

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης



Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ : Ολοκληρωμένο
δικτυοκεντρικό σύστημα βασισμένο σε XML
τεχνολογίες για την παροχή τουριστικών υπηρεσιών

editix

Των φοιτητριών :

Λούκου Ειρήνη AM:2729

Ντούνη Βασιλική AM:2458

Επιβλέπων καθηγητής:

Παπαδάκης Νίκος

Περιεχόμενα

1.1	Περίληψη	6
1.2	Abstract	7
1.	Εισαγωγή	8
1.3	Περίληψη.....	8
1.4	Κίνητρο για την Διεξαγωγή της Εργασίας	8
1.5	Σκοπός και Στόχοι Εργασίας	8
1.6	Δομή εργασίας.....	9
2	Ιστορία του διαδικτύου και η εξέλιξη του	10
2.1	Τι είναι διαδίκτυο	10
2.2	Ιστορία του διαδικτύου.....	11
2.3	Εξέλιξη του διαδικτύου	12
2.4	Δικτυοκεντρικό Σύστημα	13
2.5	Online Σύστημα- Ηλεκτρονικό εμπόριο.....	14
2.6	Ιστορία-Εξέλιξη του ηλεκτρονικού εμπορίου	14
2.7	Πλεονεκτήματα Ηλεκτρονικού Εμπορίου για τις επιχειρήσεις.....	15
2.7.1	Πλεονεκτήματα ηλεκτρονικού εμπορίου για τους καταναλωτές	16
2.8	Μειονεκτήματα του ηλεκτρονικού εμπορίου	16
2.9	Ηλεκτρονικές εφαρμογές με τη βοήθεια της XML	18
3	Γλώσσες και προγράμματα που χρησιμοποιήθηκαν	19
3.1	XML (Extensible Markup Language)	19
3.1.1	Δομή των Xml δεδομένων- Περιγραφή.....	20
3.1.2	Κανόνες της XML	20
3.1.3	Σχήμα ενός εγγράφου XML	22
3.1.4	XPATH.....	23
3.1.5	XSLT	24
3.1.6	XQUERY	25
3.1.7	Σύγκριση XQuery και XSLT.....	26
3.2	Τι είναι HTML	27
3.3	XML vs HTML	28
3.4	Τι είναι JavaScript	30
3.4.1	JavaScript με HTML	33

3.5	PHP.....	34
3.5.1	Επεκτάσεις αρχείων.....	34
3.5.2	Ιστορία της PHP.....	35
3.5.3	Γράφοντας PHP.....	35
3.5.4	Πλεονεκτήματα PHP.....	36
3.5.5	Μειονεκτήματα PHP.....	36
3.6	Τι είναι CSS.....	37
3.6.1	Editix.....	39
3.7	PhpMyAdmin.....	44
3.7.1	PhpMyAdmin.....	44
4	Σχεδιασμός Βάσης-Περιγραφή.....	49
4.1	Τι είναι το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) ή Database Management System (DBMS);.....	50
4.2	Στόχοι μιας Βάσης Δεδομένων.....	50
4.3	Εφαρμογές συστημάτων βάσεων δεδομένων.....	51
4.4	Περιγραφή συστήματος.....	51
4.5	Δημιουργία XSL αρχεία.....	73
5	Επίλογος.....	89
5.1	Βελτιώσεις.....	89
5.2	Συμπεράσματα.....	89
5.3	Βιβλιογραφία.....	90

Πίνακας εικόνων

Εικόνα 1 Εγκατάσταση	39
Εικόνα 2	40
Εικόνα 3	41
Εικόνα 4	41
Εικόνα 5	42
Εικόνα 6 Επιλέγουμε File New	42
Εικόνα 7 Επιλέγουμε Standard Xml document	43
Εικόνα 8 Εισαγωγή κώδικα	43
Εικόνα 9 Μορφοποίηση κώδικα	44
Εικόνα 10 phrmyadmin install	45
Εικόνα 11 Phrmyadmin install	46
Εικόνα 12 phrmyadmin install	46
Εικόνα 13 phrmyadmin install	47
Εικόνα 14 αποθήκευση στο C.....	47
Εικόνα 15 install	48
Εικόνα 16 install	48
Εικόνα 17 ολοκλήρωση εγκατάστασης.....	49
Εικόνα 18 Δημιουργία κώδικα για τα hotels.....	53
Εικόνα 19 Ο Κώδικας για τα cars_motors	55
Εικόνα 20 Δυνατότητα κράτησης αν το element available = "true"	57
Εικόνα 21 Αλλαγή μεταβλητής.....	58
Εικόνα 22 Το κουμπί κράτηση δεν υπάρχει για τις τρεις επιλογές	59
Εικόνα 23 Ο κώδικας για τα offices	60
Εικόνα 24 Δημιουργία κώδικα για τα μουσεία.....	63
Εικόνα 25 Δημιουργία κώδικα για τα αξιοθέατα.....	65
Εικόνα 26 Δημιουργία κώδικα για τα δρομολόγια	68
Εικόνα 27 Αποτελέσματα XSL.....	79
Εικόνα 28 Εμφάνιση αποτελεσμάτων του Xsl.....	81
Εικόνα 29 Αποτέλεσμα xls για τα μουσεία	82
Εικόνα 30 Αποτέλεσμα xsl για τα δρομολόγια.....	83
Εικόνα 31 Αποτέλεσμα xsl για τα αξιοθέατα	84
Εικόνα 32 Η κράτηση.....	86
Εικόνα 33 Ο κώδικας πριν τη κράτηση.....	87
Εικόνα 34 Γίνεται η κράτηση.....	87
Εικόνα 35 Το πεδίο available έχει αλλάξει από true σε false	88

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τον επιβλέποντα καθηγητή Δρ. Νίκο Παπαδάκη για την εμπιστοσύνη που μας έδειξε για την ανάθεση της συγκεκριμένης πτυχιακής εργασίας, καθώς επίσης και για την πολύτιμη βοήθεια και καθοδήγηση καθ' όλη την διάρκεια.

Επίσης θα θέλαμε να ευχαριστήσουμε θερμά την επιτροπή, που μας κάνει την τιμή να αξιολογήσει την προσπάθειά μας.

Τέλος, θα θέλαμε να αφιερώσουμε την πτυχιακή μας στις οικογένειες μας που μας στήριξαν και μας παρείχαν όλα αυτά τα χρόνια, τόσο οικονομική όσο και ηθική βοήθεια.

1.1 Περίληψη

Η συγκεκριμένη πτυχιακή εργασία έχει σαν στόχο την υλοποίηση ενός ολοκληρωμένου δικτυοκεντρικού συστήματος βασισμένο στην τεχνολογία XML. Πιο συγκεκριμένα έχει σαν στόχο την δημιουργία ενός διαδραστικού τουριστικού οδηγού που προσφέρει στον χρήστη μια σειρά από δυνατότητες, όπως για παράδειγμα περιήγηση στην ιστοσελίδα, online κρατήσεις αλλά και λοιπές πληροφορίες για το μέρος που επιθυμεί να επισκεφτεί. Οι τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας είναι η XML για την υλοποίηση της βάσης και HTML για τη δημιουργία της ιστοσελίδας.

1.2 Abstract

This thesis aims to implement an integrated system based on Net Centric Technology XML. More specifically aims to create an interactive tourist guide that offers the user a number of features, such as browsing the website, online booking and other information about the place you wish to visit.

The technologies used for the development work is the XML for the implementation of the database and HTML for website creation.

1. Εισαγωγή

1.3 Περίληψη

Το κύριο θέμα της πτυχιακής μας αφορά την εξολοκλήρου δημιουργία βάσης και ιστοσελίδας ενός τουριστικού οδηγού της πόλης της Ναυπάκτου βασισμένο στην XML. Μέσα από αυτό ο χρήστης έχει τη δυνατότητα να «περιηγηθεί» στη πόλη βλέποντας φωτογραφίες αλλά και να μάθει επιπλέον πληροφορίες για τα αξιοθέατά της.

Η φόρμα προσφέρει κάποιες επιπλέον εξειδικευμένες λειτουργίες, όπως online κρατήσεις σε ξενοδοχεία, ενοικίαση μεταφορικού μέσου αλλά και ακύρωση μιας κράτησης εφόσον ο χρήστης το επιθυμεί.

Τέτοιου είδους συστήματα αποσκοπούν στην διευκόλυνση του χρήστη ώστε να ενημερώνεται και να οργανώνει εύκολα και γρήγορα τις διακοπές του από την άνεση του σπιτιού του.

1.4 Κίνητρο για την Διεξαγωγή της Εργασίας

Ο βασικότερος λόγος που μας ώθησε να επιλέξουμε την συγκεκριμένη πτυχιακή εργασία είναι η διαχρονικότητα τέτοιου είδους τεχνολογιών. Στις μέρες μας ο ηλεκτρονικός υπολογιστής και το διαδίκτυο αποτελούν ένα αναπόσπαστο κομμάτι της καθημερινότητας μας και πλέον όλο και περισσότερος κόσμος εξοικειώνεται με ανάλογες εφαρμογές.

Η κύρια γλώσσα που επιλέξαμε να χρησιμοποιήσουμε για την διεκπεραίωση του συστήματος είναι η XML διότι αναπτύσσεται δυναμικά στον τομέα της πληροφορικής. Η συγκεκριμένη γλώσσα αναμένεται να χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις.

1.5 Σκοπός και Στόχοι Εργασίας

Σκοπός της πτυχιακής μας εργασίας είναι η εξοικείωση μιας βάσης δεδομένων με την χρήση της XML. Για να στηθεί η βάση δεδομένων μας και κατ' επέκταση η ιστοσελίδα που θα την υποστηρίζει χρησιμοποιήσαμε έναν editor και συγκεκριμένα τον editix. Με την χρήση αυτού του προγράμματος σχεδιάσαμε την βάση μας δημιουργώντας ξεχωριστά xml αρχεία που το καθένα περιείχε τις δικές του πληροφορίες, έτσι ώστε να σχεδιαστεί το κάθε πεδίο στην ιστοσελίδα μας. Για παράδειγμα, στην ιστοσελίδα θα εμφανίζεται ένα πεδίο με το όνομα «Ξενοδοχεία» που μέσα θα περιέχει κάποιες πληροφορίες γι' αυτό, λόγου χάρη όνομα, είδος, διεύθυνση και λοιπά.(επιπλέον γλώσσες)

Στόχος για την εκπόνηση της εργασίας αυτής είναι η ενασχόληση και η εμπάθυνση πάνω στις XML τεχνολογίες. Μέσα από αυτό στοχεύουμε στην απόκτηση εμπειρίας πάνω σε θέματα υλοποίησης online συστημάτων. Για την πραγματοποίηση μιας τέτοιας εφαρμογής χρειάστηκε να γίνουν μια σειρά από ενέργειες, όπως:

- A) Εγκατάσταση του προγράμματος editix.
- B) Σχεδιασμός της βάσης με την χρήση xml.
- Γ) Σύνδεση βάσης με την ιστοσελίδα.
- Δ) Έλεγχος σφαλμάτων.

E) Συγγραφή του θεωρητικού μέρους.

Σκοπός αυτής της δουλειάς είναι να εξυπηρετήσει τις ανάγκες του χρήστη να ενημερωθεί για ένα συγκεκριμένο μέρος μέσα από το διαδίκτυο, με τρόπο εύκολο και γρήγορο. Ένας online τουριστικός οδηγός παρέχει όχι μόνο πολλαπλές πληροφορίες αλλά και μια σειρά από διαδραστικές επιλογές (online κρατήσεις- ακυρώσεις). Τα πλεονεκτήματα ενός Online συστήματος είναι η ελευθερία επιλογών που έχουν οι πελάτες της υπηρεσίας για το τι είναι διαθέσιμο τη δεδομένη χρονική στιγμή. Καθώς η κράτηση γίνεται σε πραγματικό χρόνο, ο χρήστης λαμβάνει άμεση διαβεβαίωση ότι αυτό που θέλει είναι ή δεν είναι διαθέσιμο. Οι πελάτες είναι σε θέση να διεξάγουν τις δικές τους συναλλαγές να λαμβάνουν τις επιβεβαιώσεις αλλά και να κάνουν αλλαγές κατά τη διάρκεια ή ακόμη και μετά τη διαδικασία χωρίς ιδιαίτερη πολυπλοκότητα.

1.6 Δομή εργασίας

Η εργασία μας αυτή χωρίζεται σε πέντε κεφάλαια, συμπεριλαμβάνοντας και την εισαγωγή. Στο δεύτερο κεφάλαιο αναλύουμε την σημασία του διαδικτύου και πως αυτό εξελίσσεται και μπαίνει όλο και περισσότερο στην ζωή μας με την πάροδο των χρόνων. Στο αμέσως επόμενο κεφάλαιο αναλύουμε τις γλώσσες και τα προγράμματα που χρησιμοποιήσαμε για να δημιουργήσουμε ένα ολοκληρωμένο δικτυοκεντρικό σύστημα. Στη συνέχεια περιγράψαμε και αναλύσαμε λεπτομερώς την βάση μας δίνοντας παραδείγματα από κομμάτια του κώδικα που δημιουργήσαμε. Στο τελευταίο κεφάλαιο παραθέτουμε τον επίλογο της εργασίας μας συνοψίζοντας τα παραπάνω και παραθέτοντας τα τελευταία μας συμπεράσματα.

2 Ιστορία του διαδικτύου και η εξέλιξη του

2.1 Τι είναι διαδίκτυο

Το διαδίκτυο (Internet) δεν είναι τίποτε άλλο, από ένα δίκτυο υπολογιστών ανά τον κόσμο, οι οποίοι συνδέονται κι επικοινωνούν μεταξύ τους, με την βοήθεια συγκεκριμένων πρωτοκόλλων, κυρίως με τα TCP και IP, χωρίς όμως αυτά να είναι τα μοναδικά. Οι διασυνδεδεμένοι υπολογιστές, οι οποίοι συνυπάρχουν σε ένα «κοινό δίκτυο επικοινωνίας» κάνουν μια ανταλλαγή μηνυμάτων, χρησιμοποιώντας διάφορα πρωτόκολλα «τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού». Έτσι λοιπόν υπάρχει το διαδίκτυο που με τον τρόπο που περιγράψαμε, εξυπηρετεί κάθε χρήστη που θέλει να το χρησιμοποιήσει.

Όταν μιλάμε για διαδίκτυο συνήθως εννοούμε το δημόσιο διαδίκτυο. Υπάρχουν βέβαια και αρκετά ιδιωτικά που ανταλλάσσουν μηνύματα μόνο με υπολογιστές που βρίσκονται σε ένα συγκεκριμένο δίκτυο. Αυτά λέγονται ενδοδίκτυα.

