σειρά της είναι συνδεδεμένη με μία εταιρεία που άρχισε ως πρωτόπειρη εταιρεία, αποκαλούμενη ως Netscape. Η Netscape επιτυχώς εμπορευματοποίησε ένα εργαλείο, που ονομαζόταν Web Browser, το οποίο οργάνωνε, απλούστευε και βοηθούσε να επεκταθούν οι λειτουργικές προσβάσεις στο Web. Για να εδραιώσει τις προοπτικές της σε μελλοντικές αγορές, η Netscape πρόσφερε δωρεάν τον πλοηγητή στους περισσότερους ατομικούς χρήστες. Η εταιρεία είχε έσοδα από το να δίνει άδεια για χρήση του πλοηγητή όπως επίσης και άλλου λογισμικού, σε επιχειρήσεις.

1.2 Οι υπηρεσίες που προσφέρονται μέσω του Internet

Το Internet παρέχει διάφορες υπηρεσίες, οι σημαντικότερες από τις οποίες είναι:

- ✓ **Ηλεκτρονικό Ταχυδρομείο (electronic mail).** Οι χρήστες του διαδικτύου μπορούν να ανταλλάσσουν ηλεκτρονικά μηνύματα που υποστηρίζουν απλό και μορφοποιημένο κείμενο, ήχο, κινητή και κινούμενη εικόνα
- ✓ **Μεταφορά αρχείων (ftp).** Οι χρήστες μπορούν να μεταφέρουν οντότητες πληροφορίας από και προς απομακρυσμένους κόμβους.
- ✓ **Πίνακες Ανακοινώσεων (bulletin boards) και Λίστες Συζητήσεων (discussion lists).** Οι χρήστες έχουν την δυνατότητα πρόσβασης σε ομάδες

ειδήσεων, πίνακες ανακοινώσεων και λίστες συζητήσεων που έχουν κάποιο συγκεκριμένο θέμα.

- ✓ **Συζητήσεις πραγματικού χρόνου.** Είναι δυνατή η συζήτηση μεταξύ δυο ή και περισσότερων χρηστών με ανταλλαγή είτε γραπτών μηνυμάτων (talk), είτε φωνής (audioconference), είτε κινούμενης εικόνας και ήχου (videoconference).
- ✓ **Απομακρυσμένη πρόσβαση (telnet).** Οι χρήστες μπορούν να προσπελαίνουν υπολογιστές που είναι γεωγραφικά απομακρυσμένοι και να εκτελούν διάφορα προγράμματα. Μπορούν επίσης να έχουν και προσπέλαση σε απομακρυσμένες βάσεις δεδομένων.
- ✓ **Παγκόσμιος Ιστός Πληροφορίας (World Wide Web, WWW).** Το σύστημα αυτό επιτρέπει την πρόσβαση σε τεράστιο όγκο πληροφορίας που έχει την μορφή πολυμέσων και είναι οργανωμένη ιεραρχικά. Είναι δυνατή η προσπέλαση σε εφημερίδες, περιοδικά, καταλόγους βιβλιοθηκών, βιβλία και άρθρα τα οποία είναι αποθηκευμένα σε ηλεκτρονική μορφή.
- ✓ **Υπηρεσία Καταλόγου (X.500).** Πρόκειται για μια υπηρεσία που λειτουργεί σε δεύτερο επίπεδο υποστηρικτικά στις παραπάνω υπηρεσίες αλλά και αυτόνομα αφού αποτελεί ένα μοναδικό ολοκληρωμένο σύστημα. Φροντίζει για τον ορισμό, την καταχώρηση, την αποθήκευση, τη διαχείριση των οντοτήτων από τα οποία αποτελούνται τα επικοινωνιακά συστήματα που απαρτίζουν το Internet και των ονομάτων τους.

1.3 Το Μέλλον του Διαδικτύου

Η θεαματική ανάπτυξη του Διαδικτύου συνεχίζεται όλο και περισσότερο, το διαδίκτυο καλωσορίζει κάθε μήνα εκατομμύρια καινούργιους χρήστες και οι μεγαλύτερες εταιρίες στον κόσμο επενδύουν εκατομμύρια.

Με βάση αυτά τα στοιχεία οι παρακάτω προβλέψεις σχετικά με το μέλλον του διαδικτύου είναι μάλλον βέβαιες:

- Τα εργαλεία ανάπτυξης εξυπηρετητών Παγκόσμιου Ιστού θα τείνουν όλο και περισσότερο στην παροχή δυνατοτήτων που θα ελκύουν τους χρήστες.
- Η διανομή του λογισμικού θα γίνεται όλο και περισσότερο μέσω του διαδικτύου με αποτέλεσμα να μειωθεί η χρησιμοποίηση CD-ROM και δισκετών.
- Το video και audio streaming θα διαδοθεί ακόμα περισσότερο και η σύγκλιση της τηλεόρασης και υπολογιστή θα είναι πολύ κοντά.

- Θα αναπτυχθεί ιδιαίτερα το ηλεκτρονικό εμπόριο (electronic commerce).
- Η ανωνυμία ένα από τα ελκυστικότερα χαρακτηριστικά του διαδικτύου για τους περισσότερους σιγά-σιγά θα μειωθεί καθώς αναμένετε να μπου κανονισμοί και αρχές στη λειτουργία του.
- Η στρατηγική χρέωσης των υπηρεσιών του διαδικτύου αναμένετε να αλλάξει και να προσαρμοστεί στο είδος δεδομένων που κάθε χρήστης διακινεί.

1.4 Στατικές Ιστοσελίδες

Με τον όρο "στατική σελίδα" εννοούμε μία σελίδα σε μορφή HTML (ο κώδικας με τον οποίο δομείται μια σελίδα του web), η οποία μπορεί να δημιουργηθεί και από εσάς τους ίδιους, χωρίς ιδιαίτερες γνώσεις, με χρήση ενός προγράμματος επεξεργασίας HTML, ακόμη και με το γνωστό σε όλους Word.

Περισσότερες από μία στατικές σελίδες συναποτελούν ένα στατικό δικτυακό τόπο. Ουσιαστικά κάθε στατικό site αποτελεί ένα σύνολο ιστοσελίδων με σταθερό περιεχόμενο, οι οποίες μπορούν να περιλαμβάνουν εκτενείς πληροφορίες (όπως στοιχεία επικοινωνίας, τιμές, προϊόντα/υπηρεσίες, γραφικά), αλλά δεν μπορούν να τροποποιούνται εύκολα, γρήγορα και με χαμηλό κόστος.

Προκειμένου να γίνει πιο κατανοητό το παραπάνω, ας φέρουμε ως παράδειγμα ένα μικρό site εταιρίας ένδυσης, το οποίο διαθέτει ένα κατάλογο προϊόντων. Αν η επιχείρηση ασχολείται αποκλειστικά με την πώληση πέντε ειδών πουλόβερ, για παράδειγμα, τότε θα πρέπει να δημιουργηθούν πέντε διαφορετικές σελίδες προϊόντων, αναγράφοντας η καθεμία τις λεπτομέρειες του κάθε προϊόντος. Ακόμη, θα χρειαστεί μία σελίδα στην οποία θα αναφέρουμε λίγα λόγια για την εταιρία/κατάστημα, άλλη μία για τις παραγγελίες, μία σελίδα επικοινωνίας του χρήστη με το κατάστημα, και πιθανώς μερικές ακόμα.

Σε ένα στατικό site, όλες αυτές οι σελίδες δημιουργούνται μία φορά και δημοσιεύονται στον web server. Οι επισκέπτες μπορούν να έχουν πρόσβαση στις σελίδες του δικτυακού τόπου και να ενημερωθούν για τα προϊόντα ώστε να προχωρήσουν πιθανώς στην παραγγελία, η οποία στην προκειμένη περίπτωση δεν πραγματοποιείται online αλλά μέσω ηλεκτρονικού

ταχυδρομείου. Ο χρήστης δηλαδή αποστέλλει σε εσάς ένα e-mail, στο οποίο σας αναφέρει το προϊόν που θέλει να αγοράσει και εσείς στη συνέχεια θα πρέπει να επικοινωνήσετε μαζί του, να του στείλετε το προϊόν και να λάβετε τα χρήματά σας με αντικαταβολή.

Εάν υποθέσουμε τώρα ότι θέλετε να προσθέσετε μερικά ακόμη προϊόντα, πουλόβερ στο παράδειγμά μας, τότε θα πρέπει να δημιουργήσετε επιπλέον σελίδες για το κάθε προϊόν. Εάν πάλι θέλετε να αλλάξετε κάποιο από τα στοιχεία (τηλέφωνο, διεύθυνση, τιμές, κ.λπ.), τότε θα πρέπει να ανατρέξετε σε καθεμία από τις σελίδες που περιέχουν κάποιο απ' αυτά τα στοιχεία και να το αλλάξετε "με το χέρι". Φανταστείτε λοιπόν ένα site με 200 σελίδες, που όλες στο κάτω μέρος αναφέρουν το τηλέφωνο της εταιρίας σας. Για να αλλάξετε τον αριθμό, είστε υποχρεωμένος να μπειτε και στις 200 σελίδες ξεχωριστά...

1.5 Δυναμικές Ιστοσελίδες

Στον αντίποδα έχουμε τις λεγόμενες "δυναμικές σελίδες" ή αλλιώς "δυναμικά sites". Στην πραγματικότητα οι δυναμικές σελίδες δεν προϋπάρχουν - αποτελούν καταχωρίσεις σε μια βάση δεδομένων, και όταν καλούνται από το χρήστη δημιουργούνται εκείνη τη στιγμή αυτόματα για εκείνον.

Όπως και οι στατικές σελίδες, είναι οικονομικές στην κατασκευή (μικρό κόστος ανά σελίδα), και επιτρέπουν εύκολη συντήρηση και παρακολούθηση (για μεγάλους δικτυακούς τόπους, με μεγάλο αριθμό προϊόντων και ειδών). Δεν είναι όμως τόσο εύκολο να δημιουργηθούν από έναν άπειρο χρήστη, καθώς απαιτούνται κάποιες βασικές γνώσεις προγραμματισμού και χρήσης εφαρμογών και βάσεων δεδομένων (SQL, PHP, ASP, Perl, κ.λπ.).

Το κόστος τους μπορεί να είναι υψηλό για μικρά websites, ειδικά σε ό,τι αφορά τη συντήρησή τους, κι αυτό γιατί, καθώς απαιτούνται κάποιες επιπλέον γνώσεις, το πιθανότερο είναι ότι θα πρέπει να αναθέσετε σε κάποιον τρίτο την όλη διαδικασία ενημέρωσης και συντήρησης.

Παλαιότεροι browsers (παλιές εκδόσεις του Internet Explorer και του Netscape Navigator) μπορεί να παρουσιάζουν προβλήματα στην εμφάνιση των σελίδων αυτών, ωστόσο αυτό είναι ένα ζήτημα που τείνει να εκλείπει, καθώς το μεγαλύτερο ποσοστό των χρηστών του Internet χρησιμοποιεί τις πιο

πρόσφατες εκδόσεις. Υπάρχει όμως ένα λειτουργικό πρόβλημα: οι σελίδες αυτές δεν μπορούν να καταχωρηθούν αυτόνομα στις μηχανές αναζήτησης. Λογικό, αν λάβουμε υπόψη ότι ουσιαστικά οι σελίδες αυτές δεν υφίστανται. Τέλος, το κόστος μεταφοράς του δικτυακού τόπου σε άλλο πάροχο είναι συνήθως αρκετά υψηλό. Κατ' αρχάς θα πρέπει να τονίσουμε ότι ο όρος "δυναμικός" είναι πολυδιάστατος. Τις περισσότερες φορές όμως αναφέρεται στον αυτόματο τρόπο με τον οποίο δημιουργούνται οι σελίδες ενός website. Ο δυναμικός μηχανισμός δημιουργεί τις σελίδες αυτές διαβάζοντας πληροφορίες που προέρχονται από μια βάση δεδομένων, εισάγοντας τις πληροφορίες (data) σε ένα "template" και στέλνοντας τη σελίδα πίσω στον web browser.

Τι είναι το template; Στα ελληνικά θα μπορούσαμε να το αποδώσουμε με τον όρο "πρότυπο εμφάνισης". Είναι ο τρόπος εμφάνισης μιας σελίδας, το "καλούπι" και το συγκεκριμένο της στήσιμο ανεξαρτήτως περιεχομένου. Για παράδειγμα, θα μπορούσαμε να ορίσουμε από την αρχή ότι μια σελίδα παρουσίασης προϊόντος θα αποτελείται από το όνομα του προϊόντος, από κάτω θα υπάρχει ο κωδικός του, στα δεξιά μία μικρή φωτογραφία του, κάτω από τη φωτογραφία μια σύντομη περιγραφή και κάτω από την περιγραφή, στα δεξιά, η τιμή του. Μπορούμε να ακολουθήσουμε το ίδιο στήσιμο σε όλα τα προϊόντα, είτε πρόκειται για μπλούζες είτε για φούστες κ.ο.κ.

Αντί να σχεδιάσουμε λοιπόν μια ξεχωριστή σελίδα για καθένα από τα προϊόντα που διαθέτει ένα κατάστημα, φτιάχνουμε μία και μόνο, η οποία αποτελεί το βασικό σκελετό παρουσίασης των προϊόντων. Ο μηχανισμός που λειτουργεί από πίσω (back end) φροντίζει μέσα στο σκελετό αυτό να εμφανίσει το προϊόν που θα ζητήσει εκείνη τη στιγμή ο χρήστης (front end).

Σύμφωνα λοιπόν με το παράδειγμα, και ενώ η εταιρία έχει μεγαλώσει, διαθέτει πλέον περισσότερα από 100 είδη ρουχισμού. Όπως είναι λογικό, οι στατικές σελίδες είναι εξαιρετικά δύσχρηστες. Κάθε φορά που χρειάζεται να γίνει μία αλλαγή στην τιμή, να προστεθεί κάποια λεπτομέρεια, να ανανεωθεί το εικαστικό κομμάτι του site κλπ., πρέπει να παρέμβει κάποιος σε 100 διαφορετικές σελίδες.

Ένα επιπλέον αρνητικό σημείο στις στατικές σελίδες εντοπίζεται στο "index" των σελίδων. Εφόσον υπάρχουν 100 σελίδες (μία για κάθε προϊόν), με δυσκολία πλέον θα βρεθεί η συγκεκριμένη σελίδα στο back end (το μηχανισμό

που είναι υπεύθυνος για τη διαχείριση του όλου συστήματος) ώστε να γίνει μία αλλαγή.

Αντίθετα, ένας δυναμικός δικτυακός τύπος θα σας επέτρεπε να κάνετε τα ακόλουθα:

- Να αλλάζετε με ευκολία κάποια πληροφορία σε όλες τις σελίδες του site ταυτόχρονα, κάνοντας μόνο μία αλλαγή στη βάση δεδομένων ή στο template, η οποία θα ενημερώνει όλες τις σελίδες.
- Να αλλάζετε εύκολα τις πληροφορίες σχετικά με τα προϊόντα. Για παράδειγμα, επιθυμείτε να κάνετε αλλαγές στον τιμοκατάλογο που αφορά μόνο σε μπλουζάκια μεγέθους Large/X-Large. Θα προστεθεί ένα επιπλέον πεδίο στη βάση δεδομένων που θα απεικονίζει την αλλαγή αυτή.
- Να επιλέγει ο επισκέπτης τον τρόπο με τον οποίο θα βλέπει τα προϊόντα, καθώς έχει τη δυνατότητα να εξατομικεύει τις σελίδες χρησιμοποιώντας διάφορα κριτήρια δικής του επιλογής, που υπάρχουν ήδη στη βάση δεδομένων, π.χ. μέγεθος, χρώμα, ύφασμα, τιμή κ.λπ.

Η ύπαρξη ενός δυναμικού site σας παρέχει και άλλα πλεονεκτήματα.

- Ένα από τα βασικά είναι και η αποθήκευση πληροφοριών σχετικά με τις προτιμήσεις των πελατών και η εξατομίκευση των σελίδων σύμφωνα με τις ανάγκες κάθε πελάτη. Για παράδειγμα, να εμφανίζει ρούχα σε σκούρες αποχρώσεις σε κάποιον επισκέπτη που έχει αγοράσει στο παρελθόν τα περισσότερα ρούχα σε σκούρα χρώματα.
- Δυνατότητα δημιουργίας στατιστικών αναφορών για λήψη αποφάσεων.
- Επιπλέον, δίνει τη δυνατότητα δημιουργίας αναφορών σχετικά με τα προϊόντα που έχουν επιλέξει κατά καιρούς οι επισκέπτες του site, τα έχουν προσθέσει στο καλάθι τους αλλά δεν έχουν φθάσει ποτέ στην παραγγελία.
- Έχει τη δυνατότητα να "θυμάται" ποια ήταν η τελευταία σελίδα που επισκέφθηκε ο χρήστης πριν από την παραγγελία του, ώστε να προσδιοριστούν οι επιπρόσθετες πληροφορίες που χρειάζεται κανείς προκειμένου να φθάσει σε αυτήν.

- Όλα αυτά μπορούν να είναι συνδεδεμένα με μία βάση δεδομένων της αποθήκης του φυσικού καταστήματος, ώστε να προειδοποιείται ο πελάτης αν επιλέξει κάποιο προϊόν που δεν είναι διαθέσιμο

Η χρήση δυναμικών σελίδων σας παρέχει επιπλέον τη δυνατότητα αξιοποίησης τεχνολογιών όπως τα "message boards", όπου χρειάζεται να αποθηκεύονται πολλές πληροφορίες για μεγάλο διάστημα. Ο χρήστης αλλά και ο διαχειριστής του site έχουν την ευχέρεια να αναζητήσουν κάποια πληροφορία από μία βάση δεδομένων, δηλώνοντας λέξη-κλειδί στη μηχανή αναζήτησης και περιμένοντας μερικά δευτερόλεπτα. Σαφώς καλύτερο από το να ψάχνουν 100 ή 1000 web σελίδες μία προς μία.

Τέλος, οφείλουμε να προσθέσουμε ότι τα εργαλεία για την ανάπτυξη δυναμικών δικτυακών τόπων εξελίσσονται και βελτιώνονται συνεχώς, πράγμα που μειώνει σημαντικά το κόστος ανάπτυξης. Αν θέλετε λοιπόν να επιλέξετε μια λύση που να καλύπτει τις μακροπρόθεσμες ανάγκες σας με περισσότερα πλεονεκτήματα και πολύ μικρή διαφορά κόστους, αυτή θα πρέπει να είναι "δυναμική".

Εν κατακλείδι, θα λέγαμε ότι οι απλές HTML σελίδες κατασκευάζονται γρήγορα και φθηνά (για μικρά καταστήματα με λίγα είδη), είναι όμως πολύ ακριβές στην παρακολούθηση και συντήρηση για μεγάλα sites με μεγάλο αριθμό προϊόντων. Έχουν καλύτερη αναγνωσιμότητα (αν είναι σωστά φτιαγμένες, τις "βλέπουν" όλοι οι browsers), είναι φθηνότερες στη συντήρηση (που μπορεί εύκολα να γίνει από εσάς), καταχωρούνται όλες στις μηχανές αναζήτησης, και μεταφέρονται χωρίς μεγάλη δυσκολία από έναν πάροχο σε άλλο. Ωστόσο, οι δυνατότητές τους σταματούν εδώ.

Οτιδήποτε περισσότερο μπορεί να γίνει μόνο με τη χρήση δυναμικών σελίδων. Ειδικά όταν πρόκειται για ηλεκτρονικό κατάστημα, ακόμα και το πιο βασικό εργαλείο, το καλάθι αγορών (shopping cart), δεν μπορεί να λειτουργήσει αν δεν υπάρχει ο κατάλληλος μηχανισμός, που μόνο οι δυναμικές σελίδες μπορούν να προσφέρουν.

1.6 HTML γλώσσα

Η γλώσσα προγραμματισμού που χρησιμοποιείται στο World Wide Web είναι η HTML. Είναι μια δηλωτική γλώσσα ή αλλιώς γλώσσα χαρακτηρισμού. Πρόκειται για μια markup γλώσσα ιδανική για τη συγγραφή hypertext (κείμενο με το οποίο ο χρήστης μπορεί να αλληλεπιδράσει), για την ενσωμάτωση multimedia και την παρουσίαση μικρών και σχετικά απλών κειμένων στο WWW. Η HTML είναι μια συγκεκριμένη εφαρμογή μιας πιο γενικής markup γλώσσας που ονομάζεται SGML (Standard Generalized Markup Language, ISO 8879) και αποτελεί ένα σύστημα προτυποποίησης για τον καθορισμό και τη χρήση μορφοποιήσεων κειμένου στο WWW. Η HTML είναι σήμερα ο πιο συνηθισμένος τρόπος έκφρασης ιδεών και πληροφορίας σε υπερκείμενο. Με αυτήν οι προγραμματιστές έχουν τις δυνατότητες:

Να κατασκευάζουν online κείμενα με επικεφαλίδες, κείμενο, πίνακες, λίστες, φωτογραφίες κλπ.

Να εξάγουν online πληροφορία μέσω links υπερκειμένου, με ένα απλό κλικ του ποντικιού

Να σχεδιάζουν φόρμες για την αλληλεπίδραση με υπηρεσίες από απόσταση, προκειμένου να χρησιμοποιηθούν για την αναζήτηση πληροφοριών, κρατήσεις, παραγγελία προϊόντων κλπ.

Να συμπεριλαμβάνουν φύλλα εργασίας, αποσπάσματα video, ηχητικά αποσπάσματα και άλλες εφαρμογές κατευθείαν στα κείμενα τους.

Μέσω των σελίδων αυτών ο χρήστης πλοηγείται στην υπηρεσία, εισάγει τα δεδομένα που αυτός επιθυμεί προκειμένου να γίνει μια αναζήτηση, λαμβάνει τα αποτελέσματα κάθε αναζήτησης ή μηνύματα λάθους αν δεν έδωσε σωστά ή επαρκή δεδομένα και μεταφέρεται σε άλλες υπηρεσίες του εκπαιδευτικού δικτύου.

1.7 Περιορισμοί HTML

Η HTML μπορεί μόνο να παρουσιάσει κείμενο, πολυμεσικά στοιχεία και υπερσυνδέσμους.

Δεν διαθέτει η ίδια μηχανισμό για την επεξεργασία στοιχείων η δεδομένων που δίνονται από τον χρήστη

Δεν διαθέτει μηχανισμό για διαχείριση δομών δεδομένων ή για διαχείριση ενεργειών που γίνονται δυναμικά από την πλευρά του χρήστη .

Τα παραπάνω έχουν σαν αποτέλεσμα να μην μπορούμε με την HTML να αναπτύξουμε ολοκληρωμένες εφαρμογές, οι οποίες δίνουν την δυνατότητα στο χρήστη να εκτελεί υπολογισμούς ή δυναμικές λειτουργίες.

1.8 Η Δομή μιας HTML Σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία κειμένου σε μορφή ASCII, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο.

Τα αρχεία της HTML περιέχουν τα ακόλουθα :

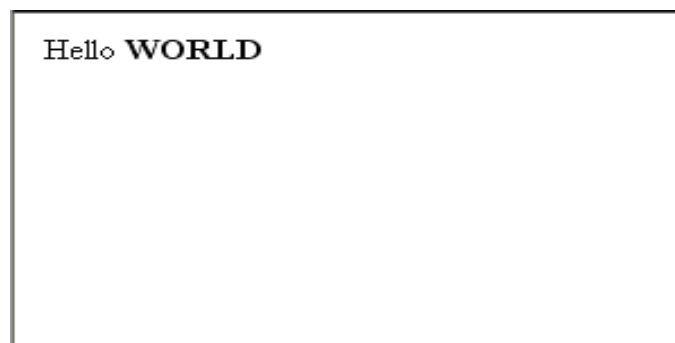
- Το κείμενο της σελίδας.
- Τα tags της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερ-κειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πολυμέσα).

Τα tags της HTML έχουν γενικά ένα tag αρχής και ένα tag τέλους ή ένα tag ανοίγματος και ένα tag κλεισίματος, τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό, όπως είναι για παράδειγμα η έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tags αρχής, αλλά με πρόθεμα τον χαρακτήρα /.

Δεν αποτελούν ζευγάρι όλα τα tags της HTML, καθώς ορισμένα είναι “μονομελή”, ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα <>. Ακόμη, όλα τα tags της HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων, δηλ. μπορούμε να τα γράφουμε είτε με κεφαλαίους είτε με πεζούς χαρακτήρες είτε με οποιονδήποτε συνδυασμό τους. Συνήθως τα γράφουμε κεφαλαία για να μπορούμε να τα ξεχωρίζουμε ευκολότερα από το κείμενο της σελίδας.

```
<table>
  <tr>
    <td>
      <p> Hello <b> WORLD </b> </p>
    </td>
  </tr>
</table>
```

Κώδικας 1. Κομμάτι κώδικα HTML

A rectangular box containing the rendered output of the HTML code from the previous block. The text inside the box is "Hello WORLD", where "WORLD" is in a bold font.

Εικόνα 2. Οπτικό αποτέλεσμα κώδικα HTML

Τα αρχεία της HTML πρέπει να τα γράφουμε σε συντάκτες κειμένων, δηλ. προγράμματα που μπορούν να αποθηκεύσουν αρχεία κειμένου σε μορφή ASCII. Τέτοια προγράμματα είναι τα Notepad, Wordpad και Write των Windows, το Edit του DOS και τα vi, emacs και pico του UNIX.

Το όνομα του αρχείου θα πρέπει να έχει επέκταση **.html** ή **.htm** σε συστήματα DOS ή Windows που επιτρέπουν επεκτάσεις μόνο τριών χαρακτήρων.

Η HTML χρησιμοποιεί τρία tags για την περιγραφή της συνολικής δομής μιας σελίδας, τα οποία παρέχουν ορισμένες απλές πληροφορίες κεφαλίδας. Αυτά τα tags προσδιορίζουν τη σελίδα μας στους φυλλομετρητές και παρέχουν επίσης απλές πληροφορίες για τη σελίδα, όπως τον τίτλο ή τον συγγραφέα της, πριν από τη φόρτωση ολόκληρης της σελίδας.

1.9 Δυναμική HTML

Οι παραδοσιακές στατικές σελίδες μπορούν τώρα να αποτελέσουν παρελθόν, επειδή οι σελίδες που συντάσσονται με τη δυναμική HTML διαθέτουν πιο ελκυστικό περιεχόμενο και σας προσφέρουν μια μοναδική, διασκεδαστική και πιο ικανοποιητική εμπειρία. Με τη δυναμική HTML, web browser παραδίδει ιστοσελίδες πολυμέσων, που συναγωνίζονται τα αγαπημένα σας αλληλεπιδραστικά CD-ROM και επιταχύνουν τη λήψη τους, επειδή αυτές οι σελίδες δημιουργούνται με χρήση της HTML, που απαιτεί λίγη μνήμη σε σύγκριση με τα γραφικά. Οι περιπτώσεις νέας μετάβασης στο διακομιστή ελαχιστοποιούνται, γεγονός που σημαίνει ότι η απόδοση του προγράμματος ανάγνωσης ιστοσελίδων είναι ταχύτερη στον επιτραπέζιο υπολογιστή σας.

Ποια είναι τα οφέλη;

- ✓ Μπορείτε να προβάλλετε τη μεγαλύτερη ποικιλία συναρπαστικού περιεχομένου του Internet με καλύτερη απόδοση. Παλιότερα, οι μόνες σελίδες που μπορούσαν να ληφθούν γρήγορα ήταν ανιαρές, στατικές και σας προσέφεραν μόνο κείμενο. Τώρα, με τη δυναμική HTML, μπορείτε να προβάλλετε ελκυστικά οπτικά εφέ που εμφανίζονται σε δευτερόλεπτα. Επειδή η δυναμική HTML αποστέλλει όλο το περιεχόμενο στην επιφάνεια εργασίας σας, μπορείτε να ταξινομήσετε τα δεδομένα και να ζητήσετε γρήγορα πρόσθετες σελίδες, αφού δεν χρειάζεται να επιστρέψετε στο διακομιστή.

1.10 Hypertext Transfer Protocol(HTTP)

Το HTTP είναι η κύρια μέθοδος που χρησιμοποιούν τα πρωτόκολλα του Παγκόσμιου Ιστού για να μεταφέρουν δεδομένα ανάμεσα σε έναν εξυπηρετητή και ένα πελάτη. Πριν τον Παγκόσμιο Ιστό και το HTTP, το FTP ήταν το κύριο πρωτόκολλο εισόδου/εξόδου που χρησιμοποιούνταν για την μεταφορά αρχείων στο διαδίκτυο. Το HTTP είναι ένα απλό, μικρό και γρήγορο πρωτόκολλο εισόδου/εξόδου το οποίο καταλαβαίνει τα URLs και

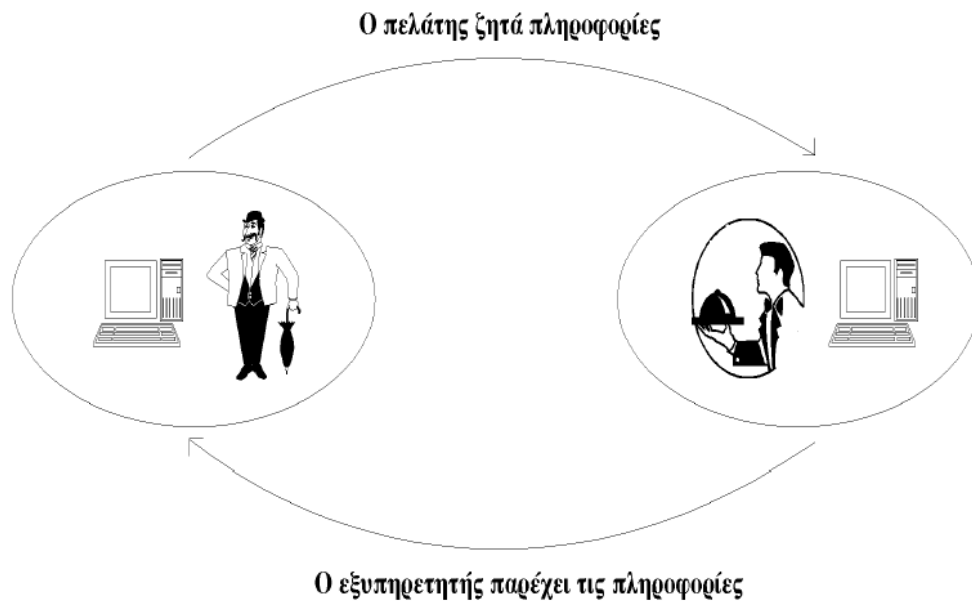
χρησιμοποιείται σε περιβάλλοντα υπερκειμένου/υπερμέσων. Δεν έχει διαφορετικές καταστάσεις σε αντίθεση με το FTP και χρησιμοποιεί λίγες εντολές. Επίσης χρησιμοποιεί MIME κωδικοποίηση υποστηρίζοντας έτσι πολλαπλά formats.

Το HTTP είναι ένα πρωτόκολλο πελάτη/εξυπηρετητή και ακολουθεί το μοντέλο αίτησης/απόκρισης. Η λειτουργία του βασίζεται στη σύνδεση ενός HTTP πελάτη(συνήθως είναι ένας φυλλομετρητής Παγκόσμιου Ιστού-Web Browser) με έναν HTTP εξυπηρετητή (συνήθως είναι ένας εξυπηρετητής Παγκόσμιου Ιστού αλλά μπορεί να είναι ένας proxy ή ένας gateway εξυπηρετητής) χρησιμοποιώντας ένα URL και απαιτώντας κάποιο πόρο, όπως ένα HTML αρχείο. Τα δεδομένα που μεταφέρονται είναι σπασμένα σε δυο κομμάτια, στα κυρίως δεδομένα(είναι σε MIME format) και στις πληροφορίες που είναι απαραίτητες για τη μεταφορά ανάμεσα στον πελάτη και τον εξυπηρετητή. Παραδοσιακά η <<συνομιλία>> ανάμεσα στον πελάτη και στον εξυπηρετητή χρησιμοποιεί την θύρα 80, παρόλα αυτά μπορούν να χρησιμοποιηθούν και άλλες θύρες αν αυτές οριστούν στο URL. Επιπλέον το HTTP μπορεί να χρησιμοποιηθεί και με άλλα πρωτοκόλλα εκτός του TCP/IP.