Τα πρώτα χρόνια, οι συσκευές των υπολογιστών περιορίζονταν σε επιτραπέζιους, σταθερούς υπολογιστές, οι οποίοι συνδέονταν στο διαδίκτυο και ο χρήστης επισκεπτόταν διάφορες ιστοσελίδες. Σήμερα η χρήση του διαδικτύου δεν περιορίζεται μόνο σε αυτούς τους υπολογιστές, αλλά όλο και περισσότερα συστήματα μπορούν να συνδεθούν με αυτό. Τέτοια παραδείγματα αποτελούν τα laptops (φορητοί υπολογιστές), κάμερες, ηλεκτρικές συσκευές και πολλά άλλα. Αυτές οι συσκευές που έχουν την δυνατότητα σύνδεσης με το διαδίκτυο, ονομάζονται «υπολογιστές (hosts) ή τερματικά συστήματα (end systems).»

Τα τερματικά συστήματα, μπορούν να συνδεθούν μεταξύ τους μέσω « ζεύξεων επικοινωνίας (communication lines) και μεταγωγείς πακέτων (packet switches) ». Τα δεδομένα, μπορούν να μεταδίδονται με διαφορετικούς ρυθμούς. Εάν τα τερματικά συστήματα θέλουν να ανταλλάξουν δεδομένα, η διαδικασία έχει ως εξής: Το τερματικό σύστημα που θα αποστείλει τα δεδομένα, χωρίζει τα δεδομένα αυτά σε τμήματα και στη συνέχεια τοποθετεί bytes επικεφαλίδας μπροστά σε κάθε ένα τμήμα. Αυτά τα πακέτα που προέκυψαν, ονομάζονται «πακέτα (packets)» και στέλνονται κατόπιν στο τερματικό σύστημα που πρόκειται να γίνει η λήψη, όπου τα πακέτα ανασυντίθενται και παίρνουν ξανά την αρχική μορφή δεδομένων τους.

Όταν ο μεταγωγέας λαμβάνει ένα πακέτο από τις εξερχόμενες ζεύξεις επικοινωνίας, το προωθεί σε μια από τις εξερχόμενες. Τότε οι μεταγωγείς πακέτων μορφοποιούνται ως δρομολογητές και μεταγωγείς επιπέδου ζεύξης (link layer switches). Και οι δύο είναι υπεύθυνοι για την προώθηση των πακέτων προς το τέλος της διαδρομής τους. Από το ένα τερματικό σύστημα στο άλλο (αποστολής-λήψης), όλη αυτή η διαδρομή που κάνουν τα πακέτα μέσα στο διαδίκτυο καλείται « διαδρομή (route) ή μονοπάτι (path)». Μέσα σε αυτή την διαδρομή εκτελούνται πρωτόκολλα, τα οποία είναι υπεύθυνα για τον έλεγχο, τόσο της αποστολής, όσο και της λήψης όλων των πληροφοριών που προκύπτουν μέσα στο διαδίκτυο, με πιο γνωστά τα IP και TCP. Το IP πρωτόκολλο είναι υπεύθυνο για τον καθορισμό των πακέτων αποστολής-λήψης μεταξύ δρομολογητών – τερματικών συστημάτων. Το TCP όμως είναι πολύ πιο αξιόπιστο. Παρόλα αυτά και τα δύο είναι απαραίτητα και χρειάζεται να ακολουθούν κάποιους κανόνες για την αποφυγή καθυστέρησης ή σύγκρουσης πακέτων ώστε να υπάρχει ομαλή σύνδεση στο διαδίκτυο.

2.2 Ιστορία του διαδικτύου

Οι πρώτες απόπειρες για την δημιουργία ενός διαδικτύου ξεκίνησαν στις ΗΠΑ κατά την διάρκεια του ψυχρού πολέμου. Η Ρωσία είχε ήδη στείλει στο διάστημα τον δορυφόρο Σπούτνικ 1 κάνοντας τους Αμερικανούς να φοβούνται όλο και περισσότερο για την ασφάλεια της χώρας τους. Θέλοντας λοιπόν να προστατευτούν από μια πιθανή πυρηνική επίθεση των Ρώσων δημιούργησαν την υπηρεσία προηγμένων αμυντικών ερευνών ARPA (Advanced Research Project Agency) γνωστή ως DARPA (*Defense Advanced Research Projects Agency*) στις μέρες μας. Αποστολή της συγκεκριμένης υπηρεσίας ήταν να βοηθήσει τις στρατιωτικές δυνάμεις των ΗΠΑ να αναπτυχθούν τεχνολογικά και να δημιουργηθεί ένα δίκτυο επικοινωνίας το οποίο θα μπορούσε να επιβιώσει σε μια ενδεχόμενη πυρηνική επίθεση.

Το αρχικό θεωρητικό υπόβαθρο δόθηκε από τον Τζ. Λικλάιντερ (*J.C.R. Licklider*) που ανέφερε σε συγγράμματά του το "γαλαξιακό δίκτυο". Η θεωρία αυτή υποστήριζε την ύπαρξη ενός δικτύου υπολογιστών που θα ήταν συνδεδεμένοι μεταξύ τους και θα μπορούσαν να ανταλλάσσουν γρήγορα πληροφορίες και προγράμματα. Το επόμενο θέμα που πρόκυπτε ήταν ότι το δίκτυο αυτό θα έπρεπε να ήταν αποκεντρωμένο έτσι ώστε ακόμα κι αν κάποιος κόμβος του δεχόταν επίθεση να υπήρχε δίοδος επικοινωνίας για τους υπόλοιπους υπολογιστές. Τη λύση σε αυτό έδωσε ο Πολ Μπάραν (*Paul Baran*) με τον σχεδιασμό ενός κατακεντρωμένου δικτύου επικοινωνίας που χρησιμοποιούσε την ψηφιακή τεχνολογία. Πολύ σημαντικό ρόλο έπαιξε και η θεωρία ανταλλαγής πακέτων του Λέοναρντ Κλάινροκ (*Leonard Kleinrock*), που υποστήριζε ότι πακέτα πληροφοριών που θα περιείχαν την προέλευση και τον προορισμό τους μπορούσαν να σταλούν από έναν υπολογιστή σε έναν άλλο.

Στηριζόμενο λοιπόν σε αυτές τις τρεις θεωρίες δημιουργήθηκε το πρώτο είδος διαδικτύου γνωστό ως ARPANET. Εγκαταστάθηκε και λειτούργησε για πρώτη φορά το 1969 με 4 κόμβους μέσω των οποίων συνδέονται 4 μίνι υπολογιστές (*mini computers 12k*): του πανεπιστημίου της Καλιφόρνια στην Σάντα Μάρμπαρα του πανεπιστημίου της Καλιφόρνια στο Λος Άντζελες, το SRI στο Στάνφορντ και το πανεπιστήμιο της Γιούτα. Η ταχύτητα του δικτύου έφθανε τα 50 kbps και έτσι επιτεύχθηκε η πρώτη *dial up* σύνδεση μέσω γραμμών τηλεφώνου. Μέχρι το 1972 οι συνδεδεμένοι στο ARPANET υπολογιστές έχουν φτάσει τους 23, οπότε και εφαρμόζεται για πρώτη φορά το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου (*e-mail*).

Παράλληλα δημιουργήθηκαν και άλλα δίκτυα, τα οποία χρησιμοποιούσαν διαφορετικά πρωτόκολλα (όπως το x.25 και το UUCP) τα οποία συνδέονταν με το ARPANET. Το πρωτόκολλο που χρησιμοποιούσε το ARPANET ήταν το NCP (*Network Control Protocol*), το οποίο, όμως, είχε το μειονέκτημα ότι λειτουργούσε μόνο με συγκεκριμένους τύπους υπολογιστών. Έτσι, δημιουργήθηκε η ανάγκη στις αρχές του 1970 για ένα πρωτόκολλο που θα ένωνε όλα τα δίκτυα που είχαν δημιουργηθεί μέχρι τότε. Το 1974 λοιπόν, δημοσιεύεται η μελέτη των Βιντ Σερφ (*Vint Cerf*) και Μπομπ Κάαν (*Bob Kahn*) από την οποία προέκυψε το πρωτόκολλο TCP (*Transmission Control Protocol*) που αργότερα το 1978 έγινε TCP/IP, προσετέθη δηλαδή το Internet Protocol (IP), ώσπου το 1983 έγινε το μοναδικό πρωτόκολλο που ακολουθούσε το ARPANET.

Το 1984 υλοποιείται το πρώτο *DNS* (Domain Name System) σύστημα στο οποίο καταγράφονται 1000 κεντρικοί κόμβοι και οι υπολογιστές του διαδικτύου πλέον αναγνωρίζονται από διευθύνσεις κωδικοποιημένων αριθμών. Ένα ακόμα σημαντικό βήμα στην ανάπτυξη του Διαδικτύου έκανε το Εθνικό Ίδρυμα Επιστημών (National Science Foundation, NSF) των ΗΠΑ, το οποίο δημιούργησε την πρώτη διαδικτυακή πανεπιστημιακή ραχοκοκκαλιά (backbone), το NSFNet, το 1986. Ακολούθησε η ενσωμάτωση άλλων σημαντικών δικτύων, όπως το Usenet, το Fidonet και το Bitnet.

Ο όρος Διαδίκτυο/Ίντερνετ ξεκίνησε να χρησιμοποιείται ευρέως την εποχή που συνδέθηκε το ARPANET με το NSFNet και Internet σήμαινε οποιοδήποτε δίκτυο χρησιμοποιούσε TCP/IP. Η μεγάλη άνθιση του Διαδικτύου όμως, ξεκίνησε με την εφαρμογή της υπηρεσίας του Παγκόσμιου Ιστού από τον Τιμ Μπέρνερς-Λι στο ερευνητικό ίδρυμα CERN το 1989, ο οποίος είναι στην ουσία, η "πλατφόρμα", η οποία κάνει εύκολη την πρόσβαση στο Ίντερνετ, ακόμα και στη μορφή που είναι γνωστό σήμερα.

2.3 Εξέλιξη του διαδικτύου

Το πρώτο δίκτυο δημιουργήθηκε το 1968 στο Εθνικό Εργαστήριο Φυσικής στο Ηνωμένο Βασίλειο. Λίγο αργότερα, η « DAPRA (Αμερικανική υπηρεσία) που αργότερα μετονομάστηκε σε ARPA» δημιούργησε ένα πιο μεγάλο δίκτυο που θα είχε στους κόμβους τους πολύ μεγάλους υπολογιστές(υπερυπολογιστές). Ένα χρόνο αργότερα , τοποθετήθηκε στο UCLA ένας κόσμος και προς το τέλος του χρόνου τοποθετήθηκαν άλλοι τρεις. Έτσι δημιουργήθηκε το ARPAnet. Οι υπολογιστές απέκτησαν την δυνατότητα να ανταλλάσσουν μηνύματα και δεδομένα. Λίγο αργότερα οι κόμβοι έγιναν πλέον δεκαπέντε και το 1972 έφτασαν τους τριάντα επτά. Πολύ γρήγορα, δημιουργήθηκαν και τα e-mail, με αποτέλεσμα το ARPAnet να αποτελεί πλέον ένα πολύ γρήγορο και υψηλής ποιότητας ηλεκτρονικό ταχυδρομείο. Πλέον το δίκτυο χρησιμοποιήθηκε για συνεχή ανταλλαγή μηνυμάτων. Οι χρήστες τότε χρησιμοποιούσαν το διαδίκτυο για πολύ συγκεκριμένα πράγματα. Πριν έρθει το 1980, το ARPAnet είχε ακόμη περισσότερους κόμβους και πιο πολλούς χρήστες, οι οποίοι μάλιστα δεν προέρχονταν από πανεπιστημιακά ιδρύματα. Έφτανε μόνο να είχαν ένα υπολογιστή και μια άδεια πρόσβασης από κάποιον υπολογιστή πανεπιστημίου. Εκείνη την εποχή δημιουργήθηκαν τα πρωτόκολλα IP και TCP, τα οποία « αντικατέστησε το NCP (που ήταν το αρχικό πρωτόκολλο του ARPAnet.» Τα TCP και IP δεν άργησαν να γίνουν πρωτόκολλα που χρησιμοποιούσε το ARPAnet. Εκείνο τον χρόνο, χρησιμοποιήθηκε για πρώτη φορά ο όρος «internet».

Στα τέλη της δεκαετίας του 1970 ξεκίνησε το δίκτυο Usenet, το οποίο βασίστηκε στο UUCP πρωτόκολλο. Στο συγκεκριμένο δίκτυο, ξεκίνησαν και οι πρώτες «ομάδες συζήτησης μεταξύ χρηστών του λειτουργικού συστήματος Unix». Οι πληροφορίες που ανταλλάσσουν ήταν κυρίως για την αναζήτηση βοήθειας σχετικά με το Unix. Το Usenet χρειαζόταν ακριβά modems, γι' αυτό και αναπτύχθηκε σχετικά αργά.

Γύρω στο 1990, δημιουργήθηκε η γλώσσα HTML στο «ερευνητικό κέντρο Φυσικής Cern, έξω από την Γενεύη» από τον Tim Berners Lee. Χάρη τον Lee, το διαδίκτυο απέκτησε την μορφή που έχει ακόμη και τώρα. Το Web έχει πολύ εύκολη λειτουργία και χρησιμοποιεί μαζί διάφορα πρωτόκολλα αλλά και υπηρεσίες. Αναμφισβήτητα το πιο «δυνατό στοιχείο του διαδικτύου» αποτελεί το Web διότι έχει την δυνατότητα να μεταφέρει πολυμεσικές πληροφορίες, όπως εικόνα, φωτογραφία ή κείμενο.

Μέχρι και το 1991 το internet το χρησιμοποιούσαν κυρίως στο πλαίσιο της εκπαίδευσης και της έρευνας. Τότε άρχισε να χρησιμοποιείται για εμπορικούς σκοπούς, συναλλαγές,

ανταλλαγές απόψεων και τότε το ARPAnet σταμάτησε ουσιαστικά να υπάρχει. Λίγα χρόνια αργότερα οι κόμβοι πλέον έφτασαν τους χιλιάδες.

Το internet αναπτύσσεται πλέον ταχύτατα και επηρεάζει τα πάντα, κυρίως όμως την ανθρώπινη επικοινωνία. Από το 1988 και μετά «ο αριθμός των κόμβων διπλασιάζεται κάθε χρόνο. Σύμφωνα με τα στοιχεία του European Information Technology Observatory ο μέσος ρυθμός αύξησης που παρουσιάζει το διαδίκτυο στην Ευρώπη μέχρι το 2000 ανέρχεται σε 37% ετησίως. Το σύνολο των Ευρωπαίων χρηστών από 12,5 εκατομμύρια το 1995 θα φτάσει τα 60 εκατομμύρια το έτος 2000. Το 1997 οι χρήστες του διαδικτύου παγκοσμίως υπολογίζονται σε 95,7 εκατομμύρια». Το διαδίκτυο πλέον βρίσκεται παντού. Τόσο σε σχολεία, ιδρύματα, οργανισμούς, αλλά και στα ίδια μας τα σπίτια. Αποτελεί πλέον ένα αξιόπιστο και οικονομικό μέσο επικοινωνίας.