Η έκδοση του HTTP που αυτή την στιγμή χρησιμοποιείτε είναι η 1.1 (<http://www.w3.org/Protocols/HTTP>), η οποία υποστηρίζετε από το σύνολο των εξυπηρετητών καθώς και των φυλλομετρητών Παγκόσμιου Ιστού.

Μερικά από τα κυριότερα χαρακτηριστικά αυτής της έκδοσης είναι τα ακόλουθα:

- Χρησιμοποιεί λιγότερες TCP συνδέσεις σε σχέση με το HTTP 1.0
- Υποστηρίζει συμπίεση και αποσυμπίεση των δεδομένων που μεταφέρονται.
- Υποστηρίζει πολλαπλές γλώσσες .
- Υποστηρίζει μεταφορά ενός κομματιού ενός αρχείου.
- Υποστηρίζει virtual hosting.



ΤΟ ΜΟΝΤΕΛΟ ΠΕΛΑΤΗ - ΕΞΥΠΗΡΕΤΗΤΗ

Εικόνα 3. Μοντέλο πελάτη- εξυπηρετητή

1.11 Apache

Ο Apache είναι ένα ένας ανοικτός κεντρικός υπολογιστής δικτύου HTTP πηγής για τα Unix-ομοειδή συστήματα (BSD, Linux, και συστήματα Unix), Microsoft Windows, Novell NetWare και άλλες πλατφόρμες. Ο Apache είναι ξεχωριστός για τη διαδραμάτιση ενός βασικού ρόλου στην αρχική αύξηση του World Wide Web, και συνεχίζει να είναι ο δημοφιλέστερος κεντρικός υπολογιστής δικτύου σε λειτουργία, που χρησιμεύει ως η πλατφόρμα αναφοράς ενάντια στην οποία άλλοι κεντρικοί υπολογιστές δικτύου σχεδιάζονται και κρίνονται. Ο Apache χαρακτηρίζει τα ιδιαίτερα διαμορφώσιμα μηνύματα λάθους, τις ΠΔΒΔ (πρόγραμμα διαχείρισης βάσεων δεδομένων)-βασισμένες βάσεις δεδομένων επικύρωσης, και τη διαπραγμάτευση περιεχομένου. Υποστηρίζεται επίσης από διάφορα γραφικά ενδιάμεσα με τον χρήστη (GUIs) που επιτρέπουν την ευκολότερη, πιο διαισθητική διαμόρφωση του κεντρικού υπολογιστή. Ο κεντρικός υπολογιστής Apache αναπτύσσεται και διατηρείται από μια ανοικτή κοινότητα των

υπεύθυνων για την ανάπτυξη υπό την αιγίδα του ιδρύματος λογισμικού
Apache.

2 Επιλογή Τεχνολογιών

2.1 Βάση Δεδομένων

Οι πιθανές επιλογές που έχουμε όσον αφορά τη βάση δεδομένων του συστήματος προέρχονται από την επιλογή των τεχνολογιών για τους μηχανισμούς κατηγοριοποίησης και κατασκευής του Portal. Συνεπώς θα πρέπει να επιλεγεί μία βάση δεδομένων η οποία να είναι πλήρως συμβατή με το μηχανισμό που θα κατηγοριοποιεί καθώς επίσης και με τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για την κατασκευή του portal. Θεωρούμε πως ο μηχανισμός δημιουργίας του δυναμικού προφίλ δύναται να ενταχθεί, είτε στο μηχανισμό κατηγοριοποίησης είτε στο μηχανισμό κατασκευής του portal.

2.1.1 Γιατί MySQL

Η MySQL είναι η δημοφιλέστερη Βάση Δεδομένων ανοιχτού κώδικα που προσφέρεται από το Δίκτυο MySQL. Η αρχιτεκτονική της την κάνουν να είναι εξαιρετικά γρήγορη και πολύ εύκολη σε αλλαγές και αναβαθμίσεις. Επιτρέπει επαναχρησιμοποίηση κώδικα όπου αυτό είναι αναγκαίο και παρέχει ένα μινιμαλιστικό τρόπο δημιουργίας στοιχείων διαχείρισης βάσης δεδομένων τέτοια ώστε να κάνουν τη MySQL ασύγκριτη σε ταχύτητα, σε κατάληψη χώρου, σταθερότητα και ευκολία. Ο μοναδικός στο είδος του διαχωρισμός του κεντρικού πυρήνα του server από το μηχανισμό

αποθήκευσης κάνει δυνατή την ύπαρξη αυστηρού ελέγχου σε συναλλαγές και μείωση ταχύτητας ή ύπαρξη θεαματικά μεγάλης ταχύτητας με απευθείας προσπέλαση των δεδομένων στοιχεία που μπορεί να χρησιμοποιηθούν ανάλογα με τις ανάγκες των χρηστών.

Η MySQL περιλαμβάνει αποθήκευση σε μηχανή InnoDB, η οποία υποστηρίζει ασφάλεια στις συναλλαγές και ACID-συμβατή μηχανή αποθήκευσης με commit, rollback, crash recovery και low-level locking δυνατότητες.

Η έκδοση της MySQL που βρίσκεται αυτή τη στιγμή σε σταθερή κατάσταση είναι η 5 και υποστηρίζει πολλά στοιχεία που αφορούν την απόδοση, τη διεθνοποίηση και τη δυνατότητα ένταξης του MySQL server σε άλλα στοιχεία υλικού και λογισμικού. Τα πιο βασικά στοιχεία που χαρακτηρίζουν τη MySQL είναι:

- Υποερωτήματα, που επιτρέπουν στους χρήστες να κάνουν σύνθετα ερωτήματα με μεγάλη ευκολία και αποδοτικά.
- Γρήγορη επικοινωνία μεταξύ server και client μέσα από ένα καινούριο πρωτόκολλο
- Μικρότερη κατανάλωση πόρων από το server μέσα από βελτιστοποίηση στις βιβλιοθήκες
- Υποστήριξη Unicode, διεθνείς χαρακτήρες και υποστήριξη αποθήκευσης στην πλειοψηφία των συνόλων χαρακτήρων
- Υποστήριξη τύπων GIS για ερωτήματα που αφορούν χάρτες και γεωγραφικά δεδομένα

Τα παραπάνω στοιχεία κάνουν τη MySQL ένα υπερπολύτιμο εργαλείο στα χέρια κάποιου χρήστη και τη θέτουν στην 1^η θέση για επιλογή ως βάση δεδομένων του συστήματός μας. [21]

2.1.2 Επιλέγοντας τη Βάση Δεδομένων

Σύμφωνα με τα παραπάνω αλλά και λαμβάνοντας υπόψη μας τους σκοπούς που έχει το σύστημά μας καταλήξαμε στην επιλογή της MySQL σαν τη βάση δεδομένων που θα χρησιμοποιηθεί στο σύστημα. Συγκρίνοντας τις δύο βάσεις δεδομένων μπορούμε να καταλήξουμε στο ότι διαθέτουν πολλά

κοινά στοιχεία, ωστόσο η MySQL φαίνεται να είναι πιο διαδεδομένη, ένας λόγος ο οποίος την κάνει πιο ισχυρή.

Επιπρόσθετα τα στοιχεία διεθνοποίησης που διαθέτει φαίνονται πολύ χρήσιμα για ένα σύστημα το οποίο μελλοντικά μπορεί να επεκταθεί ώστε να υποστηρίζει πολλές γλώσσες. Τέλος θα πρέπει να λάβουμε υπόψη μας το γεγονός πως δημιουργούμε ένα σύστημα πολυεπίπεδο με τη βάση δεδομένων να είναι ο ουσιαστικός σύνδεσμος μεταξύ των περισσότερων κομματιών και συνεπώς μία βάση δεδομένων με μεγάλη σταθερότητα και αξιοπιστία θα προσέδιδε κύρος στο συνολικό σύστημα.

Καταλήγουμε λοιπόν στη χρήση Mysql Server έκδοση 5. [21]

2.2 Τεχνολογία Δημιουργίας

Οι σελίδες θα πρέπει να έχουν απλή δομή και κατανοητή προκειμένου να μην αποπροσανατολίζεται ο χρήστης. Για το σκοπό αυτό η δυνατότητα που μας δίνεται είναι να χρησιμοποιήσουμε μία εκ των PHP ή JSP.

2.2.1 Γιατί PHP

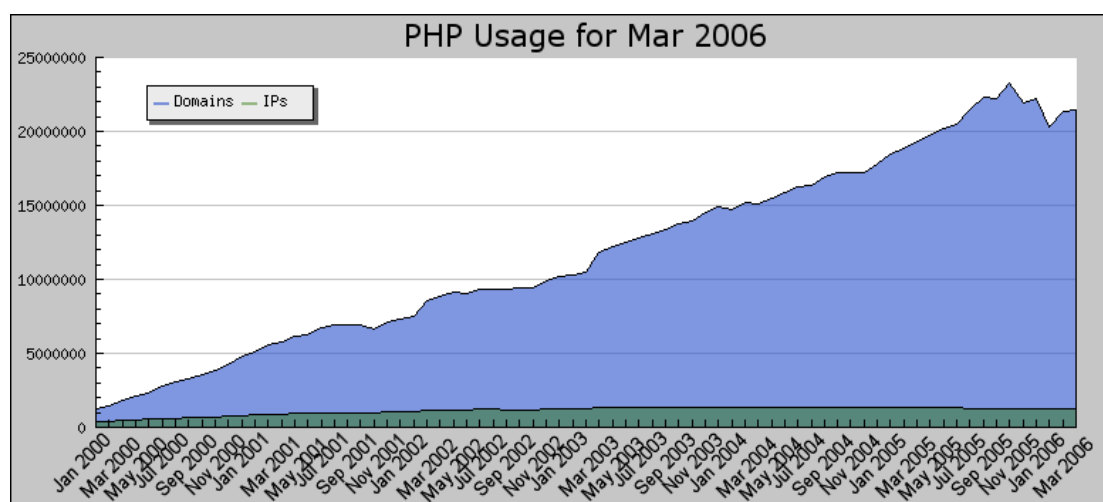
Η ευκολία στη χρήση αλλά και η ομοιότητα με της πιο κοινές γλώσσες δομημένου προγραμματισμού κάνουν την PHP μία γλώσσα η οποία ελκύει τους προγραμματιστές και οι πιο έμπειροι από αυτούς βρίσκουν εύκολη τη δημιουργία σύνθετων εφαρμογών από την πρώτη στιγμή που θα έρθουν σε επαφή με την PHP. Επίσης επιτρέπει στους έμπειρους χρήστες να δημιουργήσουν εφαρμογές Διαδικτύου με δυναμικό περιεχόμενο χωρίς να χρειάζεται να αναλωθούν σε πρακτικές ή να χρειαστεί να αποστηθίσουν σειρές από συναρτήσεις.

Ένα από τα πιο ελκυστικά κομμάτια της PHP είναι το γεγονός ότι είναι κάτι περισσότερο από μια προγραμματιστική γλώσσα. Εξαιτίας της κλιμακωτής σχεδιάσής της, μπορεί να χρησιμοποιηθεί και για τη δημιουργία γραφικών περιβαλλόντων απεικόνισης, και για την εκτέλεση προγραμμάτων μέσω της γραμμής εντολών.

Η PHP επιτρέπει την αλληλεπίδραση με ένα μεγάλο αριθμό σχεσιακών βάσεων δεδομένων όπως είναι οι Mysql, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL και SQLite ενώ η σύνταξη που χρησιμοποιείται είναι απλή

και κατανοητή. Τρέχει στα περισσότερα λειτουργικά συστήματα όπως UNIX, Linux, Windows και Mac OS X και μπορεί να υποστηριχθεί σχεδόν από όλους τους γνωστούς εξυπηρετητές εφαρμογών Διαδικτύου.

Η PHP είναι αποτέλεσμα μίας σειράς προσπαθειών από πολλούς συμμετέχοντες. Τα δικαιώματα παρέχονται με ένα SD-style license. Τέλος, μετά την έκδοση 4 η PHP υποστηρίζεται από τη μηχανή Zend [26].



Εικόνα 4. Εξέλιξη της χρήσης της PHP σε παγκόσμιο επίπεδο

2.2.2 Γιατί JSP

Η JSP έρχεται σαν απάντηση της Java στις τεχνολογίες εφαρμογών διαδικτύου. Χρησιμοποιεί τεχνολογία που βασίζεται είτε σε Java Servlets ή σε Java Beans και προσφέρει δυνατότητα ανάλογα με την επιλογή της τεχνολογίας να δημιουργηθούν από πολύ απλές Διαδικτυακές εφαρμογές μέχρι πολύ σύνθετες.

Όσον αφορά την αρχιτεκτονική, η jsp μπορεί να θεωρηθεί σαν servlet με πολύ υψηλού επιπέδου αφαίρεση η οποία υλοποιείται σαν επέκταση του API 2.1 των Servlet.

Όσον αφορά τη σύνταξη, μία σελίδα γραμμένη σε JSP μπορεί να χωριστεί στα εξής κομμάτια

- Στατικό περιεχόμενο (π.χ. HTML)
- JSP directives
- JSP μεταβλητές και στοιχεία κώδικα

- JSP action
- Tags γραμμένα από το χρήστη

Πρόκειται για τη γλώσσα προγραμματισμού που χρησιμοποιείται στις περισσότερες σύνθετες εφαρμογές που δημιουργούνται στο Διαδίκτυο γιατί προσφέρει τη δυνατότητα με τη χρήση συνδυασμού καθαρής Java, μέσω των Beans και μίας C-like γλώσσας προγραμματισμού για τη δημιουργία απλού δυναμικού περιεχομένου. Ωστόσο προορίζεται κυρίως για έμπειρους χρήστες που μπορούν να καταλάβουν τη διαφορά αντικειμενοστραφούς και συναρτησιακού προγραμματισμού και να τα συνδυάσουν κατάλληλα προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα [27].

2.2.3 Γιατί ASP

Η *ASP (Active Server Pages)* είναι μια τεχνολογία που βασίζεται στον IIS (Internet Information Server) της Microsoft. Μας δίνει τη δυνατότητα να συνδυάσουμε HTML, VBScript (Visual Basic Script) και JavaScript με επαναχρησιμοποιήσιμα ActiveX Components για να δημιουργήσουμε δυναμικές ιστοσελίδες. Υπάρχει και η τεχνολογία ASP.NET, η οποία βρίσκεται ακόμα σε ανάπτυξη.

2.2.4 Γιατί CFML

Η *CFML (Cold Fusion Markup Language)* αναπτύχθηκε αρχικά από την εταιρεία Allaire το 1995 και ξεκίνησε, όπως και η PHP, ως ένα σύνολο από scripts γραμμένα στη γλώσσα Perl. Εξελιχθηκε γρήγορα σ' έναν πλήρη application server και το 2001 η Macromedia εξαγόρασε την Allaire. Η CFML απέκτησε δημοτικότητα λόγω της σαφήνειας και της ευκολίας χρήσης της. Χρησιμοποιεί tags (ετικέτες) όπως η HTML, κάτι που την κάνει πολύ εύκολη στην εκμάθηση και γρήγορη στην υλοποίηση.

2.3 Sessions

Η υποστήριξη sessions στην PHP αποτελεί έναν τρόπο για την διατήρηση δεδομένων κατά την διάρκεια ακολουθιακών προσπελάσεων. Το

γεγονός αυτό επιτρέπει το χτίσιμο πιο παραμετροποιημένων εφαρμογών και πιο ελκυστικών site.

Όταν ένας επισκέπτης προσπελάβνει το site, του ανατίθεται ένας μοναδικός αριθμός, το session id. Το id αυτό αποθηκεύεται είτε σε κάποιο cookie ή μεταδίδεται μέσω του URL.

Η υποστήριξη sessions επιτρέπει την δημιουργία μεταβλητών που διατηρούν την τιμή τους κατά την διάρκεια πολλών request. Όταν ο επισκέπτης προσπελαύνει το site, η PHP αυτόματα ελέγχει αν κάποιο συγκεκριμένο session id έχει σταλεί με το request. Αν αυτό συμβαίνει, τότε το ήδη σωσμένο περιβάλλον μεταβλητών επαναδημιουργείται.

Παρακάτω φαίνονται παραδείγματα χρήσης των sessions για την αρχικοποίηση και διαγραφή μεταβλητών.

Παράδειγμα 1. Αρχικοποίηση μεταβλητής με την \$_SESSION.

```
<?php
session_start();
// Use $HTTP_SESSION_VARS with PHP 4.0.6 or less
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 0;
} else {
    $_SESSION['count']++;
}
?>
```

Παράδειγμα 2. Διαγραφή μεταβλητής με την \$_SESSION.

```
<?php
session_start();
// Use $HTTP_SESSION_VARS with PHP 4.0.6 or less
unset($_SESSION['count']);
?>
```

2.4 Τι Είναι το DreamWeaver

Το πρόγραμμα *DreamWeaver* της εταιρείας *Macromedia* είναι ένα κορυφαίο πρόγραμμα δημιουργίας και επεξεργασίας ιστοσελίδων, δηλαδή κώδικα HTML, που είναι ιδιαίτερα εύκολο και φιλικό στη χρήση του. Το

DreamWeaver είναι εξαιρετο για να μπορούμε να δημιουργήσουμε στα γρήγορα φόρμες (forms), πλαίσια (frames), πίνακες (tables) και άλλα αντικείμενα της HTML. Είναι, όμως, ιδιαίτερα καλό όταν θέλουμε να δώσουμε σε μια ιστοσελίδα τη δυνατότητα να κάνει κάτι. Πρέπει να έχουμε υπόψη μας ότι το DreamWeaver μπορεί να χρησιμοποιηθεί και για τη δημιουργία εφαρμογών πολυμέσων. Το DreamWeaver έχει δυνατότητες για δημιουργία δυναμικής HTML (DHMTL) και επιτρέπει κίνησης γραμμής χρόνου, απόλυτη τοποθέτηση περιεχομένων, δημιουργία επιπέδων (layers) και συγγραφή σεναρίων `al777ar` (scripts). Το DreamWeaver περιέχει δικές του συμπεριφορές (behaviors), που είναι έτοιμα scripts τα οποία μπορούμε να προσθέσουμε πολύ εύκολα σ' ένα αντικείμενο.

Το DreamWeaver μάς παρέχει την ελευθερία να σχεδιάσουμε οπτικά την εμφάνιση μιας ιστοσελίδας και τη δύναμη να την κάνουμε να λειτουργεί όπως ακριβώς θέλουμε. Μπορούμε να δημιουργήσουμε τη δική μας προσωπική ιστοσελίδα (personal web page) ή μια ολόκληρη περιοχή (web site) σ' ένα εταιρικό δίκτυο (intranet).

2.5 Τελική επιλογή τεχνολογιών

Στη δημιουργία ενός database-enabled web site υπάρχουν πολλοί τρόποι για να πετύχουμε το ίδιο αποτέλεσμα. Όλες αυτές οι διαφορετικές τεχνολογίες επιτυγχάνουν στο τέλος το ίδιο αποτέλεσμα αλλά μ' έναν ελαφρά διαφορετικό τρόπο. Οι κυριότερες εναλλακτικές λύσεις (τεχνολογίες) είναι η ASP (Active Server Pages), η JSP (Java Server Pages) και η CFML (Cold Fusion Markup Language). Υπάρχουν φυσικά κι άλλες, αλλά αυτές οι τέσσερις τεχνολογίες αποτελούν τους κυριότερους ανταγωνιστές όσον αφορά τη δημοτικότητα και υποστηρίζονται όλες από το Dreamweaver MX.

Η επιλογή των τεχνολογιών έγινε περισσότερο βάση γενικών αρχών και προτύπων προκειμένου να καταλήξουμε σε ένα τελικό σύστημα ανοιχτό, και ευέλικτο το οποίο θα μπορεί να επιδέχεται βελτιώσεις σε κάθε κομμάτι του ξεχωριστά. Έγινε, δηλαδή, προσπάθεια να μη δημιουργηθούν επικαλύψεις στον κώδικα αλλά η διασύνδεση των υποσυστημάτων να γίνει σε επίπεδο βάσης δεδομένων. Αυτό βέβαια δε μας απαγορεύει να χρησιμοποιούμε ένα κεντρικό μηχανισμό που θα κάνει διαχείριση όλων των υποσυστημάτων. Συνεπώς καταλήγουμε σε γλώσσα διαδικτύου PHP με

υποστήριξη βάσης δεδομένων MySQL γιατί επιθυμούμε απλότητα σε επίπεδο web site μέσα από την πλατφόρμα του Dreamweaver MX.

2.6 Χρήση των επιλεγμένων τεχνολογιών για δημιουργία εφαρμογής διαδικτύου

2.6.1 Δημοσίευση Δεδομένων της MySQL στο Web

Όταν κάποιος επισκεφθεί μια σελίδα σ' ένα database-driven Web site θα συμβούν τα εξής :

- Ο φυλλομετρητής του επισκέπτη θα ζητήσει τη Web page χρησιμοποιώντας ένα στάνταρτ URL.
- Το λογισμικό του Web server (Apache, IIS ή άλλο) αναγνωρίζει ότι το ζητούμενο αρχείο είναι ένα script της PHP και το διερμηνεύει (interprets) χρησιμοποιώντας το πρόσθετο (plug-in) που διαθέτει για την PHP πριν απαντήσει στην αίτηση για τη σελίδα (page request).
- Μερικές εντολές της PHP, που θα δούμε αργότερα, συνδέονται στη βάση δεδομένων της MySQL και ζητούν το περιεχόμενο (content) που ανήκει στην ιστοσελίδα (Web page).
- Η βάση δεδομένων MySQL απαντάει στέλνοντας το ζητούμενο περιεχόμενο στο script της PHP.
- Το script της PHP αποθηκεύει το περιεχόμενο σε μια ή περισσότερες μεταβλητές της PHP και μετά χρησιμοποιεί τη γνωστή μας συνάρτηση echo() για να το εμφανίσει σαν μέρος της ιστοσελίδας.
- Το plug-in της PHP τελειώνει, παραδίδοντας στον Web server ένα αντίγραφο του κώδικα της HTML που έχει δημιουργήσει.
- Ο Web server στέλνει τον κώδικα της HTML στον φυλλομετρητή (Web browser) όπως θα έκανε και μ' ένα κανονικό (απλό) αρχείο HTML, με τη διαφορά ότι αντί να προέλθει απευθείας από ένα αποθηκευμένο αρχείο HTML, η σελίδα αυτή είναι το αποτέλεσμα (έξοδος) που παρέχεται από το πρόσθετο (plug-in) της PHP.

2.6.2 Δημιουργία Σύνδεσης της MySQL με την PHP

Πριν μπορέσουμε να πάρουμε το περιεχόμενο μιας βάσης δεδομένων της MySQL για να το εισάγουμε σε μια ιστοσελίδα, πρέπει πρώτα να δούμε πώς θα καθιερώσουμε μια σύνδεση (connection) με την MySQL. Η κλήση στην ακόλουθη συνάρτηση της PHP δημιουργεί αυτή τη σύνδεση :

```
mysql_connect(<address>, <username>, <password>);
```

Όπου το *<address>* είναι η IP διεύθυνση ή το hostname του υπολογιστή στον οποίο εκτελείται το λογισμικό του MySQL server, που είναι το *localhost* αν εκτελείται στον ίδιο υπολογιστή με το λογισμικό του Web server, και τα *<username>* και *<password>* είναι τα ίδια με τα user name και password της MySQL που χρησιμοποιήσαμε για να συνδεθούμε στον MySQL server.

Η συνάρτηση `mysql_connect()` επιστρέφει έναν αριθμό για να μπορούμε να αναγνωρίσουμε (ξεχωρίσουμε) τη σύνδεση (connection). Μπορούμε να κρατήσουμε αυτή την τιμή σε μια μεταβλητή, ως εξής :

```
$link = mysql_connect("localhost", "root", "") or die("Could not connect : " . mysql_error());  
//echo "Connected successfully";  
mysql_select_db("egrisi") or die("Could not select database");
```

Η τιμή που επιστρέφει η συνάρτηση `mysql_connect()`, που θα την αποκαλούμε *connection identifier* (αναγνωριστικό σύνδεσης), αποθηκεύεται σε μια μεταβλητή με όνομα `$link`.

Εφόσον ο MySQL server αποτελεί ένα εντελώς ξεχωριστό κομμάτι λογισμικού, πρέπει να δούμε και την πιθανότητα να μην είναι διαθέσιμος ο server ή να μην είναι προσβάσιμος εξαιτίας μιας βλάβης στο δίκτυο ή επειδή ο συνδυασμός username/password που δώσαμε δεν γίνεται αποδεκτός από τον server.

Σ' αυτές τις περιπτώσεις, η συνάρτηση `mysql_connect()` δεν επιστρέφει έναν connection identifier εφόσον δεν γίνεται κάποια σύνδεση και αντί γι'

αυτὸν επιστρέφει την τιμὴ false. Ἐχουμε ἔτσι τη δυνατότητα να αντιδράσουμε σε τέτοιες καταστάσεις χρησιμοποιώντας μια εντολή if, ως εξής :

```
$link = @mysql_connect("localhost", "root", "mypasswd");
if (!$link) {
echo("<P>Η σύνδεση με τον " . "database server είναι αδύνατη</P>");
exit();
}
```

Υπάρχουν τρία καινούργια κόλπα στον παραπάνω κώδικα. Πρώτα, ἔχουμε τοποθετήσει το σύμβολο @ μπροστά από τη συνάρτηση mysql_connect(). Πολλές συναρτήσεις, ὅπως και η mysql_connect(), εμφανίζουν αυτόματα ἄσχημα μηνύματα λάθους ὅταν κάτι πάει στραβά. Η τοποθέτηση του συμβόλου @ μπροστά από το ὄνομα της συνάρτησης μᾶς δίνει τη δυνατότητα να εμφανίσουμε το δικό μας μήνυμα λάθους, που θα εἶναι πιο φιλικό.

Δεύτερο, τοποθετήσαμε ἓνα θαυμαστικό ! μπροστά από τη μεταβλητὴ \$dbcnx μέσα στη συνθήκη της εντολῆς if. Το ! εἶναι ὁ *τελεστής ἀρνήσης (negation operator)* της PHP, ὁ ὁποῖος στην ουσία μετατρέπει μια τιμὴ false σε true και μια τιμὴ true σε false. Ἐτσι, αν ἡ σύνδεση ἀποτύχει και ἡ συνάρτηση mysql_connect() επιστρέφει false, ἡ μεταβλητὴ !\$link θα γίνεῖ ἴση με true και θα ἐκτελεστοῦν οἱ εντολές που βρίσκονται μέσα στο σώμα της εντολῆς if.

Αντίθετα, αν γίνεῖ ἡ σύνδεση, ὁ connection identifier που εἶναι ἀποθηκευμένος στη μεταβλητὴ \$link θα γίνεῖ ἴσος με true, καθὼς κάθε ἀριθμὸς που εἶναι διαφορετικὸς ἀπὸ το 0 θεωρεῖται ὅτι εἶναι true στην PHP, και ἔτσι ἡ !\$link θα ἀποτιμηθεῖ σε false και δὲν θα ἐκτελεστοῦν οἱ εντολές που βρίσκονται μέσα στο σώμα της συνάρτησης if.

Το τελευταῖο ἀπὸ τα καινούργια κόλπα εἶναι ἡ συνάρτηση *exit()*, ἡ ὁποία δὲν δέχεται παραμέτρους και ἀναγκάζει τὴν PHP να σταματήσει να διαβάζει τὴ σελίδα σ' αὐτὸ το σημεῖο. Αὐτὴ εἶναι σωστὴ ἀπόκριση σε μια ἀποτυχημένη σύνδεση με μια βάση δεδομένων, ἐφόσον στις περισσότερες

περιπτώσεις η σελίδα δεν θα είναι σε θέση να εμφανίσει κάποιες χρήσιμες πληροφορίες χωρίς αυτή τη σύνδεση.

2.6.3 Εισαγωγή Δεδομένων στη Βάση Δεδομένων

Θα δούμε τώρα πώς μπορούμε να δώσουμε τη δυνατότητα στους επισκέπτες (χρήστες) του site μας να προσθέσουν τις δικές τους εγγραφές (γραμμές) στη βάση δεδομένων. Για να μπορέσει να γίνει κάτι τέτοιο, είναι απαραίτητη η χρήση μιας φόρμας (form), ως εξής :

```
<form action="update_user.php" method="post">
  <table cellpadding="0" cellspacing="0" class="border"
align="center">
  <tr>
    <th>όνομα χρήση:</th>
    <td><input type="text" name="u_name"></td>
  </tr>
  <tr>
    <th>κωδικός:</th>
    <td><input type="password" name="pass"></td>
  </tr>
  <tr>
    <th>Όνομα/μο:</th>
    <td><input type="text" name="rname" value="<?php
echo stripslashes($realname)?>"></td>
  </tr>
```

Για να καταχωρηθούν τα δεδομένα στη βάση δεδομένων, πρέπει να χρησιμοποιήσουμε τη συνάρτηση `mysql_query()` για να εκτελέσουμε ένα ερώτημα INSERT, χρησιμοποιώντας τη μεταβλητή `$uname` για την τιμή που θα υποβληθεί, ως εξής :

```
$query = "INSERT INTO useres VALUES ('', '$uname', '$pass', '$rname',
')";

    if(mysql_query($query))
        echo "Success";

    else
        die(mysql_error());
```


3 Απαιτήσεις του Συστήματος

Η συγκεκριμένη εφαρμογή που παρουσιάζουμε στην παρούσα πτυχιακή εργασία καλείται να καλύψει την ανάγκη αποθήκευσης και ταξινόμησης των ταξιδιών των εργαζόμενων μιας εταιρίας. Πιο συγκεκριμένα η εφαρμογή που σχεδιάσαμε , αναπτύξαμε και υλοποιήσαμε αφορά το Ερευνητικό Ακαδημαϊκό Ινστιτούτο Τεχνολογίας Υπολογιστών(Ε.Α.Ι.Τ.Υ).

Στα πλαίσια μηχανογράφησης της εταιρίας υπήρχε ανάγκη για αποθήκευση όλων των πληροφοριών που σχετίζονταν με τα ταξίδια των εργαζομένων και την οικονομική αποζημίωση τους γιαυτα.

Τέτοιες πληροφορίες περιλαμβάνουν τις ημερομηνίες αναχώρησης και επιστροφής, την αιτιολογία, το κόστος της ημερήσιας αποζημίωσης και άλλες πληροφορίες που θα αναφερθούν εκτενέστερα παρακάτω.

3.1 Απαιτήσεις της εταιρίας

Παρακάτω φαίνονται οι συγκεκριμένες απαιτήσεις του Ε.Α.Ι.Τ.Υ που αφορούν στις κατηγορίες των χρηστών ,τις πληροφορίες που απαιτούνται από το τμήμα μηχανογράφησης καθώς και τις διαδικασίες απόδοσης της χρηματικής αμοιβής στον εργαζόμενο.

3.1.1 Κατηγορίες χρηστών του συστήματος

Οι χρήστες του συστήματος διακρίνονται ως προς τα στοιχεία που αφορούν την εφαρμογή μας σε τρεις κατηγορίες:

- Προπτυχιακοί
- Μεταπτυχιακοί
- Καθηγητές

Επιπλέον για τη σωστή διαχείριση του συστήματος ορίζεται μια νέα κατηγορία χρηστών , ο Administrator.

3.1.2 Πληροφορίες για κάθε ταξίδι

Το τμήμα μηχανογράφησης επιβάλλει για κάθε εργαζόμενο να κρατιέται αρχείο των ταξιδιών του. Για κάθε ταξίδι οι πληροφορίες που μας ενδιαφέρουν είναι οι εξής:

- Όνομα έργου
- Αιτιολογία
- Ημερομηνία αναχώρησης και επιστροφής
- Έξοδα διαβίωσης
- Εισιτήρια
- Ξενοδοχείο
- Προκαταβολή
- Διάφορα έξοδα
- Registration Fees
- Παρατηρήσεις
- Ποσοστό χρέωσης από το E.A.I.T.Y

3.1.3 Διαδικασία απόδοσης χρηματικής αμοιβής

Το τμήμα μηχανογράφησης άπαιτη για την απόδοση της χρηματικής αμοιβής στον εργαζόμενο την προσκόμιση εγγράφου με συγκεκριμένη μορφή όπου θα παρουσιάζονται όλες οι παραπάνω πληροφορίες που αφορούν κάποιο ταξίδι. Ενδεικτική μορφή αυτού του εγγράφου φαίνεται παρακάτω.