2.4 Δικτυοκεντρικό Σύστημα

Σήμερα βλέπουμε να υπάρχει μια τάση για προσθήκη όλο και περισσότερης πληροφορίας σε ψηφιακή μορφή στο διαδίκτυο. Αυτό συμβαίνει εξαιτίας της ραγδαίας αύξησης του ποσοστού πληροφορίας που υπάρχει σε ψηφιακή μορφή στο διαδίκτυο. Υπάρχουν κάποιες τεχνολογίες όπως είναι η XML, ή τα Java Servlets οι οποίες βοηθούν όχι μόνο στο να οργανωθούν οι πληροφορίες αλλά βρίσκουν λύσεις στον τρόπο που οργανώνονται και διαχειρίζονται αυτές οι πληροφορίες. Οι διαθέσιμες τεχνολογίες που συναντούμε σήμερα, όπως οι DSL τεχνολογίες, δίνουν την δυνατότητα πιο γρήγορης πρόσβασης στο διαδίκτυο, αφήνοντας ευχαριστημένους εκατομμύρια χρήστες.

Αρχές του 20^{ου} αιώνα μπήκαν στην καθημερινότητα όλων μας οι πιο καινοτόμες τεχνολογίες, όπως το ασύρματο internet Wi-Fi και τα Bluetooth. Όπως ήταν φυσικό το κοινό ανταποκρίθηκε πλήρως στις νέες αυτές τεχνολογίες.

Οι χρήστες που μπορούν να αποκτούν πρόσβαση σε συγκεκριμένες πληροφορίες είναι πλέον πάρα πολλοί και σε αυτό έχουν συντελέσει πολύ τα κυψελωτά δίκτυα.

Άλλο ένα κομμάτι που έχει γνωρίσει ιδιαίτερη εξέλιξη, είναι αυτό των ασύρματων δικτύων. Υπάρχουν όμως και πολύ κίνδυνοι που κρύβονται πίσω από αυτά τα δίκτυα διότι όλες οι πληροφορίες είναι εκτεθειμένες σε κινδύνους. Γι' αυτόν ακριβώς τον λόγο, πραγματοποιούνται συνεχώς έρευνες για την εύρεση λύσεων που σχετίζονται με θέματα ασφάλειας.

2.5 Online Σύστημα- Ηλεκτρονικό εμπόριο

Ηλεκτρονικό Εμπόριο ή αλλιώς e-commerce, ή e-comm «ονομάζεται το εμπόριο παροχής αγαθών και υπηρεσιών που πραγματοποιείται εξ αποστάσεως με ηλεκτρονικά μέσα, βασισμένο δηλαδή στην ηλεκτρονική μετάδοση δεδομένων, χωρίς να καθίσταται αναγκαία η φυσική παρουσία των συμβαλλομένων μελών, πωλητή-αγοραστή.»

Πρόκειται για ένα πλήρες σύστημα διαδικασιών, όπως ορίζεται και στο πραγματικό κόσμο. Συναντούμε δηλαδή αγορά, πώληση, εξυπηρέτηση καταναλωτών, πληρωμές και ότι άλλο θα συναντούσαμε σε ένα «κανονικό» μαγαζί. Μάλιστα όλο και περισσότερος κόσμος προτιμά να το διαδίκτυο για τις συναλλαγές του.

2.6 Ιστορία-Εξέλιξη του ηλεκτρονικού εμπορίου

Στην αρχή το ηλεκτρονικό εμπόριο χρησιμοποιούταν για εμπορικές συναλλαγές και κυρίως για ηλεκτρονική μεταφορά χρημάτων(EFT). Αυτές οι συναλλαγές ξεκίνησαν γύρω στο 1970, προκειμένου οι επιχειρηματίες να κάνουν πιο εύκολα τις συναλλαγές τους(παραγγελίες, τιμολόγια και πολλά άλλα). Δέκα περίπου χρόνια μετά, εμφανίστηκαν οι πρώτες πιστωτικές κάρτες που αποτέλεσαν σαφώς ένα είδος ηλεκτρονικού εμπορίου. Λίγο αργότερα, η αεροπορικές εταιρίες Sabre και Travicom από την Αμερική και την Μεγάλη Βρετανία αντίστοιχα, ήταν οι πρώτες που χρησιμοποίησαν σύστημα ηλεκτρονικών κρατήσεων. Από το 1990 και μετά ενσωματώθηκε στο ηλεκτρονικό εμπόριο ένα σύστημα για την διαχείριση δεδομένων(αναζήτηση-αποθήκευση-διαγραφή στοιχείων).

Πλέον το ηλεκτρονικό εμπόριο υπάρχει για τα καλά στην καθημερινότητά μας. Χρησιμοποιείται για την τακτοποίηση λογαριασμών, μέχρι και για την παραγγελία φαγητού. Πολύ μεγάλες εταιρίες το χρησιμοποιούν συνεχώς τόσο σε πρακτικό, όσο και σε ερευνητικό επίπεδο, υπάρχουν όμως ζητήματα ασφάλειας και ακεραιότητας των δεδομένων που αποτελούν ένα πολύ σημαντικό ζήτημα.

Όσο περνούν τα χρόνια το ηλεκτρονικό εμπόριο εξελίσσεται και διευκολύνει τους καταναλωτές. Δίνει την δυνατότητα μεγάλων προσφορών, σύγκριση τιμών και πρόσβαση σε ένα ευρύ φάσμα της αγοράς από τον καναπέ του σπιτιού. Μπορεί κάποιος να βρει πολύ καλές προσφορές και ευκαιρίες σε οποιοδήποτε είδος, από ένα μικρό πορτοφόλι μέχρι ένα πολυτελές ακίνητο.

2.7 Πλεονεκτήματα Ηλεκτρονικού Εμπορίου για τις επιχειρήσεις

Τα τελευταία χρόνια τόσο σε ευρωπαϊκό επίπεδο όσο και στην Ελλάδα το Διαδίκτυο εξελίσσεται συνεχώς, ώστε να καλύψει ένα μεγάλο εύρος αναγκών τόσο των καταναλωτών, όσο και των επιχειρήσεων. Όλο και περισσότεροι άνθρωποι στρέφονται προς στις ηλεκτρονικές αγορές, καθώς αυτό εξυπηρετεί τις ανάγκες τους με τρόπο εύκολο και γρήγορο.

Σε μια εποχή που κινείται σε ψηφιακούς ρυθμούς, βλέπουμε ότι τόσο οι επιχειρήσεις όσο και οι καταναλωτές, έχουν αντιληφθεί την ανάγκη για τη διανομή προϊόντων και υπηρεσιών μέσω διαδικτύου. Όλη αυτή η αλλαγή επιφέρει σημαντικά πλεονεκτήματα σε σχέση με το «κλασικό» τρόπο πώλησης-αγοράς-ενημέρωσης που ξέραμε μέχρι τώρα.

Πρώτα από όλα, προσφέρει διεύρυνση της πελατειακής βάσης. Σε ένα δικτυακό κατάστημα εν δυνάμει μπορούν να έχουν πρόσβαση πελάτες από όλα τα σημεία του πλανήτη. Επιπλέον, οι πελάτες έχουν τη δυνατότητα να χρησιμοποιούν το ηλεκτρονικό κατάστημα οποιαδήποτε μέρα και ώρα.

Ένα πλεονέκτημα ακόμα, που αφορά περισσότερο τους επιχειρηματίες είναι η μείωση του λειτουργικού κόστους. Ένα ηλεκτρονικό κατάστημα για να λειτουργήσει χρειάζεται πολύ μικρότερο κόστος σε σχέση με ένα συμβατικό. Για παράδειγμα, μια συμβατική επιχείρηση μπορεί να πληρώνει ενοίκιο, λογαριασμούς κοινής ωφέλειας, μισθοδοσία προσωπικού, κόστος συστημάτων ασφαλείας που αποτελούν διαχρονικά σημαντικές επιβαρύνσεις για την λειτουργία της. Τα ηλεκτρονικά καταστήματα από την άλλη μπορούν να έχουν πολύ χαμηλότερα κόστη λειτουργίας και συντήρησης. Επιπλέον, προσφέρουν στον επιχειρηματία καλύτερη διαχείριση αποθεμάτων και διευκόλυνση προγραμματισμού. Η διατήρηση αποθεμάτων αποτελεί ένα σημαντικό θέμα για όλες τις επιχειρήσεις, καθώς απαιτεί την δέσμευση ενός κεφαλαίου. Με την χρήση του e-commerce, το απόθεμα αυτό ελαχιστοποιείται και κατ' επέκταση μειώνεται το κεφάλαιο που πρέπει να δεσμεύσει ο επιχειρηματίας.

Άλλο ένα σημαντικό προνόμιο του ηλεκτρονικού εμπορίου είναι το πιο αποτελεσματικό marketing σε συνδυασμό με το μειωμένο κόστος του. Το marketing με τη χρήση των δικτυακών εργαλείων, μπορεί εκτός από φθηνότερο να είναι και αποτελεσματικότερο. Οι συμβατικοί τρόποι προώθησης ενός προϊόντος είναι συνήθως πιο ακριβοί σε σχέση με τους ηλεκτρονικούς. Ας μην ξεχνάμε ότι, ένα ηλεκτρονικό κατάστημα λειτουργεί και αυτοτελώς ως μέσο διαφήμισης, ειδικά όταν συνδυαστεί με άλλα εργαλεία ηλεκτρονικής διαφήμισης (social networks, Google AdWords).

Το Διαδίκτυο είναι αμφίδρομο μέσο και έτσι μπορούν να αντλούνται χρήσιμες πληροφορίες για το προφίλ των καταναλωτών, κάτι που μπορεί να κατευθύνει καλύτερα τους υπεύθυνους marketing αλλά και τους επιχειρηματίες για την προσέλκυση υποψήφιων πελατών.

2.7.1 Πλεονεκτήματα ηλεκτρονικού εμπορίου για τους καταναλωτές

Σε μια αμφίδρομη σχέση μεταξύ καταναλωτή-επιχείρησης, το ηλεκτρονικό εμπόριο δεν επιδρά εξίσου θετικά μόνο στις επιχειρήσεις, αλλά υπάρχουν μια σειρά από πλεονεκτήματα που φέρουν πολλά οφέλη και στον καταναλωτή.

Πρώτα απ' όλα ο πελάτης-καταναλωτής έχει την δυνατότητα να αγοράσει ή να παραγγείλει κάποιο προϊόν οποιαδήποτε ώρα και στιγμή της ημέρας. Ο λόγος είναι ότι τα ηλεκτρονικά καταστήματα παραμένουν ανοιχτά 24 ώρες το 24ωρο. Με άλλα λόγια, οποιαδήποτε στιγμή επιθυμεί ο πελάτης, μπορεί να κλείσει ένα αεροπορικό εισιτήριο ή να αγοράσει ένα ρούχο.

Σημαντικό, επίσης, αποτελεί το γεγονός ότι τα προϊόντα πωλούνται σε χαμηλότερο κόστος απ' ότι αυτά των «κανονικών» καταστημάτων. Αυτό συμβαίνει διότι ένα ηλεκτρονικό κατάστημα είναι απαλλαγμένο από μεγάλο μέρος του λειτουργικού κόστους ενός πραγματικού καταστήματος (ενοικίαση χώρου, ηλεκτρικό ρεύμα κτλ.) και γενικά απαιτεί λιγότερο υπαλληλικό προσωπικό.

Με το ηλεκτρονικό εμπόριο, τα σύνορα καταργούνται, καθώς η αγορά γίνεται παγκόσμια. Με άλλα λόγια, μπορεί κάποιος μέσω του υπολογιστή, κινητού ή tablet να αγοράσει κάτι το οποίο δεν κυκλοφορεί στην Ελλάδα, χωρίς να πρέπει να ταξιδέψει στο εξωτερικό για να το αποκτήσει. Αυτός ο τρόπος καθιστά την συναλλαγή γρήγορη και άμεση, διότι από την στιγμή που ολοκληρώνεται η παραγγελία, το προϊόν φτάνει στον πελάτη μέσα σ' ένα πολύ σύντομο χρονικό διάστημα.

Εν κατακλείδι το πιο πρακτικό και σημαντικό όφελος για τον καταναλωτή από το ηλεκτρονικό εμπόριο, είναι ότι ο καθένας βρίσκει αυτό που θέλει, οποιαδήποτε στιγμή το θέλει, χωρίς κόπο και χωρίς καμιά σπατάλη χρόνου. Με τρόπο εύκολο και γρήγορο μπορεί να κάνει τα ψώνια του είτε από το σπίτι του, είτε από τον χώρο εργασίας του.

2.8 Μειονεκτήματα του ηλεκτρονικού εμπορίου

Όσον αφορά τα μειονεκτήματα του ηλεκτρονικού εμπορίου που έχουν σχέση με τους καταναλωτές παρατηρούμε ότι:

Οι καταναλωτές, ειδικά στην Ελλάδα, δεν εμπιστεύονται ακόμα 100% το Internet για τις αγορές τους και κυρίως διστάζουν να αποστείλουν τον αριθμό της πιστωτικής τους κάρτας μέσω αυτού. Αν και τα περισσότερα ζητήματα ασφάλειας έχουν λυθεί μέσω της κρυπτογράφησης και της πιστοποίησης, δεν έχει ακόμη δημιουργηθεί εκείνο το κλίμα που θα πείθει τους καταναλωτές ότι οι συναλλαγές τους είναι απολύτως ασφαλείς.

Άλλο ένα ζήτημα που προκύπτει στις συναλλαγές μέσω διαδικτύου, είναι ότι δεν μπορεί να υπάρξει επαρκής έλεγχος στην ποιότητα των προϊόντων που πωλούνται, ειδικά όταν πρόκειται για είδη ρουχισμού, τροφίμων κτλ. Επιπλέον, τα ηλεκτρονικά καταστήματα που πωλούν μεγάλα ή βαριά αντικείμενα είναι ελάχιστα, αλλά και αυτά που υπάρχουν τις περισσότερες φορές έχουν μεγάλο κόστος αποστολής. Γενικά οτιδήποτε απαιτεί τη δική μας φυσική παρουσία για την αγορά του είναι αρκετά δύσκολο, προς το παρόν τουλάχιστον, να πουληθεί μέσω του Internet.

Εξαιτίας του γεγονότος ότι ιδιαίτερα στην Ελλάδα, το ηλεκτρονικό εμπόριο άρχισε να έχει εμφανή παρουσία τα τελευταία χρόνια, δεν υπάρχουν ακόμα πολλά ηλεκτρονικά καταστήματα που θα δημιουργούσαν τον απαραίτητο ανταγωνισμό, από τον οποίο

προκύπτουν και πολλά από τα οφέλη του εν λόγω είδους εμπορίου.

Από τη μεριά της επιχείρησης τα περισσότερα όμως από τα μειονεκτήματα του ηλεκτρονικού εμπορίου σήμερα, που προκύπτουν είναι οι οικονομικές επιβαρύνσεις. Πιο συγκεκριμένα, το κόστος για τη δημιουργία ενός απλού ηλεκτρονικού καταστήματος μπορεί να μεταβληθεί δραματικά, καθώς προστίθενται καθημερινά νέες τεχνολογίες που βοηθούν στην υλοποίηση του. Κάθε επιχείρηση θα πρέπει να επενδύει ένα αρκετά μεγάλο χρηματικό ποσό κάθε χρόνο για τη συντήρηση και βελτιστοποίηση του ηλεκτρονικού της καταστήματος.