ΦΥΛΛΟ ΕΓΚΡΙΣΗΣ ΔΑΠΑΝΩΝ ΤΑΞΙΔΙΩΝ ΗΜΕΡ/ΝΙΑ:18/03/06

ΣΥΝΕΡΓΑΤΗΣ: Σοφία Παππά				
ΕΡΓΟ: E-Learn 2006			ΠΟΣΟΣΤΟ ΧΡΕΩΣΗΣ ΕΡΓΟΥ:	
ΑΙΤΙΟΛΟΓΙΑ: 3rd IEEE International Conference on Advanced Learning Technologies				
ΗΜΕΡΟΜΗΝΙΑ ΑΝΑΧΩΡΗΣΗΣ: 17 Μαΐου 2006		ΗΜΕΡΟΜΗΝΙΑ ΕΠΙΣΤΡΟΦΗΣ: 25 Μαΐου 2006		
ΚΑΤΗΓΟΡΙΑ ΔΑΠΑΝΩΝ	ΠΛΗΡΩΤΕΑ ΣΤΟΝ ΣΥΝΕΡΓΑΤΗ(1)	ΠΛΗΡΩΤΕΑ ΑΠΟ ΤΟ Ε.Α.Ι.Τ.Υ(2)	ΚΩΔΙΚΟΣ ΕΡΓΟΥ(*)	ΚΩΔΙΚΟΣ ΑΙΤΗΜΑΤΟΣ ΕΓΓΡΑΦΗΣ(*)
ΕΙΣΙΤΗΡΙΑ		400 €		
ΞΕΝΟΔΟΧΕΙΟ	584 €			
ΕΞΟΔΑ ΔΙΑΒΙΩΣΗΣ 2*22 =44€ 3*44=132 € 4*66 =264 €	440€			
Registration Fees				
ΔΙΑΦΟΡΑ ΕΞΟΔΑ αυτοκίνητο:40€ Διόδια:12€	52€			
ΣΥΝΟΛΟ	1076€	400 €		
ΓΕΝΙΚΟ ΣΥΝΟΛΟ(1)+(2)	1476€			
			ΕΓΓΡΑΦΕΣ ΓΙΑ ΠΛΗΡΩΤΕΑ ΣΤΟ ΣΥΝΕΡΓΑΤΗ	
ΠΡΟΚΑΤΑΒΟΛΗ	200 €			
ΥΠΟΛΟΙΠΟ ΠΡΟΣ ΠΛΗΡΩΜΗ	1276€			
ΥΠΟΓΡΑΦΗ ΣΥΝΕΡΓΑΤΗ		ΕΓΚΡΙΣΗ ΥΠΕΡΓΟΥ	ΕΓΚΡΙΣΗ & ΕΛΕΓΧΟΣ ΔΙΑΧΕΙΡΙΣΤΗ	ΕΓΚΡΙΣΗ Δ.Δ.Υ.Ο
ΠΑΡΑΤΗΡΗΣΕΙΣ:				

(*): ΣΥΜΠΛΗΡΩΝΕΤΑΙ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ

4 Σχεδιασμός του συστήματος

Στο παρόν κεφάλαιο θα περιγραφούν ζητήματα που προέκυψαν κατά τη διάρκεια κατασκευής της εφαρμογής, αλλά και κομμάτια του κώδικα τα οποία παίζουν σημαντικό ρόλο στην καλή λειτουργία του συστήματος.

4.1 Η Ανάγκη για ένα Database-Driven Web Site

Σήμερα στο Web, το *περιεχόμενο (content)* είναι αυτό που κυριαρχεί. Αφού έχετε μάθει πολύ καλά την HTML και μερικά κόλπα στην JavaScript και τη Δυναμική (Dynamic) HTML, μπορείτε να δημιουργήσετε ένα Web site με εντυπωσιακή εμφάνιση. Έρχεται, όμως, η ώρα που πρέπει να γεμίσετε τη σελίδα σας με *πραγματικές πληροφορίες (real information)*. Όποιο site τραβάει το ενδιαφέρον των επισκεπτών σας επανειλημμένα, θα πρέπει να περιέχει καινούργιο και συνεχώς ανανεούμενο περιεχόμενο.

Το πρόβλημα είναι ότι συχνά, οι άνθρωποι που παρέχουν το περιεχόμενο για ένα site δεν είναι οι ίδιοι μ' αυτούς που κάνουν και τη σχεδίασή του (design). Αλλά είναι σύνηθες, ο παροχέας του περιεχομένου να μην γνωρίζει καθόλου από HTML. Πώς, όμως, μπορούμε να πάρουμε τότε το περιεχόμενο από τον παροχέα και να το εγκαταστήσουμε στο Web site;

Δεν μπορούν φυσικά όλες οι εταιρείες να απασχολούν πλήρως έναν Webmaster και οι περισσότεροι από τους Webmasters έχουν άλλες δουλειές

να κάνουν από το να αντιγράψουν αρχεία του Word μέσα σε tags της HTML. Η λύση σ' αυτό το πρόβλημα είναι το *database-driven site design*. Πετυχαίνοντας τον πλήρη διαχωρισμό ανάμεσα στη σχεδίαση του site και το περιεχόμενο που θέλουμε να παρουσιάσουμε μέσα σ' αυτό, μπορούμε να δουλέψουμε με το ένα χωρίς να μας απασχολεί το άλλο.

Αντί να δημιουργούμε ένα HTML αρχείο για την κάθε σελίδα του site, χρειάζεται μόνο να δημιουργήσουμε μια σελίδα για το κάθε είδος πληροφορίας που θέλουμε να παρουσιάσουμε. Αντί να επικολλάμε συνέχεια νέο περιεχόμενο στις σελίδες μας, δημιουργούμε ένα απλό *σύστημα διαχείρισης περιεχομένου (content management system)* που δίνει τη δυνατότητα στους συγγραφείς να δημοσιεύουν (post) το καινούργιο περιεχόμενό τους οι ίδιοι χωρίς να κάνουν καθόλου χρήση της HTML.

4.2 Ξεκίνημα με το Dreamweaver MX

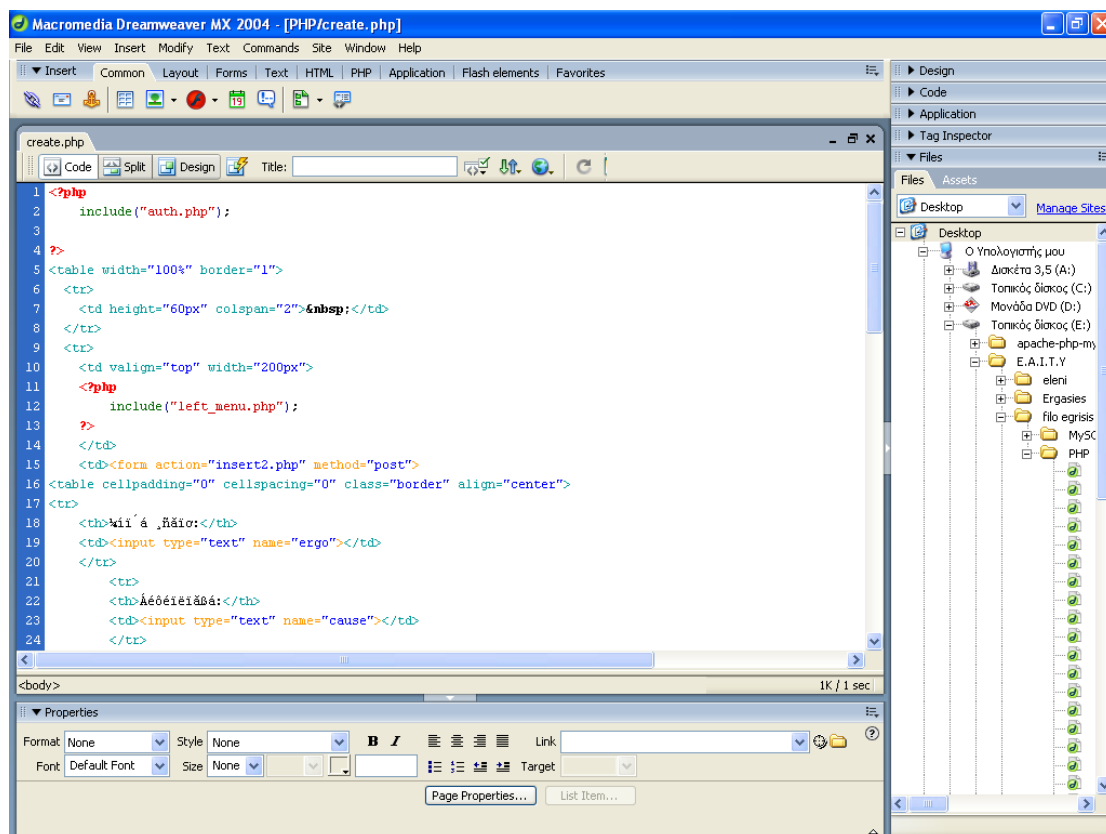
Τώρα που έχουμε έναν web server που εκτελεί την PHP και έναν MySQL database server, είναι καιρός να ασχοληθούμε με το κυρίως θέμα μας που είναι ο σχεδιασμός PHP ιστοσελίδων με το Dreamweaver MX. Θα ξεκινήσουμε δίνοντας στο Dreamweaver MX όλες τις απαραίτητες πληροφορίες που θα χρειασθεί για να μπορέσει να συνεργασθεί με το περιβάλλον ανάπτυξης, δημιουργώντας έναν ορισμό του δικτυακού τόπου (Site Definition).

Αφού θα έχει ορισθεί το site, θα δούμε πώς μπορούμε να δημιουργήσουμε μια σύνδεση με τη βάση δεδομένων (database connection) μέσα στο Dreamweaver, ώστε να μπορούμε να έχουμε πρόσβαση στη βάση δεδομένων της MySQL που έχουμε δημιουργήσει νωρίτερα. Ο ορισμός ενός δικτυακού τόπου (site definition) είναι το βασικό σημείο αρχής κατά την ανάπτυξη ενός οποιουδήποτε web site. Με τις πληροφορίες που θα παρέχονται εδώ, το Dreamweaver θα είναι σε θέση να εντοπίζει τη δομή των φακέλων (directory structure) του site, να παρακολουθεί όλους τους

εσωτερικούς και εξωτερικούς υπερσυνδέσμους (hyperlinks) και να διατηρεί μια λίστα όλων των εικόνων, των χρωμάτων και των εξωτερικών αρχείων.

4.2.1 Εισαγωγή στο Dreamweaver MX

Το Dreamweaver MX έχει ενσωματώσει τα χαρακτηριστικά και τις δυνατότητες των προγραμμάτων (πακέτων) Dreamweaver, UltraDev και HomeSite σ' ένα μόνο εργαλείο. Το Dreamweaver MX διαθέτει επίσης ενισχυμένη υποστήριξη για συστήματα server, όπως είναι τα PHP, ASP.NET και ColdFusion MX. Για πρώτη φορά το Dreamweaver MX μάς δίνει τη δυνατότητα να επιλέξουμε το στυλ και τη διάταξη (layout) του User Interface (διασύνδεση χρήστη). Όταν ξεκινάμε το Dreamweaver MX για πρώτη φορά, θα εμφανισθεί ένα πλαίσιο διαλόγου όπου θα μπορούμε να καθορίσουμε το στυλ του User Interface που θέλουμε να χρησιμοποιήσουμε.



Εικόνα 5. Επιφάνεια εργασίας του Dreamweaver

Το *Insert panel* αποτελείται από πολλές καρτέλες, όπου η κάθε καρτέλα περιέχει ένα διαφορετικό σύνολο από εργαλεία. Μερικές απ' αυτές τις

καρτέλες αλλάζουν δυναμικά ανάλογα με το είδος του εγγράφου με το οποίο εργαζόμαστε. Για παράδειγμα, αν δημιουργήσουμε μια PHP σελίδα θα εμφανισθεί μια καρτέλα με τον τίτλο PHP, η οποία θα περιέχει κάποια εργαλεία χρήσιμα όταν δουλεύουμε σε πρωτογενή κώδικα (source code). Παρόμοια, αν δουλεύουμε με μια σελίδα ColdFusion, θα εμφανισθούν κάποιες άλλες καρτέλες και η καρτέλα της PHP θα εξαφανισθεί καθώς δεν θα έχει νόημα η παρουσία της. Αυτό το είδος των αλλαγών αποκαλείται *context sensitivity* (ευαισθησία στο κείμενο) και είναι ειδικά σχεδιασμένο ώστε να μας βοηθάει να βρίσκουμε τα εργαλεία που χρειαζόμαστε, όταν τα χρειαζόμαστε.

Από το *Design panel* έχουμε εύκολη πρόσβαση στα στυλ CSS (Cascading Style Sheet) και στα στυλ της HTML που χρησιμοποιούμε. Επίσης, αν και δεν αποτελεί αυστηρά ένα στοιχείο του design, υπάρχει και η καρτέλα *Behaviors* από την οποία μπορούμε να έχουμε εύκολη πρόσβαση σε μια σειρά από ενσωματωμένες *συμπεριφορές (behaviors)* της JavaScript.

Οι συμπεριφορές αυτές είναι μικρά κομμάτια κώδικα με τα οποία μπορούμε να προσθέσουμε ενδιαφέροντα και ζωντανά εφέ στις ιστοσελίδες μας, όπως από το να εμφανίζουμε παράθυρα φυλλομετρητή όταν γίνεται κλικ σε κάποιον σύνδεσμο (link) έως και να ελέγχουμε διάφορα animations χρησιμοποιώντας το timeline του Dreamweaver.

Το *Code panel* περιέχει χρήσιμα εργαλεία και επιλογές. Από εδώ μπορούμε να δούμε ολόκληρη τη δομή της σελίδας μας χρησιμοποιώντας τον Tag Inspector, ο οποίος εμφανίζει τη σελίδα με τη μορφή ενός δένδρου και μπορούμε να επέμβουμε απευθείας στα διάφορα tags χωρίς να είμαστε αναγκασμένοι να πάμε σε κατάσταση πρωτογενούς κώδικα (Code view).

Με την καρτέλα *Snippets* μπορούμε εύκολα να αποθηκεύσουμε και να διατηρήσουμε κομμάτια κώδικα που ίσως χρειασθεί να ξαναχρησιμοποιήσουμε σ' άλλες σελίδες ή και σ' άλλα web sites. Από έτοιμα meta tags έως και συναρτήσεις της JavaScript που υπολογίζουν την περιφέρεια ενός κύκλου, υπάρχει πάντα κάτι για τον καθένα. Η καρτέλα *Reference* περιέχει σύντομους οδηγούς αναφοράς γι' όλες τις υποστηριζόμενες γλώσσες server.

Το *Application panel* είναι αυτό στο οποίο θα ξοδέψουμε τον περισσότερο χρόνο μας όταν θα δημιουργήσουμε τις αλληλεπιδράσεις στην πλευρά του server (server-side interactions) του δικτυακού μας τόπου (web site). Αυτό το panel περιέχει τέσσερις καρτέλες (tabs) : Databases, Bindings, Server Behaviors και Components. Από την καρτέλα *Databases* μπορούμε να δούμε τη δομή (structure) και τη διάταξη (layout) όποιας βάσης δεδομένων έχουμε δημιουργήσει στα web sites μας. Από εδώ μπορούμε επίσης να προσθέσουμε καινούργιες συνδέσεις με βάσεις δεδομένων και να τις κάνουμε άμεσα διαθέσιμες.

Όταν ανακτούμε δυναμικά δεδομένα για την ιστοσελίδα μας, θα πρέπει να δημιουργήσουμε αυτό που είναι γνωστό ως *Data Binding*, το οποίο παρέχει τον ουσιαστικό σύνδεσμο ανάμεσα στο περιβάλλον ανάπτυξης του Dreamweaver και στα δεδομένα μας. Από την καρτέλα αυτή μπορούμε να δημιουργήσουμε πολλά διαφορετικά Data Bindings για να ανακτούμε πληροφορίες από βάσεις δεδομένων, πεδία φόρμας, cookies και πολλά άλλα.