Τέλος, όσο περισσότεροι πελάτες επισκέπτονται ένα ηλεκτρονικό κατάστημα, τόσο αυξάνονται και οι ανάγκες που θα πρέπει να υποστηρίζονται από την βάση δεδομένων της ηλεκτρονικής φόρμας. Αυτό πρακτικά σημαίνει πως στην βάση πρέπει να ενσωματώνονται συνεχώς καινούριες λειτουργίες, άρα περισσότερα χρήματα θα ξοδεύονται για την συντήρησή της.

2.9 Ηλεκτρονικές εφαρμογές με τη βοήθεια της XML

Το ηλεκτρονικό εμπόριο (e - commerce) όπως, και άλλες εφαρμογές του Διαδικτύου παρέχουν Web interfaces (διεπαφές) ώστε να δίνουν τη δυνατότητα πρόσβασης σε πληροφορίες που είναι αποθηκευμένες σε μία ή περισσότερες βάσεις δεδομένων. Αυτές οι βάσεις δεδομένων συχνά αναφέρονται ως πηγές δεδομένων. Τέτοιου είδους εφαρμογές στηρίζονται στο μοντέλο client/ server και χρησιμοποιούν την αρχιτεκτονική τριών επιπέδων. Πιο συγκεκριμένα, το πρώτο επίπεδο αποτελεί το περιβάλλον του τελικού χρήστη (user interface), το δεύτερο επίπεδο περιγράφει τον εξυπηρετή εφαρμογής (Application Server) και τέλος το τρίτο επίπεδο αφορά τον εξυπηρετή βάσης δεδομένων (Database Server). Το ηλεκτρονικό εμπόριο(E – commerce) καθώς και άλλες παρόμοιες εφαρμογές έχουν σχεδιαστεί για να αλληλεπιδρούν με τον χρήστη μέσω Web interfaces.

Για να υλοποιηθούν αυτές οι εφαρμογές απαραίτητο είναι το περιεχόμενό τους να μορφοποιηθεί κατάλληλα με τη χρήση εγγράφων υπερκειμένου (hypertext documents). Υπάρχουν διάφορες γλώσσες για την σύνταξη αυτών των εγγράφων, η πιο ευρέως γνωστή είναι η HTML. Αν και η HTML χρησιμοποιείται κατά κόρον, για τη μορφοποίηση και δόμηση αυτών των εγγράφων έχει ένα βασικό μειονέκτημα. Η γλώσσα αυτή δεν έχει τη δυνατότητα να εξάγει δεδομένα μιας βάσης από δομημένα έγγραφα. Αυτό το πρόβλημα έρχεται η XML να λύσει.

Η XML έχει οριστεί ως μια γλώσσα πρότυπο για τη δόμηση και την ανταλλαγή δεδομένων μέσω Web. Εκτός από αυτό, δίνει τη δυνατότητα στον προγραμματιστή να εξάγει τα δεδομένα από μια βάση πιο εύκολα. Επιπλέον, καθορίζει τον τρόπο με τον οποίο θα εμφανιστεί η σελίδα στον τελικό χρήστη. Για να επιτευχθεί η μορφοποίηση ενός εγγράφου η XML περιλαμβάνει τις αντίστοιχες γλώσσες όπως, την XSL και XSLT. Αξίζει να σημειωθεί ότι η XML έχει προταθεί ως πιθανό μοντέλο αποθήκευσης και ανάκτησης δεδομένων παρόλο αυτά έχουν γίνει λίγες δοκιμές υλοποίησης βάσεων δεδομένων.

Με τη χρήση της HTML μπορεί κάποιος να δημιουργήσει στατικές ιστοσελίδες με σταθερό κείμενο και άλλα αντικείμενα. Οι εφαρμογές, όπως, το ηλεκτρονικό εμπόριο απαιτούν ιστοσελίδες που παρέχουν διαδραστικά χαρακτηριστικά με το χρήστη. Για παράδειγμα, αν κάποιος θέλει να κλείσει ένα δωμάτιο ξενοδοχείου, θα πρέπει να εισάγει πληροφορίες, όπως, την ημερομηνία και το είδος του δωματίου που επιθυμεί, στα κατάλληλα πεδία της φόρμας της ιστοσελίδας. Το πρόγραμμα Web πρέπει πρώτα να υποβάλει ερώτημα στη βάση δεδομένων του ξενοδοχείου να ανακτήσει τις εν λόγω πληροφορίες και στη συνέχεια, να εμφανίσει τα αντίστοιχα αποτελέσματα . Αυτές οι ιστοσελίδες , όπου μέρος της πληροφορίας που προέρχεται από βάσεις δεδομένων ή άλλες πηγές δεδομένων ονομάζονται δυναμικές Ιστοσελίδες , επειδή τα δεδομένα που εξάγονται και εμφανίζονται κάθε φορά, θα είναι διαφορετικά για τις κρατήσεις και τις ημερομηνίες .

3 Γλώσσες και προγράμματα που χρησιμοποιήθηκαν

3.1 XML (Extensible Markup Language)

Τι είναι Xml

Η XML (**Extensible Markup Language**) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple.

Η XML όπως η Hyper-text Markup Language (HTML) στην οποία βασίζεται το World Wide Web έχει τις ρίζες της στη διαχείριση εγγράφων και παράγεται από μια γλώσσα για δόμηση μεγάλων εγγράφων, που είναι γνωστά ως Standard Generalized Markup Language (SGML). Ωστόσο, αντίθετα από την SGML και την HTML, η XML μπορεί να αναπαραστήσει δεδομένα βάσεων δεδομένων, όπως επίσης πολλά άλλα είδη δομημένων δεδομένων που χρησιμοποιούνται σε επαγγελματικές εφαρμογές. Είναι πολύ χρήσιμη για μορφοποίηση των δεδομένων όταν μια εφαρμογή πρέπει να επικοινωνεί με μια άλλη εφαρμογή, ή να ενοποιεί πληροφορίες από διάφορες εφαρμογές. Όταν η XML χρησιμοποιείται με αυτόν τον τρόπο, προκύπτουν πολλά θέματα σε σχέση με τις βάσεις δεδομένων, όπως πώς να οργανώνει ο χρήστης, να χειρίζεται και να κάνει ερωτήματα στα XML δεδομένα.

3.1.1 Δομή των Xml δεδομένων- Περιγραφή

Για να δημιουργηθεί ένα αρχείο Xml θα πρέπει ο κώδικας να παίρνει τη μορφή ετικετών (tag) που περικλείονται από αγκύλες (<>). Μέσα σε αυτές τις αγκύλες περιέχονται το όνομα του στοιχείου και πιθανώς κάποιες επιπλέον πληροφορίες. Το τέλος του τμήματος που αναφέρεται η ετικέτα περιλαμβάνει μια κάθετο (/) ακολουθούμενη από το όνομα του στοιχείου. Παραδείγματος χάριν,

```
<title> OurProject </title>.
```

Η βασική δομή σε ένα Xml έγγραφο είναι το στοιχείο (element). Ένα στοιχείο είναι απλώς ένα ζευγάρι από ετικέτες αρχής και τέλους και όλο το κείμενο μεταξύ τους.

Τα XML έγγραφα πρέπει να έχουν ένα στοιχείο ρίζας, που περιέχει όλα τα άλλα στοιχεία του εγγράφου.

Οι ετικέτες είναι σωστά ένθετες αν κάθε ετικέτα αρχής έχει μια μοναδική, αντίστοιχη ετικέτα τέλους που βρίσκεται στο περιβάλλον του ίδιου γονικού στοιχείου.

```
<account>..... <balance>.....</balance>.....</account>
```

Στην Xml το κείμενο μπορεί να αναμιχθεί με τα υποστοιχεία ενός στοιχείου. Όπως και πολλές άλλες λειτουργίες της Xml, αυτή η ελευθερία έχει περισσότερη έννοια σε περιβάλλον επεξεργασίας εγγράφων παρά σε περιβάλλον επεξεργασίας δεδομένων και δεν είναι ιδιαίτερα χρήσιμη για αναπαράσταση πιο δομημένων δεδομένων, όπως περιεχομένων βάσεων δεδομένων στην Xml.

Η δυνατότητα που υπάρχει να τοποθετούνται ένθετα στοιχεία μέσα σε άλλα στοιχεία παρέχει έναν εναλλακτικό τρόπο για να αναπαρίστανται πληροφορίες.

3.1.2 Κανόνες της XML

Τα XML έγγραφα προκειμένου να είναι καλά ορισμένα θα πρέπει να προσαρμόζονται στους ακόλουθους ακριβείς κανόνες σύνταξης:

Τα XML στοιχεία θα πρέπει να έχουν μια ετικέτα αρχής και τέλους.

Όλα τα στοιχεία(elements) πρέπει να έχουν μια ετικέτα ανοίγματος και μια ετικέτα (tag)κλεισίματος. Για παράδειγμα, εάν ένας XML επεξεργαστής διαβάσει ένα αρχείο HTML χωρίς την γνώση αυτού και συναντήσει το ή το <P> ή πολλές άλλες ετικέτες έναρξης, δεν θα είχε κανένα τρόπο για να ξέρει εάν πρέπει ή όχι να αναμένει μια ετικέτα τέλους, πράγμα που καθιστά αδύνατο να γνωρίζει εάν το υπόλοιπο του αρχείου είναι σωστό ή όχι, επειδή έχει χάσει το σημάδι για το αν είναι μέσα σε ένα στοιχείο ή αν έχει τελειώσει με αυτό.

Τα καλοσχηματισμένα XML έγγραφα επομένως απαιτούν ετικέτες έναρξης και ετικέτες τέλους σε κάθε συνηθισμένο αρχείο, και σε οποιοδήποτε κενά (empty) στοιχεία πρέπει να γίνουν σαφή, είτε με χρήση ετικετών τέλους, είτε με την επισύναψη μιας κάθετου στην ετικέτα έναρξης πριν από το κλείσιμο > ως σημάδι ότι δεν θα υπάρξει καμιά ετικέτα τέλους.

Χρήση κενών (empty) Στοιχεία

Η XML επιτρέπει τα κενά στοιχεία με την παρακάτω μορφή σύνταξης:

```
<Project><Project/> Κανονική σύνταξη
```

<Project/> Σύνταξη με σύντμηση

Οποιοδήποτε κενό στοιχείο(πχ. Αυτά που δε χρησιμοποιούν ετικέτα τέλους στην HTML όπως τα ,
) πρέπει είτε να τελειώσουν με />είτε θα πρέπει να μοιάσουν με μη κενά στοιχεία έχοντας μια πραγματική ετικέτα τέλους(χωρίς περιεχόμενο ενδιάμεσα στις ετικέτες). Για παράδειγμα το
 θα γινόταν είτε
 είτε

 (χωρίς να υπάρχει κάτι ενδιάμεσα).

Οι ετικέτες πρέπει να είναι κατάλληλα τοποθετημένες

Τα επικαλυπτόμενα στοιχεία δεν επιτρέπονται. Ένα στοιχείο πρέπει να έχει μια ετικέτα τέλους πριν από την ετικέτα έναρξης του επόμενου στοιχείου και να είναι ενθετημένα το ένα μέσα στο άλλο κατάλληλα.

<i> This text is not correct</i> Λάθος

<i> This text is now correct</i> Σωστό

Οι XML ετικέτες διακρίνουν τα κεφαλαία από τα μικρά γράμματα (case sensitive)

Τα παρακάτω προσδιορίζουν διαφορετικά παραδείγματα

<Crete> <crete> <CRETE>

<City> Heraklion </city> Λάθος

<City> Heraklion </City> Σωστό

Τα XML έγγραφα πρέπει να έχουν ένα στοιχείο ρίζας

Όλα τα XML έγγραφα πρέπει να περιέχουν ένα ενιαίο, μοναδικό ζευγάρι από ετικέτες για να καθορίσουν το στοιχείο ρίζας (root element). Όλα τα άλλα στοιχεία πρέπει να τοποθετηθούν ως ένθετα μέσα στο στοιχείο ρίζας. Όλα τα στοιχεία μπορούν να έχουν (ένθετα) στοιχεία παιδιά (child elements). Τα στοιχεία παιδιά πρέπει να είναι σε ζευγάρια και σωστά τοποθετημένα μέσα στο γονικό τους στοιχείο.

</subchild>

</child>

</root>

Οι τιμές των ιδιοτήτων πρέπει να βρίσκονται σε εισαγωγικά

Ένα στοιχείο μπορεί προαιρετικά να περιέχει μια ή περισσότερες ιδιότητες στην ετικέτα έναρξής τους. Μια ιδιότητα είναι ένα ζευγάρι της μορφής «όνομα-τιμή» που χωρίζεται από ένα σύμβολο ισότητας(=). Οι τιμές μιας ιδιότητας πρέπει πάντα να βρίσκονται σε εισαγωγικά.

<City code="30300">Nafpaktos</City>

Code="30300" είναι η ιδιότητα του στοιχείου <city>.

Οι ιδιότητες χρησιμοποιούνται για να συνδέσουν πρόσθετες, δευτερεύουσες πληροφορίες με ένα στοιχείο. Οι ιδιότητες μπορούν, επίσης να δεχτούν προκαθορισμένες τιμές, ενώ τα στοιχεία δε μπορούν. Κάθε ιδιότητα ενός στοιχείου μπορεί να προσδιοριστεί μόνο μια φορά, αλλά με οποιαδήποτε σειρά.

<message date="9/05/2014"> Σωστό

<message date=9/05/2014> Λάθος

<message ID= "200">

Η ιδιότητα ID μπορεί να χρησιμοποιηθεί για να προσδιορίσει το μήνυμα (message)

<message ID="201">

Όλες οι τιμές των ιδιοτήτων πρέπει να είναι σε εισαγωγικά. Μπορούν να χρησιμοποιηθούν τα μονά εισαγωγικά (η απόστροφος) εάν η τιμή περιέχει έναν χαρακτήρα διπλού εισαγωγικού, και αντίστροφα. Εάν οι χαρακτήρες των εισαγωγικών περιέχονται στα δεδομένα, θα πρέπει στη θέση τους να χρησιμοποιηθούν οι οντότητες ' (στη περίπτωση απλού εισαγωγικού) " (στη περίπτωση διπλού εισαγωγικού).

Ειδικοί χαρακτήρες

Τα δεδομένα δεν πρέπει να περιέχουν δεσμευμένους χαρακτήρες σήμανσης, όπως για τις οντότητες < ή & αντίστοιχα, ενώ η ακολουθία]]> μπορεί μόνο να εμφανιστεί στο τέλος του τμήματος CDATA. Σε περίπτωση που χρησιμοποιείται για οποιοδήποτε άλλο σκοπό πρέπει να δοθεί ως]] >.