Στο Application panel έχουμε πρόσβαση και σε επιπλέον συμπεριφορές (behaviors), οι οποίες όμως δεν προσθέτουν κώδικα JavaScript στις ιστοσελίδες μας. Εισάγουν server-side code για την τελική αλληλεπίδραση (interaction) ανάμεσα στο web site και τον χρήστη (επισκέπτη). Αυτή η καρτέλα (Server Behaviors) είναι context-sensitive, πράγμα που σημαίνει ότι αλλάζει ανάλογα με το server model που χρησιμοποιούμε, όπως για παράδειγμα PHP ή CFML κοκ.

Η καρτέλα *Components* δεν χρησιμοποιείται όταν εργαζόμαστε με PHP ιστοσελίδες αλλά προορίζεται ειδικά για άλλα server models, όπως είναι το ASP.

Από το *Files panel* και την καρτέλα Site έχουμε πλήρη πρόσβαση στο τοπικό (local) και στο απομακρυσμένο (remote) σύστημα αρχείων. Μπορούμε να μεταφέρουμε αρχεία και φακέλους (καταλόγους) προς και από τον κατάλογο εργασίας μας χωρίς την ανάγκη ύπαρξης ενός ξεχωριστού προγράμματος FTP (File Transfer Protocol).

Η καρτέλα *Assets* καταγράφει όλα τα εξωτερικά αρχεία, συνδέσμους (links) και χρώματα που χρησιμοποιούνται όταν δημιουργούμε το site μας, τα οποία είναι διαθέσιμα για εύκολη αναφορά, τροποποίηση και επαναχρησιμοποίηση.

Από το *Answers panel* μπορούμε να έχουμε άμεση πρόσβαση σε tutorials και άλλα χρήσιμα βοηθήματα. Αν κάνουμε κλικ στο πλήκτρο Update μπορούμε να συνδεθούμε με το site της Macromedia και να κατεβάσουμε ό,τι καινούργια στοιχεία υπάρχουν.

Από το panel *Properties* μπορούμε να ελέγχουμε απευθείας ειδικά στοιχεία των HTML tags και των σελίδων μας χωρίς να χρειασθεί να πάμε σε πηγαίο κώδικα. Αυτό το panel αλλάζει (προσαρμόζεται) συνέχεια ανάλογα με το στοιχείο που είναι επιλεγμένο στη σελίδα.

4.3 CSS (Cascade Style Sheet)

Στο παρελθόν η εμφάνιση των σελίδων στο Web εξαρτιόταν από τον browser που χρησιμοποιούσε ο κάθε χρήστης που τις επισκεπτόταν. Με την εισαγωγή των CSS (Cascading Style Sheets) και των CSS-P (Cascading Style Sheets Positioning) έχουμε πλήρη έλεγχο πάνω στην εμφάνιση των σελίδων μας. Επίσης, μας επιτρέπουν να ξεχωρίσουμε το περιεχόμενο των σελίδων από το σχεδιασμό. Μ' αυτόν τον τρόπο διευκολύνονται οι αλλαγές και οι επανασχεδιασμοί στις σελίδες.

Τα CSS υποστηρίζονται από τον Microsoft Internet Explorer browser, έκδοση 4 και πάνω και Netscape Navigator, έκδοση 4 και πάνω. Υπάρχουν παρόλα αυτά μερικές διαφορές στα CSS που υποστηρίζει ο κάθε browser.

Υπάρχουν τρεις διαφορετικοί τύποι Cascading Style Sheets. Έχουμε τα embedded style sheets, τα inline style sheets και τα linked style sheets. Είναι ευκολότερο να κατανοήσουμε τα βασικά των CSS παρουσιάζοντας τα embedded style sheets γι' αυτό και αυτό θα κάνουμε στη συνέχεια.

4.3.1 Embedded Style Sheets

Όλα τα embedded style sheets χρησιμοποιούν το ίδιο format και τοποθετούνται στην περιοχή <HEAD> των σελίδων σας. (αν και οι

περισσότεροι browsers θα σας επιτρέψουν να τα τοποθετήσετε μέσα στο <BODY> των σελίδων σας). Στη συνέχεια βλέπουμε το format που χρησιμοποιείται:

```
<STYLE TYPE="text/css">
<!--
TAG {property: value;}
-->
</STYLE>
```

Μέσα στο <STYLE> tag, υπάρχει μια λίστα από HTML tags που ακολουθούνται από style sheet properties που τα ορίζουν. Το παραπάνω format δείχνει ένα tag που προσδιορίζεται από μια CSS property (ιδιότητα). Μπορείτε να βάλετε έναν απεριόριστο αριθμό από tags και το καθένα να προσδιορίζεται από ένα απεριόριστο αριθμό από CSS properties. Όλες οι CSS properties πρέπει να περικλείονται σε αγκύλες ({}), και αν χρησιμοποιούμε πολλές properties πρέπει να τις διαχωρίζουμε με ένα ελληνικό ερωτηματικό (;).

Στη συνέχεια βλέπουμε ένα απλό παράδειγμα ενός embedded style sheet:

```
<STYLE TYPE="text/css">
<!--
BODY {font-family: Verdana, Arial, Helvetica;}
-->
</STYLE>
```

Σ' αυτό το παράδειγμα καθορίζουμε την font family για το κυρίως κείμενο. Ολόκληρο το κείμενο στην περιοχή του <BODY> θα χρησιμοποιεί τον τύπο font, font face, Verdana (ή Arial ή Helvetica αν η Verdana δεν είναι διαθέσιμη στο σύστημα του χρήστη).

Ερώτηση: το κείμενο μέσα στα <P> tags θα επηρεαστεί? στα <DIV>? στα <TD>?

Απάντηση: Το κείμενο μέσα στα <P> και <DIV> tags θα επηρεαστεί (εκτός και αν έχουν οριστεί αυτά τα tags να έχουν άλλες ιδιότητες) επειδή

κληρονομούν το font face από το body. Το κείμενο μέσα σε ένα κελί πίνακα δεν κληρονομεί τα χαρακτηριστικά που έχουμε καθορίσει στο body.

4.3.2 Classes

Αν καθορίσουμε τις επικεφαλίδες, H1 headers, να είναι μπλέ, τότε όλες οι H1 headers θα έχουν αυτή τη μορφή. Υπάρχει όμως η περίπτωση να θέλουμε μερικές επικεφαλίδες να έχουν διαφορετικό χρώμα (ή κάποια άλλη ιδιότητα). Οι Classes μας δίνουν αυτή τη δυνατότητα..

Το παρακάτω παράδειγμα δείχνει πώς χρησιμοποιούνται οι classes:

```
<STYLE TYPE="text/css">
<!--
H1.blue {color: blue;}
-->
</STYLE>
```

Αν βάλετε το παραπάνω style sheet μέσα στο HTML αρχείο, δε θα δείτε τα H1 headers να γίνονται μπλε. Για να γίνει αυτό θα πρέπει "πείτε" συγκεκριμένα στον header να υιοθετήσει αυτό το style. Αυτό μπορεί να συμβεί αν τοποθετήσουμε την CLASS attribute μέσα σε ένα H1 tag ως εξής:

```
<H1 CLASS="blue">This header is blue</H1>
```

Παρόλα αυτά ένα H2 tag με την μπλε class δε θα είναι μπλε, αφού στο style sheet η μπλε class ανατέθηκε στο <H1> tag. Για να χρησιμοποιήσουμε την τάξη μπλε , .blue, σε κάθε tag γράφουμε τα εξής:

```
<STYLE TYPE="text/css">
<!--
.blue {color: blue;}
-->
</STYLE>
```

Παράδειγμα

```

<STYLE TYPE="text/css">
<!--

BODY {font-family: Times New Roman, Serif; color: black;}

H1 {font-family: Verdana, Arial, Sans-serif; font-size: 140%; color: maroon;}

H2 {font-family: Verdana, Arial, Sans-serif; font-size: 110%; color: maroon;}

TABLE {border: 3px solid black; padding: 2px;}

TD.head {background-color: #FFFC;}

.verdana {font-family: Verdana, Arial, Sans-serif;}

-->
</STYLE>

```

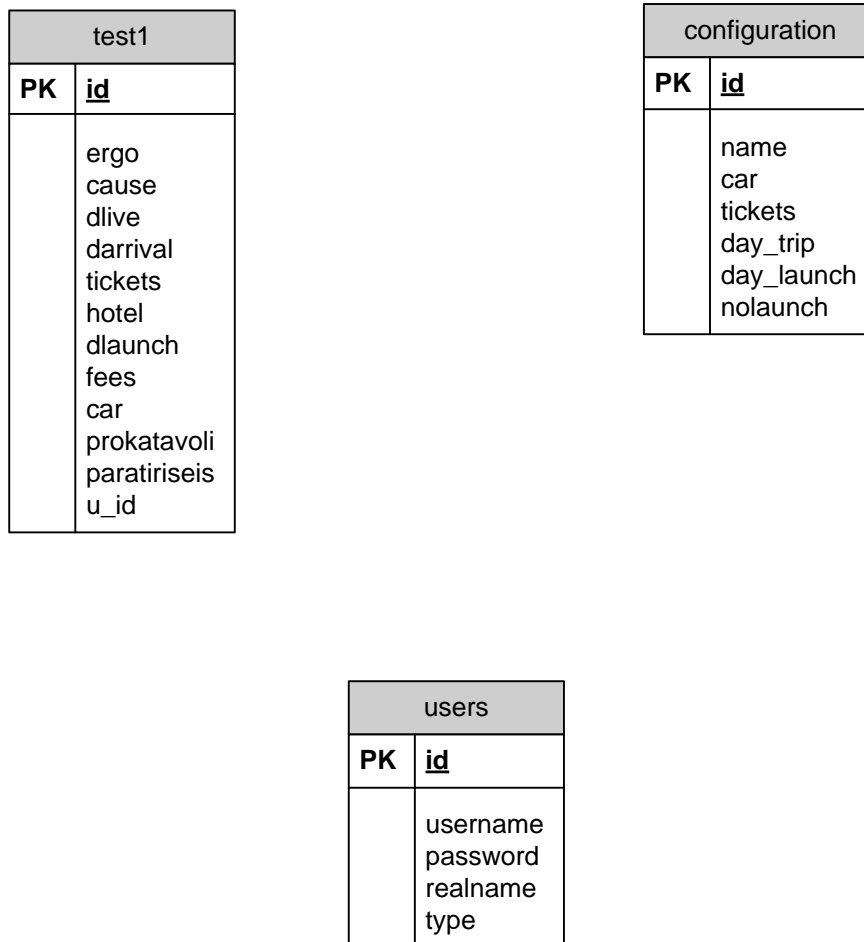
Στο style sheet αυτό καθορίζουμε το font face να είναι Times New Roman (και ένα δεύτερο τύπο font να είναι το serif). Το χρώμα μαύρο καθορίζεται επίσης.

Στη συνέχεια καθορίζονται τα H1 και H2 headers χρησιμοποιώντας Verdana, Arial, Sans-serif ως font face και maroon-καφέ για χρώμα. Τα H1 headers θα είναι το 140% του μεγέθους των κανονικών κειμένων του body και τα H2 headers θα είναι το 110% του μεγέθους σε σχέση με το κείμενο του body.

Οι πίνακες της σελίδας θα έχουν μαύρο, 3 pixel μόνο περίγραμμα και ένα 2 pixel padding. Τα κελιά των πινάκων στα οποία χρησιμοποιούμε την head class θα έχουμε χρώμα στο background το #FFFC (δηλ. ανοιχτό κίτρινο). Τέλος, η τάξη verdana καθορίζεται χωρίς tag έτσι ώστε να εφαρμόζεται με κάθε tag, όταν θέλουμε το κείμενο να έχει αυτή τη γραμματοσειρά.

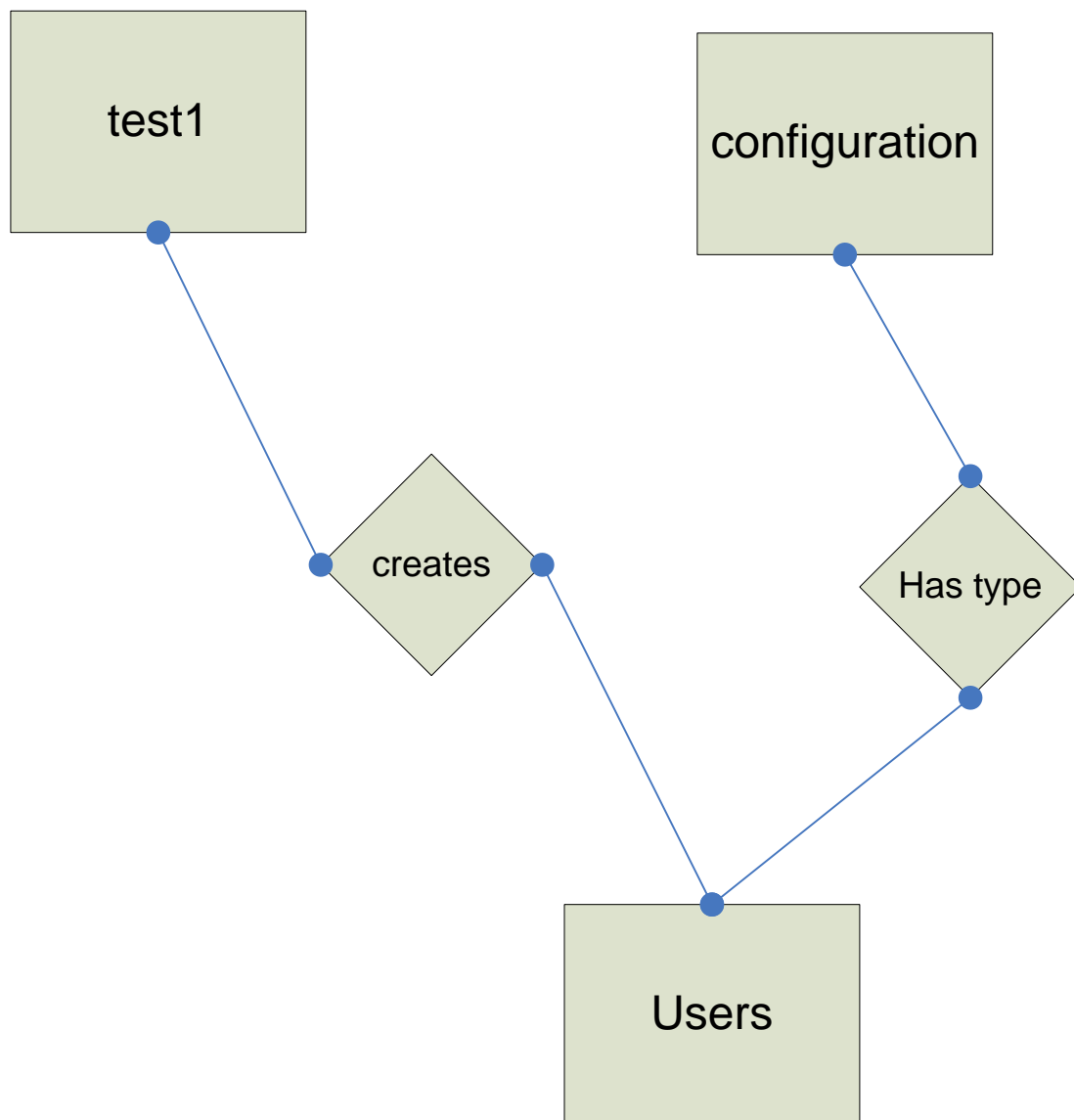
4.4 Η βάση δεδομένων και ανάλυση του κώδικα

Η βάση δεδομένων της εφαρμογής μας αποτελείται από 3 πίνακες. Οι πίνακες φαίνεται και στο παρακάτω διάγραμμα.



Εικόνα 6 . Σχεδιάγραμμα της Βάσης Δεδομένων

Οι συσχετίσεις μεταξύ των οντοτήτων της βάσης δεδομένων μας απεικονίζονται στο παρακάτω διάγραμμα οντοτήτων συσχετίσεων (ER διάγραμμα).



Εικόνα 7 . Διάγραμμα ER

4.4.1 Ανάλυση των πινάκων της βάσης δεδομένων

Στην ενότητα αυτή θα γίνει ανάλυση όλων των πινάκων που απαρτίζουν τη βάση δεδομένων και θα αναλυθούν τα πεδία που απαρτίζουν τον κάθε πίνακα και ο σκοπός ύπαρξής τους.

4.4.1.1 Πίνακας users

Ο πίνακας αυτός περιέχει τα στοιχεία για τους χρήστες της εφαρμογής. Περιέχει τα εξής πεδία:

- **Id(integer[11]):**Πρόκειται για το αναγνωριστικό του πίνακα και είναι το κλειδί του συγκεκριμένου πίνακα.
- **Username(varchar[200]) :**Περιγράφει το όνομα του χρήστη.
- **Password(varchar[128]):**:Ο κωδικός του χρήστη που θα χρησιμοποιεί για το διαχειριστικό μέρος της εφαρμογής.
- **Realname(varchar[200]):**Το πλήρες όνομα του χρήστη
- **Type(integer[4]):**Τον τύπο του χρήστη.

4.4.1.2 Πίνακας configuration

Ο πίνακας αυτός περιέχει στοιχεία που είναι χρήσιμα για τον υπολογισμό μέρος των εξόδων του ταξιδιού ανάλογα με την ιδιότητα του χρήστη.

- **Id(integer[4]):**Αναγνωριστικό του πίνακα και ξένο κλειδί του πίνακα users.
- **Name(varchar[200]):**Ιδιότητα /χαρακτηριστικό του χρήστη
- **Car(tinyint[3]):**έξοδα αυτοκίνητου(βενζίνη)
- **tickets(double):**Αεροπορικά εισιτηρια.
- **day_trip(int[11]):**κόστος ημέρας ταξιδιού
- **day_launch(int[11]):**κόστος με φαγητό
- **nolaunch(int[11]):**κόστος χωρίς φαγητό

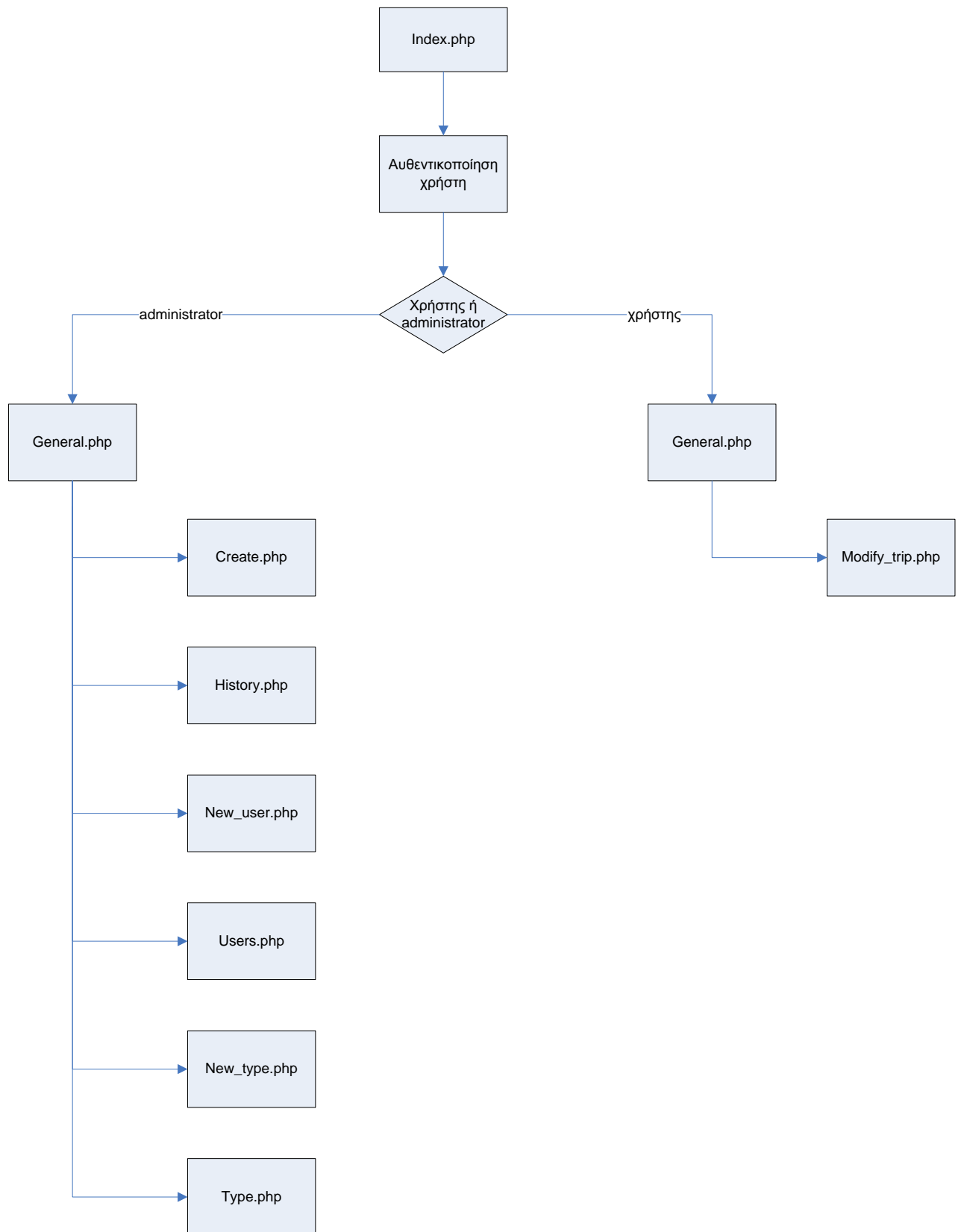
4.4.1.3 Πίνακας test1

Ο πίνακας αυτός περιέχει 13 πεδία τα οποία δηλώνουν πληροφορίες που θα μας χρησιμέψουν στο να υπολογίσουμε τα έξοδα του ταξιδιού και να πάρουμε πληροφορίες σχετικά με αυτό. Από αυτόν τον πίνακα προκύπτουν τα περισσότερα στοιχεία που εμφανίζονται στο τελικό έγγραφο υποβολής.

- `id(integer[11])`:Αναγνωριστικό του πίνακα και ξένο κλειδί του πίνακα `configuration`
- `ergo(varchar[200])`:Όνομα έργου
- `cause(varchar[200])`:Αιτιολογία έργου
- `dlive(date)`:Ημερομηνία άφιξης
- `darrival(date)`:Ημερομηνία επιστροφής
- `tickets(double)`:Αεροπορικά εισιτήρια
- `hotel(double)`:κόστος ξενοδοχείου
- `dlaunch(integer[11])`:Ημέρες που έχει γεύμα στο συνέδριο
- `fees(double)`:Προηγούμενη κράτηση του χρήστη
- `car(tinyint[3])`:Επιλογή αν ο χρήστης θα μετακινηθεί με αυτοκίνητο.
- `prokatavoli(double)`:Προκαταβολή για τα έξοδα ταξιδιού
- `paratiriseis(varchar[254])`:Παρατηρήσεις
- `u_id(integer[11])`:`session id`(δημιουργείτε όταν ο χρήστης εισέλθει με τα δικά του χαρακτηριστικά)

4.4 Περιγραφή του συστήματος

Σε αυτήν την ενότητα θα περιγράψουμε τις λειτουργίες κάθε σελίδας και τις δυνατότητες που έχει κάθε μια από αυτές. Παρακάτω φαίνεται το διάγραμμα ροής πληροφορίας του συστήματος .

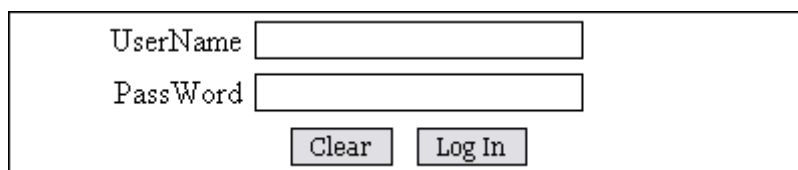


Εικόνα 8. Διάγραμμα ροής πληροφορίας

4.5.1 Αρχική σελίδα εφαρμογής

Παρακάτω βλέπουμε την αρχική σελίδα της εφαρμογής μας. Εδώ ο χρήστης πρέπει να εισάγει ένα `username` και ένα `password` για να μπορέσει να διαχειριστεί την εφαρμογή. Οι χρήστες που θα μπορούν να διαχειριστούν την εφαρμογή χωρίζονται σε δυο κατηγορίες τον διαχειριστή και τους απλούς χρήστες οι οποίοι με την σειρά τους χωρίζονται στους προπτυχιακούς, τους μεταπτυχιακούς και τους καθηγητές.

Ο διαχειριστής είναι υπεύθυνος για την εφαρμογή, μπορεί δηλαδή να προσθέσει χρηστές στο σύστημα, να αλλάξει τον τρόπο υπολογισμού των εξόδων του ταξιδιού και να τροποποιήσει τις χρεώσεις που ισχύουν για τον κάθε χρήστη. Οι χρήστες εισάγουν τις πληροφορίες που χρειάζονται για να συμπληρωθεί ο πίνακας `test1` της βάσης μας.



The image shows a login form with the following elements:

- A text label "UserName" followed by a text input field.
- A text label "PassWord" followed by a text input field.
- Two buttons: "Clear" and "Log In", positioned below the input fields.

Εικόνα 9. Αρχική σελίδα εφαρμογής

Εδώ ο χρήστης συμπληρώνει τα στοιχεία του για να εισέλθει στο σύστημα. Αφού γίνει η αναγνώριση του χρήστη(διαχειριστής ή απλός χρήστης) ,εμφανίζετε η ανάλογη σελίδα με τις δυνατότητες και τα δικαιώματα του καθενός.

Θα εξετάσουμε και τις δυο περιπτώσεις ξεκινώντας από αυτή του διαχειριστή.

Η πρώτη σελίδα που βλέπει ο διαχειριστής είναι η παρακάτω:

Κεντρική σελίδα Δημιουργία Ιστορικό Δημιουργία χρήστη Αλλαγή χρήστη Δημιουργία configuration Αλλαγή configuration	<p>Ταξίδια που πρόκειται να γίνουν</p> <p>χρήστης: Eleni Anagnostopoulou Αποσύνδεση</p>

Εικόνα 10. Σελίδα Διαχειριστή

Στην αριστερή περιοχή βλέπουμε τις δυνατότητες και τις ενέργειες που μπορεί να κάνει, όπως το να δημιουργήσει καινούργιο χρήστη και καινούργιο configuration καθώς επίσης και να αλλάξει τα παραπάνω.

Στο κέντρο της σελίδας ο χρήστης μπορεί να δει μια λίστα με τους τίτλους από τα μελλοντικά ταξίδια καθώς επίσης και την δυνατότητα να τροποποιήσει πληροφορίες για αυτά.

Κεντρική σελίδα Δημιουργία Ιστορικό	<p>Ταξίδια που πρόκειται να γίνουν</p> <p>χρήστης: Σοφία Παππά Αποσύνδεση</p> <p>1 - 3rd IEEE International Conference on Advanced Learning Technologies - 17 Μαΐου 2006 modify</p>

Εικόνα 11. Σελίδα μελλοντικών ταξιδιών

4.5.2 Δημιουργία

Σε αυτήν την σελίδα ο διαχειριστής μπορεί να εισάγει τα δεδομένα που χρειάζονται για την δημιουργία μιας νέας εγγραφής ταξιδιού.

Τα στοιχεία που πρέπει να συμπληρώσει ο διαχειριστής φαίνονται παρακάτω:

Κεντρική σελίδα Δημιουργία Ιστορικό Δημιουργία χρήστη Αλλαγή χρήστη Δημιουργία configuration Αλλαγή configuration	<p>Όνομα Έργου: <input type="text"/></p> <p>Αιτιολογία: <input type="text"/></p> <p>Ημέρια Αναχώρησης: <input type="text"/> Η ημερομηνία θα πρέπει να εισάγετε ως εξής: yyyy-mm-dd</p> <p>Ημέρια Επιστροφής: <input type="text"/></p> <p>Εισιτήρια: <input type="text"/></p> <p>Τιμή Βραδιάς: <input type="text"/></p> <p>Πόσες μέρες έχει γένμα: <input type="text"/></p> <p>Registration: <input type="text"/></p> <p>Διάφορα έξοδα: <input type="checkbox"/> αυτοκίνητο</p> <p>Προκαταβολή: <input type="text"/></p> <p>Παρατηρήσεις: <input style="width: 100%; height: 40px;" type="text"/></p> <p style="text-align: right;"> <input type="button" value="Υποβολή"/> <input type="button" value="Ανάρεση"/> </p>
---	---

Εικόνα 12. Εισαγωγή νέας εγγραφής

Αυτά τα στοιχεία θα αποθηκευτούν στον πίνακα test1.

4.5.3 Ιστορικό

Η συγκεκριμένη επιλογή εμφανίζει πληροφορίες για όλα τα ταξίδια που έχει πραγματοποιήσει ο χρήστης. Αρχικά εμφανίζετε μια λίστα με τους τίτλους όλων των ταξιδιών που έχει πραγματοποιήσει ο χρήστης και αφού επιλέξει ένα από αυτά το σύστημα του εμφανίζει συμπληρωμένη την τελική φόρμα σε εκτυπώσιμη μορφή. Ένα παράδειγμα φαίνεται στην παρακάτω εικόνα.

ΦΥΛΛΟ ΕΓΚΡΙΣΗΣ ΔΑΠΑΝΩΝ ΤΑΞΙΔΙΩΝ ΗΜΕΡ/ΝΙΑ:18/03/06

ΣΥΝΕΡΓΑΤΗΣ: Σοφία Παππά				
ΕΡΓΟ: E-Learn 2006			ΠΟΣΟΣΤΟ ΧΡΕΩΣΗΣ ΕΡΓΟΥ:	
ΑΙΤΙΟΛΟΓΙΑ: 3rd IEEE International Conference on Advanced Learning Technologies				
ΗΜΕΡΟΜΗΝΙΑ ΑΝΑΧΩΡΗΣΗΣ: 17 Μαΐου 2006		ΗΜΕΡΟΜΗΝΙΑ ΕΠΙΣΤΡΟΦΗΣ: 25 Μαΐου 2006		
ΚΑΤΗΓΟΡΙΑ ΔΑΠΑΝΩΝ	ΠΛΗΡΩΤΕΑ ΣΤΟΝ ΣΥΝΕΡΓΑΤΗ(1)	ΠΛΗΡΩΤΕΑ ΑΠΟ ΤΟ Ε.Α.Ι.Τ.Υ(2)	ΚΩΔΙΚΟΣ ΕΡΓΟΥ(*)	ΚΩΔΙΚΟΣ ΑΙΤΗΜΑΤΟΣ ΕΓΓΡΑΦΗΣ(*)
ΕΙΣΙΤΗΡΙΑ		400 €		
ΞΕΝΟΔΟΧΕΙΟ	584 €			
ΕΞΟΔΑ ΔΙΑΒΙΩΣΗΣ 2*22 =44€ 3*44=132 € 4*66 =264 €	440€			
Registration Fees				
ΔΙΑΦΟΡΑ ΕΞΟΔΑ αυτοκίνητο:40€ Διόδια:12€	52€			
ΣΥΝΟΛΟ	1076€	400 €		
ΓΕΝΙΚΟ ΣΥΝΟΛΟ(1)+(2)	1476€			
			ΕΓΓΡΑΦΕΣ ΓΙΑ ΠΛΗΡΩΤΕΑ ΣΤΟ ΣΥΝΕΡΓΑΤΗ	
ΠΡΟΚΑΤΑΒΟΛΗ	200 €			
ΥΠΟΛΟΙΠΟ ΠΡΟΣ ΠΛΗΡΩΜΗ	1276€			
ΥΠΟΓΡΑΦΗ ΣΥΝΕΡΓΑΤΗ		ΕΓΚΡΙΣΗ ΥΠΕΡΓΟΥ	ΕΓΚΡΙΣΗ & ΕΛΕΓΧΟΣ ΔΙΑΧΕΙΡΙΣΤΗ	ΕΓΚΡΙΣΗ Δ.Δ.Υ.Ο
ΠΑΡΑΤΗΡΗΣΕΙΣ:				

(*): ΣΥΜΠΛΗΡΩΝΕΤΑΙ ΑΠΟ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ

Εικόνα 13. Εκτυπώσιμη μορφή ταξιδιού

<p>Κεντρική σελίδα</p> <p>Δημιουργία</p> <p>Ιστορικό</p> <p>Δημιουργία χρήστη</p> <p>Αλλαγή χρήστη</p> <p>Δημιουργία configuration</p> <p>Αλλαγή configuration</p>		<p>χρήστης: Eleni Anagnostopoulou Αποσύνδεση</p> <p>Όλα τα ταξίδια</p> <p>1 - 19th ACM Symposium on Applied Computing - 21 Σεπτεμβρίου 2005</p> <p>2 - IADIS International Conference - 16 Μαρτίου 2006</p>
--	--	--

Εικόνα 14. Λίστα ταξιδιών

4.5.4 Δημιουργία Χρηστών

Με την επιλογή της δημιουργίας χρήστη πρέπει ο διαχειριστής να οδηγείται σε μια σελίδα όπου εκεί θα εισάγει τα στοιχεία του νέου χρήστη τα οποία είναι τα ακόλουθα :

Όνομα χρήστη:

Κωδικός:

Όνομα/μο:

Τύπος:

Ακολουθεί η εικόνα της σελίδας

<p>Κεντρική σελίδα</p> <p>Δημιουργία</p> <p>Ιστορικό</p> <p>Δημιουργία χρήστη</p> <p>Αλλαγή χρήστη</p> <p>Δημιουργία configuration</p> <p>Αλλαγή configuration</p>		<p>όνομα χρήστη: <input type="text"/></p> <p>κωδικός: <input type="text"/></p> <p>Όνομα/μο: <input type="text"/></p> <p>τύπος: Προπτυχιακός <input type="button" value="▼"/></p> <p><input type="button" value="Υποβολή"/> <input type="button" value="Αναίρεση"/></p>
--	--	--

Εικόνα 15. Εισαγωγή νέου χρήστη

4.5.5 Αλλαγή χρήστη

Εδώ ο διαχειριστής έχει την δυνατότητα να δει όλους τους χρήστες του συστήματος και αφού επιλέξει κάποιον χρήστη να μπορέσει να τροποποιήσει τα στοιχεία του.