3.1.3 Σχήμα ενός εγγράφου XML

Οι βάσεις δεδομένων έχουν σχήματα (schema), που χρησιμοποιούνται για να περιορίζουν τις πληροφορίες που μπορούν να αποθηκευτούν στη βάση δεδομένων και να περιορίσουν τους τύπους δεδομένων των αποθηκευμένων πληροφοριών. Αντίθετα εξ' ορισμού, τα Xml έγγραφα μπορούν να δημιουργηθούν χωρίς κάποιο σχετικό σχήμα. Ένα στοιχείο μπορεί μετά να έχει οποιοδήποτε υποστοιχείο ή ιδιότητα. Ενώ μια τέτοια ελευθερία μπορεί να είναι αποδεκτή εξαιτίας της φύσης που περιγράφει από μόνη της τη μορφή των δεδομένων, δεν είναι σχετικά χρήσιμο όταν πρέπει να γίνει επεξεργασία των Xml εγγράφων ατομικά ως μέρος μιας εφαρμογής, ή ακόμα και όταν μεγάλες ποσότητες σχετικών δεδομένων πρόκειται να μορφοποιηθούν στην Xml.

Παρακάτω περιγράφεται το Document Type Definition(DTD) που περιλαμβάνεται ως μέρος της τυποποίησης XML.

Ο ορισμός τύπου εγγράφου (Document Type Definition-DTD), είναι ένα προαιρετικό μέρος ενός Xml εγγράφου. Ο κύριος σκοπός του DTD είναι να περιορίζει τον τύπο των πληροφοριών που παρουσιάζονται στο έγγραφο. Ωστόσο, το DTD δεν περιορίζει τους τύπους, σε σχέση με τους βασικούς τύπους, όπως ακέραιοι ή συμβολοσειρές. Αντίθετα, περιορίζει μόνο την εμφάνιση υποστοιχείων και ιδιοτήτων μέσα σ' ένα στοιχείο. Το DTD είναι βασικά μια λίστα από κανόνες για το πιο μοτίβο υποστοιχείων θα εμφανίζεται μέσα σε ένα υποστοιχείο.

3.1.4 XPATH

Το XPATH είναι μια γλώσσα ερωτημάτων για την επιλογή κόμβων από ένα XML έγγραφο. Πολλές φορές όμως χρησιμοποιείται για να κάνει τον υπολογισμό τιμών, όπως τιμές Boolean, Strings ή και αριθμούς που βρίσκονται μέσα στο έγγραφο αυτό. Επιπλέον, το XPATH αναφέρει μέρη ενός εγγράφου XML με παραστάσεις διαδρομών. Οι διαδρομές αυτές είναι μια σειρά από βήματα θέσεων που αντί να είναι χωρισμένα με «.» όπως συμβαίνει στην SQL, χωρίζονται με «/». Το αποτέλεσμα μιας παράστασης διαδρομής, δεν είναι τίποτα άλλο, από ένα σύνολο τιμών.

Το σύμβολο «/» που βρίσκεται στην αρχή δείχνει την ρίζα του εγγράφου. Όταν ένα όνομα στοιχείου εμφανίζεται πριν την επόμενη «/» αναφέρεται σε όλα τα στοιχεία του ονόματος, που είναι παιδιά των στοιχείων στο συγκεκριμένο σύνολο από στοιχεία.

Για παράδειγμα, αν δώσουμε την XPath παράσταση :

```
/EPP/Student/Name
```

Η παραπάνω παράσταση θα μας επιστρέψει τα εξής στοιχεία:

```
<Name>Ntouni</Name>
```

```
<Name>Loukou</Name>
```

Η παράσταση :

```
/EPP/Student/Name/text()
```

Θα μας επιστρέψει τα ίδια τα ονόματα χωρίς τις ετικέτες.

Το XPath υποστηρίζει διάφορες λειτουργίες. Μια από αυτές αφορά τα κατηγορήματα επιλογής τα οποία μπορούν να ακολουθούν διάφορα βήματα σε μια διαδρομή, περικλειόμενα από αγκύλες. Τα υποστοιχεία έχουν τη δυνατότητα να πραγματοποιούν ελέγχους με τελεστές σύγκρισης, έτσι ώστε, η παράσταση να επιστρέψει τις επιθυμητές τιμές. Επίσης, το XPath παρέχει διάφορες συναρτήσεις με τις οποίες πραγματοποιεί ελέγχους στους τρέχοντες κόμβους για να δει ποια τιμή ταιριάζει από τους κόμβους – αδερφών. Για παράδειγμα, η παράσταση

```
/ EPP/test/[student/count()>2]
```

Επιστρέφει τα test δυο ή περισσότερων φοιτητών.

Συχνά συναντάμε τον τελεστή «|» ο οποίος επιτρέπει να ενώνονται αποτελέσματα παραστάσεων, όμως ο τελεστής αυτός δεν πρέπει να είναι ένθετος μέσα σε άλλους τελεστές. (Οι τιμές ιδιοτήτων μπορούν να προσπελαστούν χρησιμοποιώντας «@»).

```
/TEI/EPP/id(@ Students) | /TEI/DOMIKON/id(@teachers)
```

Μια XPath παράσταση δίνει τη δυνατότητα να βρίσκονται κάθε φορά τα απαιτούμενα δεδομένα χωρίς πλήρη γνώση του σχήματος χρησιμοποιώντας το «//».

3.1.5 XSLT

Το XSL είναι μια επεκτάσιμη γλώσσα (Extensible Stylesheet Language) η οποία σχεδιάστηκε, ώστε, να καλύψει την ανάγκη για ενσωμάτωση σ' ένα XML αρχείο, HTML κώδικα. Το XSLT αποτελεί ένα ξεχωριστό αρχείο XML το οποίο μετατρέπει το έγγραφο από μια μορφή σε μια άλλη. Διαφορετικές XSL μορφοποιήσεις μπορούν να χρησιμοποιηθούν για να παρουσιάσουν το ίδιο XML έγγραφο με διαφορετικούς τρόπους.

Η συγκεκριμένη γλώσσα διέπεται από κάποιους αναδρομικούς κανόνες που ονομάζονται πρότυπα (template). Τα πρότυπα επιτρέπουν επιλογή κόμβων σε ένα XML δέντρο από μια XPath παράσταση και επιτρέπουν επίσης, τη δημιουργία νέων XML περιεχομένων. Η XSLT παρόλο που χρησιμοποιείται και ως γλώσσα επερωτήσεων έχει πολύ διαφορετική σύνταξη από αυτήν της SQL.

Ένα απλό πρότυπο για XSLT αποτελείται από ένα μέρος match και ένα μέρος select. Το match χρησιμοποιείται για να συνδέσει ένα πρότυπο με ένα στοιχείο XML. Το χαρακτηριστικό αυτό μπορεί επίσης να χρησιμοποιηθεί για να ορίσει ένα πρότυπο για ολόκληρο το έγγραφο XML. Η τιμή του match είναι μια έκφραση XPath (δηλαδή match = "/" προσδιορίζει ολόκληρο το έγγραφο).

Η `<xsl:template>` στοιχείο χρησιμοποιείται για τη δημιουργία προτύπων.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <tr>
      <td>.</td>
      <td>.</td>
    </tr>
  </table>
  </body>
  </html>
</xsl:template>
```

Το στοιχείο `<xsl:value-of>` μπορεί να χρησιμοποιηθεί για την εξαγωγή της τιμής ενός στοιχείου XML για τη παράσταση XPath:

```
</xsl:stylesheet>

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
```



```
<body>
<h2>My CD Collection</h2>
<table border="1">
  <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
  </tr>
  <tr>
    <td><xsl:value-of select="catalog/cd/title"/></td>
    <td><xsl:value-of select="catalog/cd/artist"/></td>
  </tr>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

3.1.6 XQUERY

Η XQuery είναι μια γλώσσα ερωτημάτων(query) που έχει σχεδιαστεί για να μετατρέπει τα ερωτήματα και τα δεδομένα που συλλέγει σε μορφή Xml. Η γλώσσα αυτή αναπτύχθηκε από την Xml ομάδα της διεθνούς επιτροπής W3C. Το βασικό πλεονέκτημα της XQuery είναι πως μας δίνει τη δυνατότητα να εξάγουμε και να χειριζόμαστε τα δεδομένα από έγγραφα XML ή οποιαδήποτε πηγή δεδομένων που μπορεί να θεωρηθεί ως XML, όπως σχεσιακές βάσεις δεδομένων ή έγγραφα γραφείου.

Αντίθετα με την XSLT, η XQuery δεν αναπαριστά ερωτήματα της XML αλλά μοιάζει περισσότερο με SQL. Ερωτήματα που είναι οργανωμένα σε παραστάσεις που αποτελούνται από τέσσερις ενότητες: for, let, where και return. (εν συντομία αναφέρονται και ως “FLWR”). Ας δούμε τι κάνει το καθένα:

- **For:** δίνει μια σειρά από μεταβλητές που διαφέρουν από τα αποτελέσματα των αναπαραστάσεων XPath. Όταν καθορίζονται περισσότερες από μια μεταβλητές, τα αποτελέσματα περιλαμβάνουν το καρτεσιανό γινόμενο των πιθανών τιμών που μπορεί να λάβουν οι μεταβλητές, κάνοντας τον όρο for κάτι σαν το from σ' ένα SQL ερώτημα.
- **Let:** Επιτρέπει να αντιστοιχούνται περίπλοκες παραστάσεις σε ονόματα μεταβλητών προκειμένου να γίνει πιο απλή η αναπαράσταση.
- **Where:** Ακριβώς όπως και ο όρος where της SQL, κάνει επιπλέον ελέγχους για τις συνδεδεμένες εγγραφές από την ενότητα for.
- **Return:** Επιτρέπει την κατασκευή αποτελεσμάτων στην Xml.

3.1.7 Σύγκριση XQuery και XSLT

Αν και η XQuery αρχικά σχεδιάστηκε ως μια γλώσσα επερωτήσεων για μεγάλες συλλογές εγγράφων XML, είναι επίσης ικανή να μετατρέψει μεμονωμένα έγγραφα. Ως εκ τούτου, οι δυνατότητές της συμπίπτουν με την XSLT, η οποία σχεδιάστηκε για να επιτρέψει ρητά έγγραφα εισόδου XML ώστε να μετατραπεί σε HTML ή άλλες μορφές. Τα XSLT 2.0 and XQuery πρότυπα αναπτύχθηκαν από χωριστές ομάδες εργασίας στο πλαίσιο του W3C, που εργάζονται από κοινού για να εξασφαλιστεί μια κοινή προσέγγιση κατά περίπτωση. Μοιράζονται το ίδιο μοντέλο δεδομένων (XDM), σύστημα τύπου, και βιβλιοθήκη συναρτήσεων, και οι δύο περιλαμβάνουν XPath 2.0 ως sublanguage.

Οι δύο γλώσσες, ωστόσο, έχουν τις ρίζες τους σε διαφορετικές παραδόσεις και εξυπηρετούν τις ανάγκες των διαφορετικών κοινοτήτων. Η XSLT είχε κυρίως σχεδιαστεί ως μια stylesheet language, του οποίου πρωταρχικός στόχος ήταν να καταστεί η XML πιο εύκολη και ευανάγνωστη για τον χρήστη XQuery ήταν κυρίως σχεδιαστεί ως μια βάση δεδομένων γλώσσα επερωτήσεων στην παράδοση της SQL. Επειδή οι δύο γλώσσες προέρχονται από διαφορετικές κοινότητες, η XSLT είναι ισχυρότερη κατά το χειρισμό της αφήγησης εγγράφων με πιο ευέλικτη δομή, ενώ η XQuery είναι ισχυρότερη στα δεδομένα του χειρισμού (για παράδειγμα, κατά την εκτέλεση σχεσιακών ενώνων).

Διάφορες επιστημονικές μελέτες που έχουν γίνει κατά καιρούς, έχουν δείξει ότι η XQuery είναι πιο εύκολη στην ανάγνωσή της, ειδικά για κάποιον που έχει ασχοληθεί με SQL σε σχέση με την XSLT. Λόγω της ευελιξίας που προσφέρει η XQuery καθίσταται πιο εύχρηστη από την XSLT, η οποία περιλαμβάνει ένθετα στοιχεία. Ακόμα, η σύνταξη XQuery είναι πιο κατάλληλη για ενσωμάτωση σε παραδοσιακές γλώσσες προγραμματισμού όπως Java ή C#.

Αντίθετα, η XSLT είναι πιο ισχυρή από την XQuery για τις εφαρμογές επειδή επιτρέπει τις μικροαλλαγές σε ένα έγγραφο, όπως για παράδειγμα, την διαγραφή όλων των στοιχείων με ένα συγκεκριμένο όνομα. Μια δυνατότητα που λείπει από την XQuery είναι ο πολυμορφισμός και η δυνατότητα δυναμικής δέσμευσης. Η απουσία αυτής της ικανότητας αρχίζει να γίνεται αισθητή όταν γράφονται μεγάλες εφαρμογές, ή κατά τη σύνταξη κώδικα που έχει σχεδιαστεί για να επαναχρησιμοποιηθεί σε διαφορετικά περιβάλλοντα. Η XSLT παρέχει δύο συμπληρωματικούς μηχανισμούς σε αυτόν τον τομέα, πρώτον τη δυναμική αντιστοίχιση των κανόνων ενός Template, και δεύτερον την ικανότητα να παρακάμπτει τους κανόνες που χρησιμοποιούν `xsl:import`, που καθιστούν δυνατό να γράψουν εφαρμογές με πολλαπλά layers.

Τέλος, παρόλο που αυτές οι δυνατότητες κάνουν την XSLT πιο ισχυρή δεν μειώνουν τη δυναμική της XQuery. Αντιθέτως, προσδίδουν στην γλώσσα το πλεονέκτημα να είναι πιο εύκολη στην ανάλυσή της αλλά και να εντοπίζει τα σφάλματα ευκολότερα κατά τη μεταγλώττισή της.

3.2 Τι είναι HTML

Η HTML (**H**yper**T**ext **M**arkup **L**anguage) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περιλαμβάνονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

```
<!DOCTYPE>  
  
<HTML>  
  
<Head><Title> Naupaktos</Title></Head>  
  
<Body></Body>  
  
</HTML>
```

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Πλεονεκτήματα της HTML

1. Είναι εύκολη στη χρήση της καθώς έχει απλή σύνταξη.
2. Υποστηρίζεται σχεδόν σε κάθε πρόγραμμα περιήγησης, αν όχι όλους.
3. Χρησιμοποιείται ευρέως. Σχεδόν όλες οι ιστοσελίδες χρησιμοποιούν κώδικα HTML.
4. Η σύνταξή της μοιάζει με αυτήν της XML, η οποία χρησιμοποιείται όλο και περισσότερο για την αποθήκευση δεδομένων.
5. Είναι εύκολη στη μάθηση και στη δημιουργία κώδικα ακόμα και για αρχάριους προγραμματιστές.

6. Ο προγραμματιστής – χρήστης έχει τη δυνατότητα να χρησιμοποιήσει τα αντίστοιχα προγράμματα δωρεάν.