Κεντρική σελίδα Δημιουργία Ιστορικό Δημιουργία χρήστη Αλλαγή χρήστη Δημιουργία configuration Αλλαγή configuration	<p style="text-align: right;">χρήστης: Eleni Anagnostopoulou Αποσύνδεση</p> <p>Όλοι οι χρήστες</p> <p>1 - Eleni Anagnostopoulou</p> <p>2 - Σοφία Παππά</p> <p>3 - Μαρία Καμπά</p> <p>4 - Γιώργος Αλεξίου</p>

Εικόνα 16. Εμφάνιση χρηστών

4.5.6 Δημιουργία configuration

Στη συγκεκριμένη σελίδα δημιουργούμε το προφίλ του χρήστη. Οι κατηγορίες που έχουμε είναι 3: Προπτυχιακος, Μεταπτυχιακός και Καθηγητής. Ανάλογα με την επιλογή που θα κάνουμε οι χρεώσεις θα διαφέρουν από κατηγορία σε κατηγορία.

Κεντρική σελίδα Δημιουργία Ιστορικό Δημιουργία χρήστη Αλλαγή χρήστη Δημιουργία configuration Αλλαγή configuration	<p>Όνομα: <input type="text"/></p> <p>Αντοκίνητο: <input type="text"/></p> <p>Διόδια: <input type="text"/></p> <p>Ημέρες ταξιδιού: <input type="text"/></p> <p>Ημέρες με γεύμα: <input type="text"/></p> <p>Ημέρες χωρίς γεύμα: <input type="text"/></p> <p><input type="button" value="Υποβολή"/> <input type="button" value="Αναίρεση"/></p>

Εικόνα 17. Δημιουργία configuration

4.5.7 Αλλαγή configuration

Με αυτήν την επιλογή ο διαχειριστής αφού επιλέξει την κατηγορία που επιθυμεί να τροποποιήσει μπορεί να αλλάξει το configuration της. Το σύστημα για κάθε πεδίο εμφανίζει την ισχύουσα τιμή.

Κεντρική σελίδα Δημιουργία Ιστορικό Δημιουργία χρήστη Αλλαγή χρήστη Δημιουργία configuration Αλλαγή configuration	<p>Όλα τα configurations</p> <p>1 - Προπτυχιακός</p> <p>2 - Μεταπτυχιακός</p> <p>3 - Καθηγητής</p>	<p>χρήστης: Eleni Anagnostopoulou Αποσύνδεση</p>

Εικόνα 18. Εμφάνιση όλων των configuration

4.6 Διασύνδεση βάσεων δεδομένων και Web

Όταν μιλάμε για διασύνδεση βάσης δεδομένων και Web εννοούμε τη δυνατότητα να μπορούμε μέσα από ένα Web Site, με τη συμπλήρωση για παράδειγμα κάποιας φόρμας, να δημιουργήσουμε ένα query (ερώτημα) προς μια βάση δεδομένων και αφού αυτό εκτελεστεί από το σύστημα διαχείρισης της βάσης δεδομένων, να μας επιστρέψει τα αποτελέσματα, δηλαδή τα περιεχόμενα της βάσης δεδομένων που ικανοποιούν τις παραμέτρους (συνθήκες) που θέσαμε μέσω του query.

Η εκτέλεση ενός query σε μια βάση δεδομένων έχει σαν αποτέλεσμα την εισαγωγή δεδομένων στη βάση, την ενημέρωση ή και την διαγραφή δεδομένων της βάσης.

Τα δεδομένα αυτά θα παρουσιαστούν σε εμάς, σε μορφή HTML μέσα από μια web σελίδα στο site στο οποίο ήδη βρισκόμαστε.

Έτσι έχοντας πρόσβαση σε μια βάση δεδομένων, έχουμε τη δυνατότητα να χρησιμοποιήσουμε τα περιεχόμενα της με διάφορους τρόπους στο Web.

4.6.1 Σελίδες Πολλαπλού Σκοπού (Multi-Purpose Pages)

Ας υποθέσουμε ότι θέλουμε να φτιάξουμε ένα site ώστε να εμφανίζει το όνομα του επισκέπτη (χρήστη) στην κορυφή της κάθε σελίδας. Με το παράδειγμα που είδαμε προηγουμένως με την εμφάνιση του χαιρετιστήριου μηνύματος, έχουμε κάνει τη μισή δουλειά.

Για να μπορέσουμε να επεκτείνουμε το παράδειγμα σ' αυτό που θέλουμε να πετύχουμε, θα πρέπει να ξεπεράσουμε τα εξής προβλήματα :

- Θα πρέπει να εμφανίσουμε το όνομα σε κάθε σελίδα του site και όχι μόνο σε μία.
- Δεν μπορούμε να ελέγξουμε ποια σελίδα του site βλέπει πρώτα ένας χρήστης.

Το πρώτο πρόβλημα δεν είναι πολύ δύσκολο να ξεπεραστεί. Εφόσον έχουμε το όνομα του χρήστη σε μια μεταβλητή σε κάποια σελίδα, μπορούμε να το μεταβιβάσουμε σε μια άλλη σελίδα προσθέτοντας το όνομα στο query string όλων των συνδέσμων (links), ως εξής :

```
<A HREF="newpage.php?name=<?php echo(urlencode($name));?>"> A link  
</A>
```

Έχουμε ενσωματώσει τον κώδικα της PHP στη μέση ενός tag της HTML. Αυτό είναι νόμιμο και δουλεύει κανονικά. Η συνάρτηση echo() είναι γνωστή, αλλά η συνάρτηση *urlencode()* είναι καινούργια και η δουλειά της είναι να λάβει σαν είσοδο κάποιους ειδικούς χαρακτήρες του string, όπως τα κενά για παράδειγμα, και να τους μετατρέψει σε ειδικούς κωδικούς για να μπορέσουν να εμφανισθούν στο query string.

Μέχρι εδώ έχουμε κατορθώσει να μεταβιβάσουμε το όνομα του χρήστη σ' όλους τους συνδέσμους (links) του site. Τώρα αυτό που πρέπει να κάνουμε είναι να ζητάμε πρώτα απ' όλα αυτό το όνομα. Στο παράδειγμα με το μήνυμα

καλωσορίσματος, είχαμε μια ειδική HTML σελίδα με μια φόρμα που προέτρεπε τον χρήστη να καταχωρήσει το όνομά του.

Το πρόβλημα είναι, όμως, ότι δεν μπορούμε να αναγκάσουμε τον χρήστη να μπαίνει στο Web site από εκείνη τη σελίδα κάθε φορά που επισκέπτεται το site μας.

Η λύση είναι να κάνουμε έναν έλεγχο σε κάθε σελίδα του site για να διαπιστώσουμε αν έχει καταχωρηθεί ένα όνομα και να ζητάμε ένα όνομα από τον χρήστη αν χρειασθεί. Αυτό σημαίνει ότι όλες οι σελίδες του site είτε θα εμφανίζουν το περιεχόμενό τους ή μια προτροπή για να καταχωρηθεί ένα όνομα ανάλογα με το αν έχει μια τιμή η μεταβλητή \$name.



5 Θέματα Υλοποίησης

5.1 Αρχεία Εφαρμογής

Παρακάτω θα παρουσιάσουμε τα αρχεία της εφαρμογής μας και τις λειτουργίες του καθενός. Ενδεικτικός κώδικας των βασικών αρχείων βρίσκεται στο Παράρτημα. Τα αρχεία, τα οποία βρίσκονται και στο συνοδευτικό CD της εργασίας, είναι τα παρακάτω:

- 1) Users.php
- 2) Auth.php
- 3) authorize.php
- 4) Config.php
- 5) config new_user.php
- 6) Create.php
- 7) create_md5.php

- 8) Form1.php
- 9) General.php
- 10) History.php
- 11) Index.php
- 12) Insert1.php
- 13) Insert2.php
- 14) left_menu.php
- 15) Logout.php
- 16) modify_configuration.php
- 17) modify_trip.php
- 18) modify_user.php
- 19) New_type.php
- 20) New_user.php
- 21) Ru6.css
- 22) show_apodosi.php

23) show_user_info.php

24) Showform.php

25) stylesheet.css

26) Table.php

27) Teliko.php

28) Type.php

29) update_trip.php

30) update_type.php

31) update_user.php

5.2 Πηγαίος Κώδικας ενδεικτικών λειτουργιών

Παρακάτω μπορούμε να δούμε κάποιους σημαντικούς κώδικες της εφαρμογής μας.

5.2.1 Σύνδεση στη βάση δεδομένων

```
$link = mysql_connect("localhost", "root", "")  
    or die("Could not connect : " . mysql_error());  
//echo "Connected successfully";  
mysql_select_db("egrisi") or die("Could not select database");
```

5.2.3 Αρχικοποίηση session

```
<?php
session_start();
if(!isset($_SESSION['iamin']) || $_SESSION['iamin'] !=
99)
{
header('Location: index.php');
}
?>
```

5.2.4 Αυθεντικοποίηση του χρήστη

```
<?php
session_start();
// edw tha prepei na ginei connection stin databsae...

$link = mysql_connect("localhost", "root", "")
    or die("Could not connect : " . mysql_error());
//echo "Connected successfully";
mysql_select_db("egrisi") or die("Could not select
database");
$username = addslashes($_POST['username']);
$password = $_POST['pass'];
$query = "SELECT id,username,password from users where
username=\"".$username."\"";
$result = mysql_query($query);
$stoixeia = mysql_fetch_array($result);
$cid = $stoixeia[0];
$usr = $stoixeia[1];
$pwd = $stoixeia[2];

$realpass = md5($password);
if($username!= $usr || $realpass!=$pwd)
{
    echo "Couldn't Connect";
    exit();
}
```

```

}
else if ($username == $usr && $realpass == $pwd)
{
session_register("iamin");
session_register("u_id");

$_SESSION['u_id'] = $sid;
$_SESSION['iamin'] = 99;
echo "<a href=\"general.php\">Click to Continue</a>";
}
?>

```

5.2.5 Εισαγωγή στοιχείων από φόρμα στη βάση δεδομένων

```

<form action="config.php" method="post">
<table cellpadding="0" cellspacing="0" class="border"
align="center">
  <tr>
    <th>Όνομα:</th>
    <td><input type="text" name="name"></td>
  </tr>
  <tr>
    <th>Αυτοκίνητο:</th>
    <td><input type="text" name="car"></td>
  </tr>
  <tr>
    <th>Διόδια:</th>
    <td><input type="text" name="tolls"></td>
  </tr>
  <tr>
    <th>Ημέρες ταξιδιού:</th>
    <td><input type="text" name="day_trip"></td>
  </tr>
  <tr>
    <th>Ημέρες με γεύμα:</th>
    <td><input type="text" name="day_launch"></td>
  </tr>
</tr>

```


ΠΑΡΑΡΤΗΜΑ

Αρχείο: authorize.php

```
<?php
session_start();
// edw tha prepei na ginei connection stin databsae...

$link = mysql_connect("localhost", "root", "")
    or die("Could not connect : " . mysql_error());
//echo "Connected successfully";
mysql_select_db("egrisi") or die("Could not select database");
$username = addslashes($_POST['username']);
$password = $_POST['pass'];
$query = "SELECT id,username,password from users where
username=\"".$username.\"\"";
$result = mysql_query($query);
$result_array = mysql_fetch_array($result);
$id = $result_array[0];
$username = $result_array[1];
$password = $result_array[2];

$realpassword = md5($password);
if($username != $username || $realpassword != $password)
{
    echo "Couldn't Connect";
    exit();
}
else if ($username == $username && $realpassword == $password)
{
    session_register("iamin");
    session_register("u_id");

$_SESSION['u_id'] = $id;
$_SESSION['iamin'] = 99;
echo "<a href=\"general.php\">Click to Continue</a>";
}??>
```


Αρχείο: auth.php

```
<?php
session_start();
if(!isset($_SESSION['iamin']) || $_SESSION['iamin'] != 99)
{
header('Location: index.php');
}
?>
```

Αρχείο: ru6.css

```
.full {
    border: 1px solid #000000;
}
.inputs {
    border: 1px solid #000000;
    font-family:Georgia, "Times New Roman", Times, serif;
    font-size:12px;
}
/*.links:link {
font-family:Georgia, "Times New Roman", Times, serif;
font-size:14px;
color:black;
text-decoration:none;
}
.links:active {
font-family:Georgia, "Times New Roman", Times, serif;
font-size:14px;
color:black;
text-decoration:none;
}
.links:visited {
font-family:Georgia, "Times New Roman", Times, serif;
font-size:14px;
color:black;
text-decoration:none;
}

.links:hover {
font-family:Georgia, "Times New Roman", Times, serif;
font-size:14px;
color:black;
text-decoration:underline;
}*/
```

Αρχείο: update.php

```
<?php
    include("auth.php");
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```



```
</tr>
</table>
</body>
</html>
```

Αρχείο: insert1.php

```
<?php
$link = mysql_connect("localhost", "root", "") or die("Could not
connect : " . mysql_error());
//echo "Connected successfully";
mysql_select_db("egrisi") or die("Could not select database");

$sinergatis=stripslashes ($_POST['sinergatis']);
$ergo =stripslashes ($_POST['ergo']);
$cause=stripslashes($_POST['cause']);
$dlive=$_POST['dayliving'];
$darrival=$_POST['arrival'];
$tickets=intval($_POST['tickets']);
$hotel=intval($_POST['hotel']);
$dlaunch=intval($_POST['daylaunch']);
$fees=intval($_POST['fees']);
$car=$_POST['car'];
$prokat=$_POST['prokatab'];
$paratiriseis=nl2br(addslashes($_POST['mes']));

if ($car == "on")
    $tmp_car=1;
else
    $tmp_car = 0;

$user_id=$_SESSION['u_id'];
$query = "INSERT INTO test1 VALUES ('', '$sinergatis','$ergo',
'$cause', '$dlive',
'$darrival','$tickets','$hotel','$dlaunch','$fees','$tmp_car','$proka
t','$paratiriseis','$user_id')";

        if(mysql_query($query))
            echo "Success";

        else
            die(mysql_error());

?>
```

Αρχείο: logout.php

```
<?php
session_start();
ob_start();
$_SESSION['u_id'] = 0;
$_SESSION['iamin'] = 0;
header("Location: index.php");
ob_end_flush();
?>
```