Μειονεκτήματα της HTML

1. Κάποιες φορές, η δομή των HTML εγγράφων είναι δύσκολο να κατανοηθούν.
2. Η HTML δε μπορεί να παράγει δυναμική έξοδο από μόνη της, δεδομένου ότι είναι μια στατική γλώσσα.
3. Θα πρέπει να ενημερώνεται για επεξεργασμένες ετικέτες και να φροντίσει να μην τις χρησιμοποιεί.
4. Η HTML έχει κενά ως προς την ασφάλεια των δεδομένων.

3.3 XML vs HTML

Η βασική διαφορά ανάμεσα στην HTML και την XML έγκειται στο γεγονός ότι η πρώτη δίνει έμφαση στην εμφάνιση των δεδομένων, ενώ η δεύτερη έχει ως πρωταρχικό σκοπό την μεταφορά και αποθήκευση των δεδομένων.

Αρχικά, η XML έχει δομημένο κείμενο σε αντίθεση με την HTML που έχει μορφοποιημένο. Η δομή της είναι καθορισμένη από τον χρήστη σε σχέση με την HTML που έχει προκαθορισμένη μορφή. Η XML έχει αποδοτικότερη διασύνδεση υπερκειμένου ενώ η HTML περιορισμένη. Τέλος η XML διαθέτει ένα ευέλικτο και προσαρμοσμένο κείμενο.

	HTML	XML
Ορισμός	Markup γλώσσα για την εμφάνιση ιστοσελίδων σε ένα πρόγραμμα περιήγησης στο web. Σχεδιασμένο για να εμφανίσετε τα δεδομένα με έμφαση στο πώς τα βλέμματα των δεδομένων	Markup γλώσσα ορίζει ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων που μπορεί να διαβαστεί από τους ανθρώπους και μηχανές. Σχεδιασμένο με έμφαση στην αποθήκευση και τη μεταφορά δεδομένων.
Ημερομηνία όταν εφευρέθηκε	1990	1996
Επέκταση από	SGML	SGML
Τύπος	Στατικός	Δυναμικός
Χρήση	Εμφανίστε μια ιστοσελίδα	Μεταφορά δεδομένων μεταξύ της εφαρμογής και της βάσης δεδομένων. Να αναπτύξουν άλλο σήμα μέχρι γλώσσα.
Επεξεργασία / Κανόνες	Δεν υπάρχουν αυστηροί κανόνες. Browser θα εξακολουθούν να παράγουν δεδομένα με τον καλύτερο την ικανότητά του	Αυστηροί κανόνες πρέπει να ακολουθούνται ή επεξεργαστής θα τερματίσει την επεξεργασία των αρχείων
Τύπος Γλώσσα	Παρουσίαση	Ούτε η παρουσίαση, ούτε προγραμματισμός
Ετικέτες	Προκαθορισμένος	Προσαρμοσμένες ετικέτες μπορούν να δημιουργηθούν από τον συγγραφέα
White Space	Δεν μπορεί να διατηρήσει το λευκό κενό	Διατηρεί το λευκό κενό
Περιορισμοί	Τα στοιχεία δεν είναι η ίδια ξέρει πολύ καλά. Τα δεδομένα δεν μπορούν να αλλάζουν ανάλογα με το περιβάλλον. Τα δεδομένα δεν μπορούν να συντηρηθούν εύκολα. Δεν είναι δυνατή η αποθήκευση ή κλήση σε μεταβλητές. Στερείται τη δυνατότητα να καθορίσουν νέες δομές με τον καθορισμό των σχέσεων μεταξύ των τάξεων. Οι ετικέτες δεν είναι χρήσιμα για την ανταλλαγή των εγγράφων μεταξύ των εφαρμογών.	Δεν μπορεί να χρησιμοποιηθεί ως υποτύπου ενός sql_variant παράδειγμα. Δεν υποστηρίζει χύτευση ή τη μετατροπή σε κείμενο ή για ntext. Δεν υποστηρίζει τις ακόλουθες στήλη και τον πίνακα περιορισμούς .XML παρέχει το δικό της κωδικοποίησης. Collations ισχύει μόνο για τους τύπους χορδών. Δεν μπορεί να συγκριθεί ή να ταξινομηθούν. Δεν μπορεί να

		χρησιμοποιηθεί σε Κατανεμημένα υποσυστημάτων Προβολές. Όχι καλά υποστηρίζεται από τους browsers.
--	--	--

3.4 Τι είναι JavaScript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού που αναπτύχθηκε από την εταιρία Netscape , σε συνεργασία με την Sun Microsystems και η πρώτη της έκδοση δημοσιεύτηκε το 1995. Ακολούθησε η αντίστοιχη γλώσσα της Microsoft η οποία ονομάστηκε Jscript και η επόμενη έκδοση της JavaScript που είχε το όνομα ECMAScript που αργότερα όμως καθιερώθηκε με το όνομα που είναι γνωστό μέχρι σήμερα. Σκοπός της είναι να δημιουργεί δυναμικές ιστοσελίδες. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η γλώσσα αυτή, κτίστηκε ουσιαστικά πάνω στο πρότυπο των γλωσσών C, C++ και Java. Όμως η διαφορά της είναι στο ότι διαχειρίζεται τους τύπους δεδομένων πιο χαλαρά (loosely typed) σε σχέση με τη σφικτή διαχείριση τύπων δεδομένων (strongly typed) που γίνεται στις προαναφερόμενες γλώσσες. Στην JavaScript οι μεταβλητές δεν είναι απαραίτητο να έχουν ένα συγκεκριμένο τύπο, αλλά μπορούν να αλλάζουν κατά τη διάρκεια της ζωής τους. Επίσης, κάτι πολύ βασικό που πρέπει να επισημάνουμε, είναι πως η JavaScript δεν έχει καμία σχέση με την Java της Sun Microsystems. Η χρήση του ονόματος JavaScript έγινε για λόγους προώθησης της γλώσσας σε μία εποχή που η εξάπλωση της Java ήταν πολύ μεγάλη. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως τοNode.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Η JavaScript οργανώνει όλα τα στοιχεία μιας web σελίδας σε μια ιεραρχία. Κάθε στοιχείο

της σελίδας βλέπεται σαν αντικείμενο. Κάθε αντικείμενο έχει τις δικές του ιδιότητες (properties) και μεθόδους (methods). Με την βοήθεια τις JavaScript μπορεί ο χρήστης να διαχειριστεί εύκολα τα αντικείμενα. Για να γίνει όλο αυτό κατανοητό απαραίτητη προϋπόθεση είναι να κατανοηθεί η ιεραρχία μιας HTML σελίδας.

Παρακάτω ακολουθεί ένα παράδειγμα κώδικα σε JavaScript:

```
<html>
<head>
<title> ExampleJavaScript</title>
<script LANGUAGE="JavaScript">
<!--actual JavaScript follows bellow
Alert ("Hello!!");
//ending the script→
</script>
</head>
</body>Content of the page</body>
</html>
```

Πλεονεκτήματα JavaScript

Η γλώσσα JavaScript χρησιμοποιείται κυρίως για την εξυπηρέτηση των παρακάτω σκοπών:

- Λιγότερος φόρτος των server: Ο έλεγχος και η επικύρωση των δεδομένων που εισάγονται από τους χρήστες γίνεται από τη μεριά του browser κι έτσι δεδομένα τα οποία δεν είναι σε κατάλληλη μορφή δεν αποστέλλονται στον server. Αυτό όμως δεν σημαίνει ότι ο έλεγχος δεν πρέπει να γίνεται και στη μεριά των εξυπηρετητών καθώς κάποιος χρήστης μπορεί να μην έχει ενσωματωμένη την JavaScript στον browser του ή υπάρχει πιθανότητα να την έχει απενεργοποιήσει.
- Άμεση αλληλεπίδραση με τους χρήστες: Με την χρήση της JavaScript για τον έλεγχο των δεδομένων μειώνονται οι χρόνοι αναμονής του χρηστών αφού αυτοί δεν χρειάζεται να περιμένουν μεγάλα χρονικά διαστήματα επαναφόρτωσης της σελίδας σε περίπτωση που έχουν ξεχάσει να εισάγουν κάποιο δεδομένο ή έχουν εισάγει κάτι λάθος.
- Αυτόματη διόρθωση λαθών: Ένα παράδειγμα που μπορεί να κάνει περισσότερο κατανοητό το πώς μπορεί να χρησιμοποιηθεί η JavaScript με αυτόν τον τρόπο είναι αυτό της ημερομηνίας. Πολλά συστήματα βάσεων δεδομένων αποθηκεύουν δεδομένα ημερομηνιών σε μορφή dd-mm-yyyy. Αν κάποιος χρήστης εισάγει κάποια ημερομηνία σε μορφή dd/mm/yyyy τότε κάτι τέτοιο θα μπορούσε να ανιχνευτεί

αυτόματα από τον browser και να μετατραπεί στην σωστή μορφή πριν τα δεδομένα αποσταλούν στον server.

- Αυξημένη χρηστικότητα: Αυτό επιτυγχάνεται επιτρέποντας στον χρήστη την αλλαγή και αλληλεπίδραση με το γραφικό περιβάλλον χωρίς την επαναφόρτωση της σελίδας. Ένα τέτοιο παράδειγμα είναι τα πτυσσόμενα μενού.
- Αυξημένη δυνατότητα αλληλεπίδρασης: Ένα τέτοιο παράδειγμα είναι όταν τα μενού τα οποία αλληλεπιδρούν όταν ο χρήστης περάσει το mouse πάνω από αυτά – η λειτουργία hover – κάτι το οποίο έχει ως αποτέλεσμα να δημιουργηθεί μία σειρά από γεγονότα τα οποία έχουν προγραμματιστεί να λειτουργούν με έναν συγκεκριμένο τρόπο.
- Καλύτερα γραφικά περιβάλλοντα: Χρησιμοποιώντας την JavaScript μπορούν να συμπεριληφθούν αντικείμενα με λειτουργίες drag-and-drop καθώς και plug-ins, όπως είναι το Flash.
- Ελαφρότερα περιβάλλοντα: Εδώ δεν χρειάζεται να γίνει download ενός μεγάλου αρχείου Java applet ή ενός Flash movie διότι τα προγράμματα γραμμένα σε JavaScript είναι μικρά σε μέγεθος και αποθηκεύονται στη μνήμη του browser μόλις κατέβουν

Μειονεκτήματα JavaScript

- Ένα από τα σημαντικότερα μειονεκτήματα για τη χρήση JavaScript είναι ότι τείνει να φορτώνει τις ιστοσελίδες. Ο κώδικας JavaScript έχει την δυνατότητα να προσθέτει πάρα πολλές γραμμές κώδικα, κάτι που μπορεί να αποτελέσει πρόβλημα. Όμως το πρόβλημα αυτό, μπορεί να λυθεί με την αποθήκευση του κώδικα σε ξεχωριστά αρχεία πηγαίου κώδικα JavaScript που έχουν μια επέκταση .Js .Με αυτόν τον τρόπο καθαρίζεται ο κώδικας της σελίδας, επειδή ο κώδικας JavaScript αποθηκεύεται χωριστά στην ίδια σελίδα HTML, αφήνοντας μια καθαρότερη και πιο εύχρηστη ιστοσελίδα. Αυτή η τάση της JavaScript να φορτώνει τις ιστοσελίδες, μπορεί βέβαια να είναι πολύ επιζήμια για την μηχανή αναζήτησης μιας ιστοσελίδας. Αυτό συμβαίνει γιατί όταν μια μηχανή αναζήτησης φθάνει στο site, ψάχνει για ποιοτικό περιεχόμενο και λέξεις-κλειδιά για να καθορίσει το περιεχόμενο της σελίδας. Επειδή οι πολλές γραμμές κώδικα μπορούν να δημιουργήσουν πρόβλημα, γίνεται τακτοποιημένη αποθήκευση κώδικα JavaScript στα script αρχεία με επέκταση .JS και σύνδεση με το αρχείο δέσμης ενεργειών σε έγγραφα HTML.
- Άλλο ένα πρόβλημα που συναντάται στην JavaScript είναι τα προβλήματα με τα κενά ασφάλειας. Όταν η JavaScript αλληλοεπιδρά με το DOM, υπάρχει κίνδυνος για τους τελικούς χρήστες, επιτρέποντας κακόβουλους φορείς να παραδίδουν σενάρια μέσω του web και την διαχείρισή του σε υπολογιστές-πελάτες. Μπορεί στα σύγχρονα πρότυπα να υπάρχουν περιορισμοί, τόσο στις ιστοσελίδες, όσο και στους browsers, παρ' όλα αυτά, οι κακόβουλοι κώδικες, μπορούν ακόμα να εκτελούνται.

3.4.1 JavaScript με HTML

Για να προσθέσουμε κώδικα JavaScript σε ένα αρχείο HTML υπάρχουν δυο τρόποι. Ο πρώτος τρόπος αναφέρεται στην ενσωμάτωση του κώδικα JavaScript κατευθείαν στο αρχείο HTML. Πιο συγκεκριμένα, μπορεί ο χρήστης με τη βοήθεια των ετικετών <SCRIPT> και </script> να συμπεριλάβουν κώδικα JavaScript απευθείας σε ένα αρχείο HTML. Στο παρακάτω παράδειγμα βλέπουμε τη δημιουργία κώδικα ώστε ο χρήστης να έχει τη δυνατότητα μέσω του κουμπιού κλικ να κάνει οποιοσδήποτε ενέργειας.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
// Δηλώσεις JavaScript πηγαίνετε εδώ
ProcessOrder λειτουργία () {
// Περισσότερα δηλώσεις JavaScript πηγαίνετε εδώ
}
</ script>
</ HEAD>
<BODY>
<FORM NAME = "myForm ">
<INPUT TYPE="button" NAME="processOrder" VALUE="Click για την επεξεργασία σας
order" onClick="processOrder();">
...
</ HTML>
```

Ο δεύτερος τρόπος αναφέρεται στη δημιουργία ξεχωριστού αρχείου JavaScript το οποίο έχει κατάληξη .Js και ενσωματώνεται στο αρχείο HTML με το κατάλληλο path. Στο παρακάτω παράδειγμα, βλέπουμε αυτόν τον τρόπο:

```
<HTML>
<HEAD>
<SCRIPT language="JavaScript" SRC="myJSfile.js">
</ script>
</ HEAD>
<BODY>
<A HREF="javascript:processOrder();"> Κάντε κλικ εδώ για να επεξεργαστούμε σας
Προκειμένου. </ A>
...
</ BODY>
</ HTML>
```

3.5 PHP

Η **PHP** είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο, είναι, δηλαδή, Server-side scripting όπου το πρόγραμμα εκτελείται στον web-server. Η εκτέλεση του προγράμματος γίνεται με τη βοήθεια του μεταγλωττιστή PHP (parser) που πρέπει να έχει εγκατασταθεί στον web server. Η PHP είναι HTML ενσωματωμένη scripting γλώσσα. Αυτό σημαίνει ότι, κώδικας PHP συμπεριλαμβάνεται μέσα σε μια σελίδα HTML. Κατά τη διαδικασία της μεταγλώττισης η PHP αφήνει την σελίδα HTML όπως είναι, μέχρι να συναντήσει ένα από τα ειδικά tags έναρξης PHP «<?» και τότε αρχίζει να μεταγλωττίζει το κείμενο ως κώδικα PHP μέχρι να συναντήσει το επόμενο PHP tag κλεισίματος «>», δηλαδή, οτιδήποτε βρίσκεται έξω από τα tags της PHP μένει όπως είναι. Το αποτέλεσμα της μεταγλώττισης μορφοποιείται σε HTML και όλη η ιστοσελίδα αποστέλλεται στον browser.

Σύμφωνα με τα παραπάνω συμπεραίνεται ότι η PHP είναι διερμηνέας (interpreter), δηλαδή, δεν παράγει εκτελέσιμο αρχείο. Είναι command line scripting, αυτό σημαίνει ότι, ένα PHP script μπορεί να τρέξει χωρίς browser και Server. Χρειαζόμαστε μόνο τον μεταγλωττιστή. Η PHP υποστηρίζει πολλά πρωτόκολλα για επικοινωνία με άλλες υπηρεσίες όπως, IMAP, SNMP, NNTP, HTTP, POP3, καθώς επίσης, και τα socket. Υποστηρίζεται από τους περισσότερους web servers.

Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

3.5.1 Επεκτάσεις αρχείων

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης, ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

3.5.2 Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το OnLine βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter.

Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν. Ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της.

Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

3.5.3 Γράφοντας PHP

- Αρχείο "myfirst.php"

```
<?php  
echo "Hello World";  
?>
```

Η εντολή echo στέλνει στον Web Browser κείμενο. Οι εντολές στην PHP τελειώνουν με «;» όπως και οι εντολές στην C++.

- Σχόλια στην PHP

```
<?php  
// Αυτό είναι σχόλιο  
*/  
Σχόλια πολλών γραμμών  
*/  
?>
```
- Μεταβλητές στην PHP
Όλες οι μεταβλητές στην PHP ξεκινούν με το σύμβολο \$. Οι μεταβλητές μπορούν να περιέχουν, πίνακες και τιμές τύπου true-false.
- Κανόνες ονοματολογίας

Ένα όνομα μεταβλητής ξεκινά από γράμμα ή κάτω παύλα «_». Περιέχει μόνο αλφαριθμητικούς χαρακτήρες και κάτω παύλα «_» πχ (a-Z, 0-9, και _). Δεν περιέχει κενά πχ \$my_String ή \$myString και όχι \$my String.

3.5.4 Πλεονεκτήματα PHP

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερισιμότητα
- Διαθεσιμότητα του κώδικα προέλευσης
- Είναι εύκολα προσβάσιμη
- Τρέχει σε πολλά διαφορετικά λειτουργικά συστήματα
- Λειτουργεί στον Server (δεν χρειάζεται ο client – μόνο με θέματα rendering , εάν χρησιμοποιείται CSS).
- Πολύ εύκολη η πρόσβαση σε άλλα web-based εργαλεία μέσω της PHP (πχ Google maps, κτλ).

3.5.5. Μειονεκτήματα PHP

- Ασφάλεια: Επειδή η php είναι ανοιχτή γλώσσα, όλοι οι άνθρωποι (προγραμματιστές και χρήστες) μπορούν να δουν τον πηγαίο κώδικά της. Εφόσον υπάρχουν σφάλματα στον πηγαίο κώδικα, μπορεί να χρησιμοποιηθεί από τους ανθρώπους για να εξερευνηθεί η αδυναμία της PHP
- Δεν είναι κατάλληλη για μεγάλες εφαρμογές: Είναι αρκετά δύσκολο να διατηρηθεί, δεδομένου ότι δεν είναι σπονδυλωτή (modular)
- Ασθενής Τύπος: Υποκρυπτόμενη μετατροπή μπορεί να εκπλήξει τους απρόσεκτους προγραμματιστές και να οδηγήσει σε απρόβλεπτα σφάλματα.
- Για να δημιουργηθεί κάτι παραπάνω από μια απλή HTML / CSS σελίδα, θα πρέπει να προστεθεί κώδικας JavaScript, Java ή κάποια άλλη client-side γλώσσα (ισχύει και για Perl και για άλλες γλώσσες)
- Η PHP τείνει να εκτελεί πιο αργά άλλες γλώσσες, όπως από την C και την Assembly.

3.6 Τι είναι CSS

Η CSS (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) ή αλλιώς (αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ και χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Με άλλα λόγια η CSS είναι μια γλώσσα υπολογιστή που έχει σαν στόχο να αναπτύσσει στυλιστικά μια ιστοσελίδα, δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά. Όπως για παράδειγμα, χρώματα ή στοίχιση. Η γλώσσα αυτή δίνει περισσότερες δυνατότητες σε σχέση με την html. Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του Web(Web design) επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων μονομιάς. Σαν Web developer μπορεί να οριστεί ένα στυλ για κάθε HTML στοιχείο και να εφαρμοστεί σε όσες ιστοσελίδες κρίνεται απαραίτητο.. Για να γίνει καθολική (global) αλλαγή, απλά γίνεται αλλαγή στο στυλ μια φορά και όλα τα στοιχεία του Web ενημερώνονται αυτόματα.

Η Αλληλουχία εφαρμογής των φύλλων στυλ

Για ένα έγγραφο πχ xhtml θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το Φύλλο στυλ (Μηχανισμός καθορισμού και τροποποίησης μμορφοποίησης εγγράφων από τους συγγράφεις) που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

- το συγγραφέα μιας ιστοσελίδας
- το χρήστη του πλοηγού
- τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Κατά συνέπεια, για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες. Το πρότυπο css για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί, όπως για παράδειγμα η δήλωση που είναι πρώτη στη σειρά.

Η Σύνταξη των CSS

Η σύνταξη των CSS αποτελείται από τρία μέρη : έναν επιλογέα (selector), μια ιδιότητα (property) και μια τιμή (value) :

επιλογέας {ιδιότητα: τιμή}

selector {property: value}

Ο *επιλογέας* είναι συνήθως το στοιχείο/tag που θέλουμε να ορίσουμε, η *ιδιότητα* είναι το χαρακτηριστικό που θέλουμε να αλλάξουμε και η κάθε ιδιότητα μπορεί να πάρει μια *τιμή*. Η ιδιότητα και η τιμή ξεχωρίζουν από τον χαρακτήρα : και περικλείονται από τους χαρακτήρες { }, ως εξής :

```
body {color: black}
```

Αν η τιμή αποτελείται από πολλές λέξεις, πρέπει να τοποθετήσουμε εισαγωγικά :

```
p {font-family: "sans serif" }
```

Αν θέλουμε να ορίσουμε περισσότερες από μία ιδιότητες, πρέπει να ξεχωρίσουμε την κάθε ιδιότητα με τον χαρακτήρα ;. Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να ορίσουμε μια κεντραρισμένη παράγραφο:

```
p {text-align: center; color: red}
```

Για να κάνουμε τους ορισμούς των στυλ πιο ευανάγνωστους, μπορούμε να γράψουμε από μία ιδιότητα σε κάθε γραμμή, ως εξής :

```
p  
{  
  
    text-align: center;  
  
    color: black;  
  
    font-family: arial  
  
}
```

Παράδειγμα κώδικα CSS

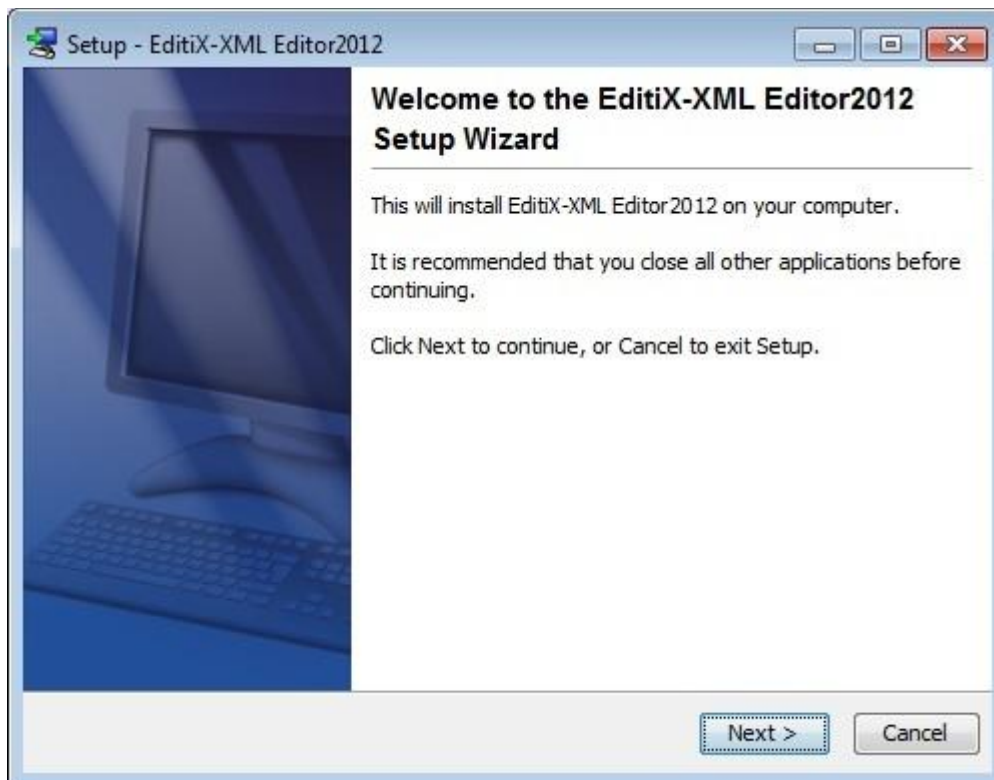
Το αρχείο του εξωτερικού φύλλου στυλ είναι το εξής:

```
body {background-color: yellow}  
  
h1 {font-size: 36pt}  
  
h2 {color: blue}  
  
p {margin-left: 50px}
```

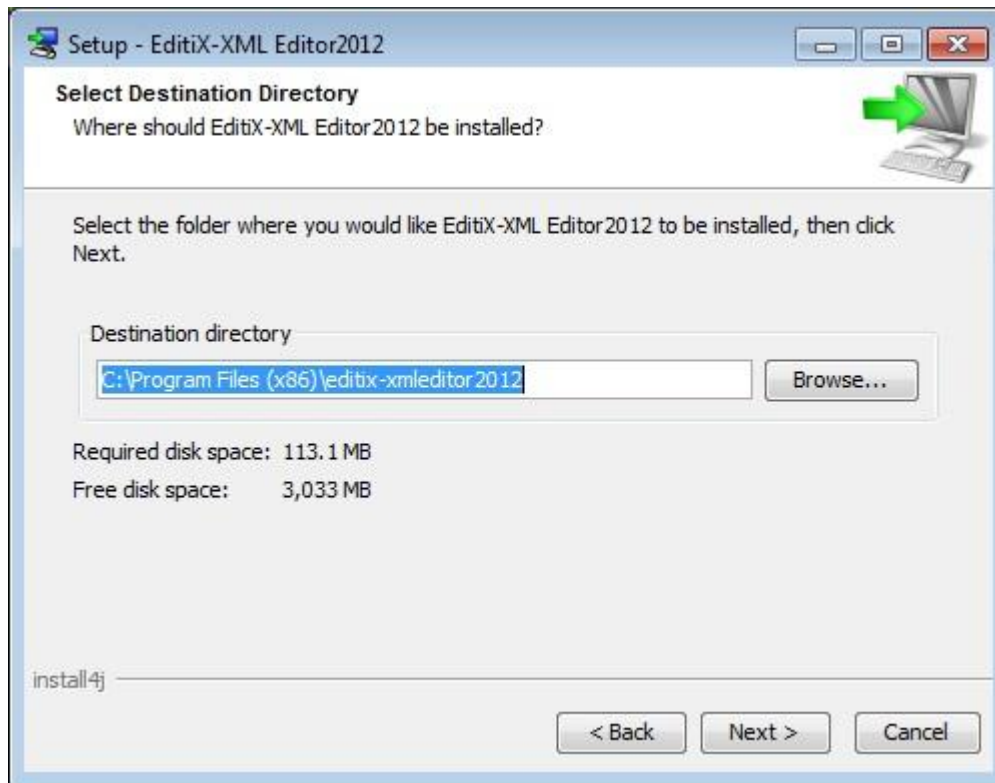
3.6.1 Editix

Το Editix XML Editor είναι ένα ολοκληρωμένο πρόγραμμα επεξεργασίας XML εγγράφων. Παρέχει τη δυνατότητα στο χρήστη να δημιουργήσει μια άρτια βάση δεδομένων με τη χρήση της XML. Επιπλέον, μέσα από διάφορες λειτουργίες του προγράμματος μπορεί να προστεθεί κώδικας σ' ένα έτοιμο Template ή να δημιουργηθεί από την αρχή και να συνδεθεί με τη βάση. Ακόμα, ο χρήστης μπορεί να προσθέσει και κώδικα Html.

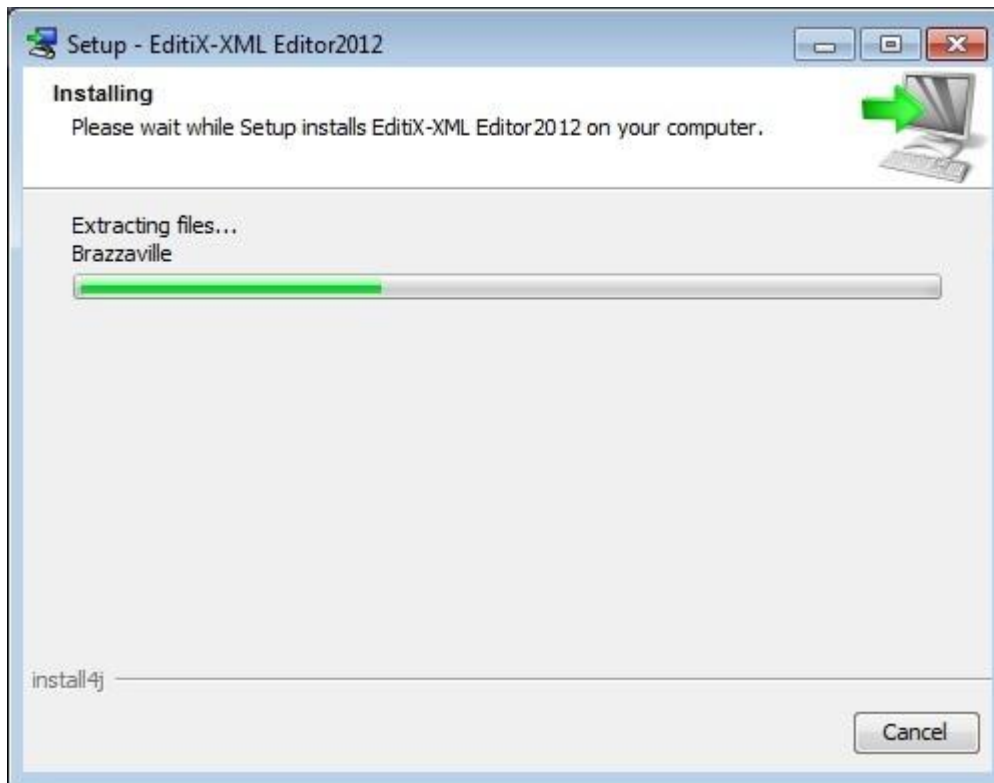
Ο συγκεκριμένος Editor προσφέρει μια σειρά από οδηγούς για διάφορους τύπους εγγράφων (DTD , XSLT , XSL-FO , Relax NG). Ο XSLT Debugger είναι διαθέσιμος για τις εκδόσεις 1.0 και 2.0. Το πρόγραμμα δίνει τη δυνατότητα στοίχισης κειμένου- κώδικα και συνεπώς κάνει την ανάγνωσή του πιο εύκολη. Επίσης, μέσα από το πρόγραμμα αυτό υπάρχει η δυνατότητα ελέγχου συντακτικών λαθών. Ο έλεγχος σύνταξης του κώδικα γίνεται κατά τη διάρκεια της συγγραφής του. Αυτό το πρόγραμμα επεξεργασίας XML εγγράφων παρέχει σε πραγματικό χρόνο τη δημιουργία Xpath κώδικα μέσα από ένα ήδη έτοιμο XML αρχείο αλλά και τη δημιουργία XQuery. Παρακάτω βλέπουμε βήμα – βήμα την εγκατάσταση του προγράμματος:



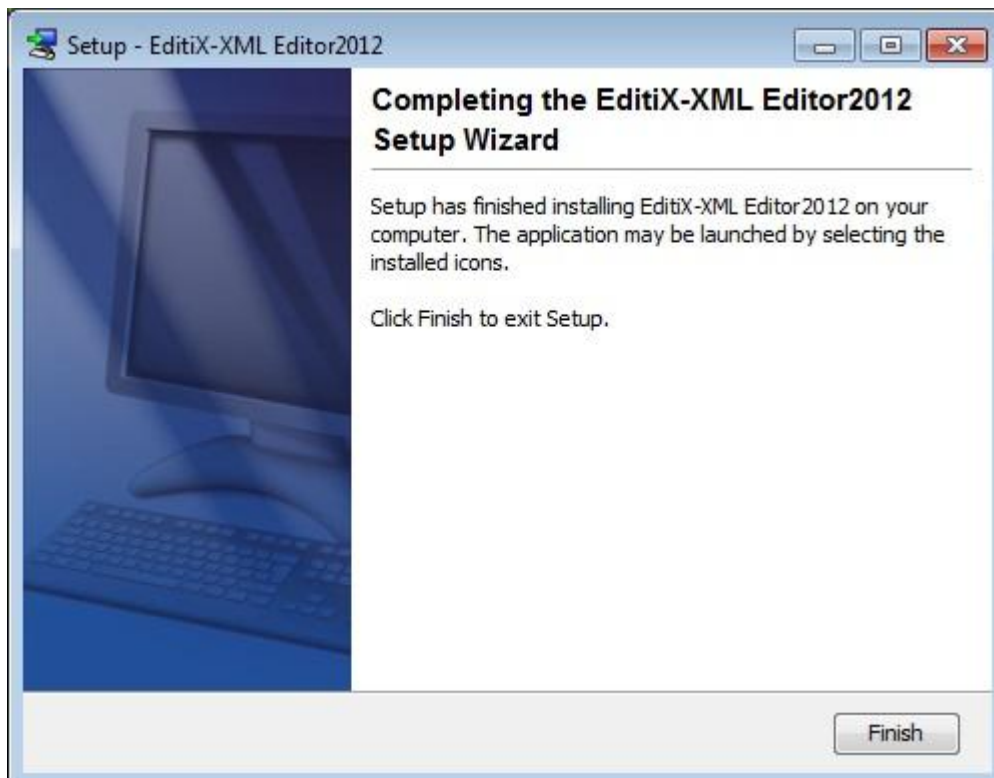
Εικόνα 1 Εγκατάσταση



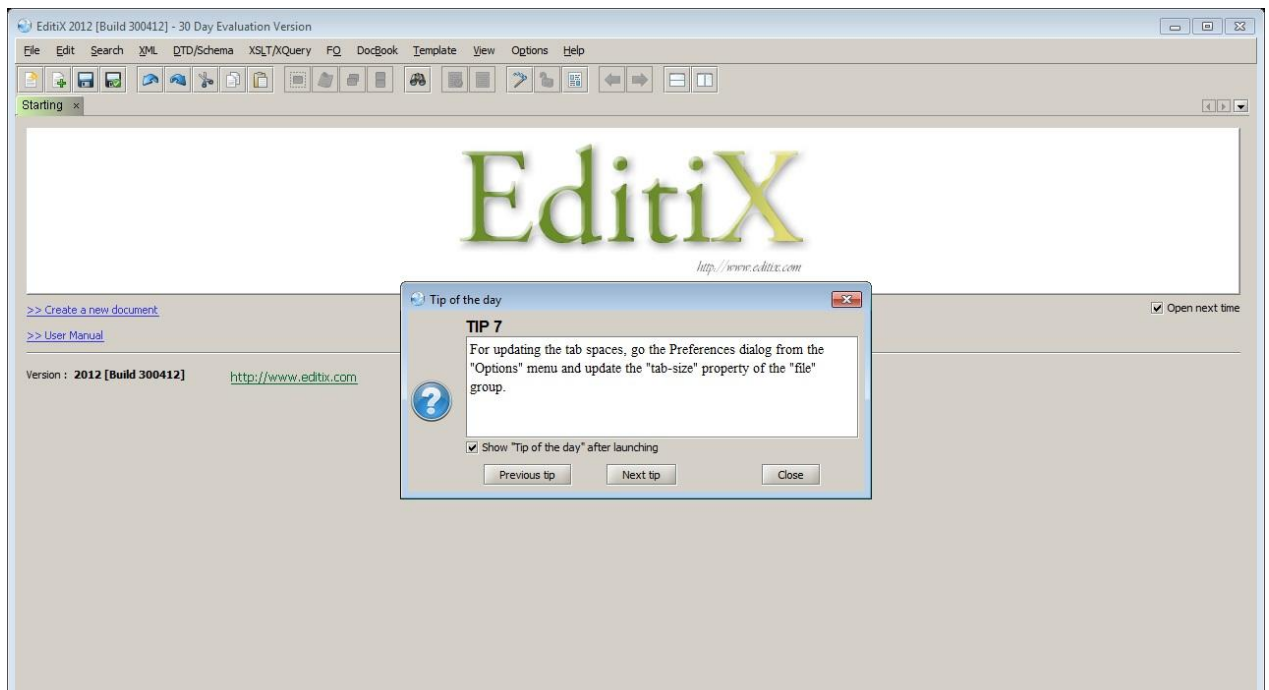
Εικόνα 2



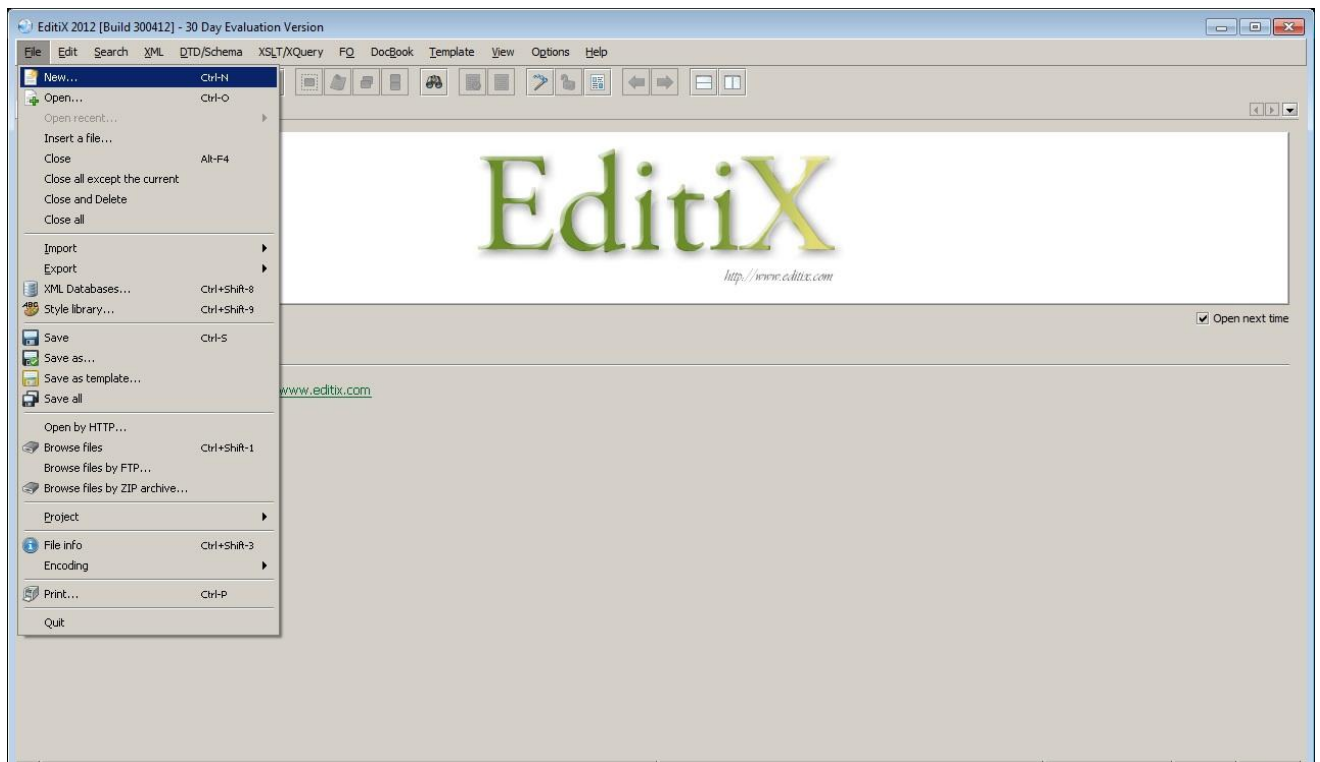
Εικόνα 3



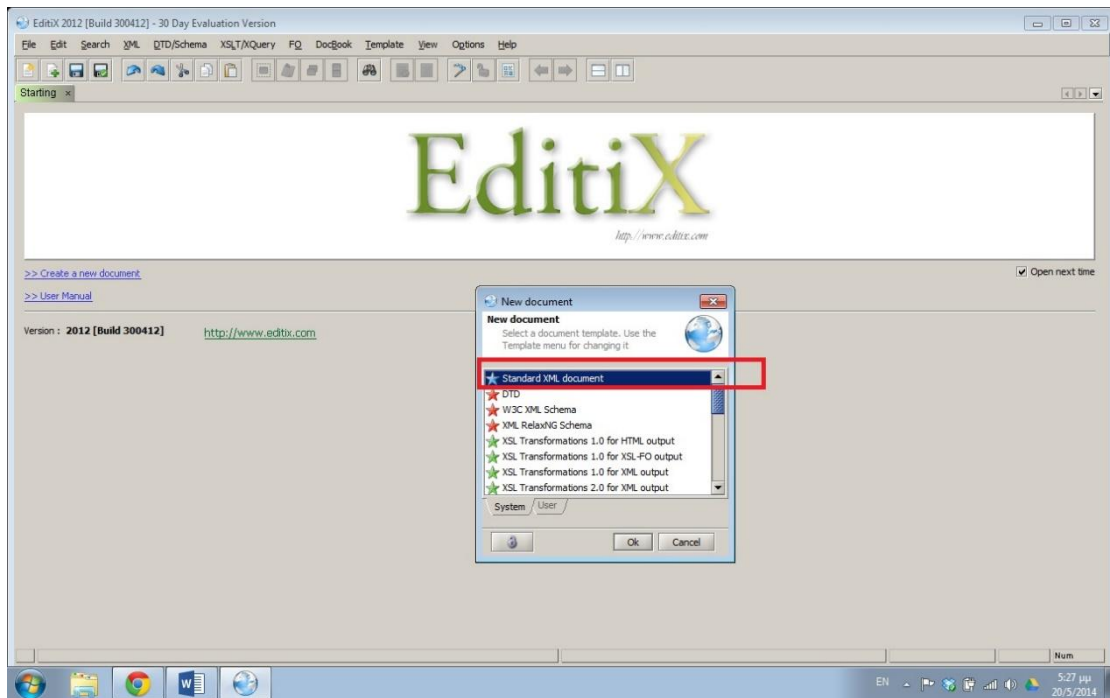
Εικόνα 4



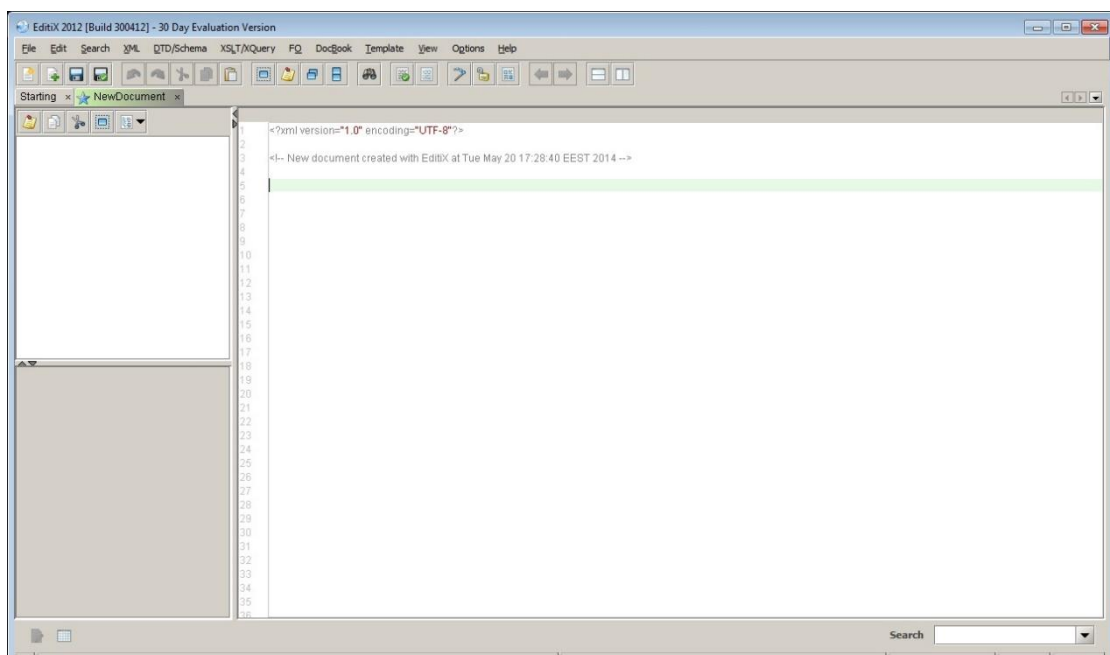
Εικόνα 5



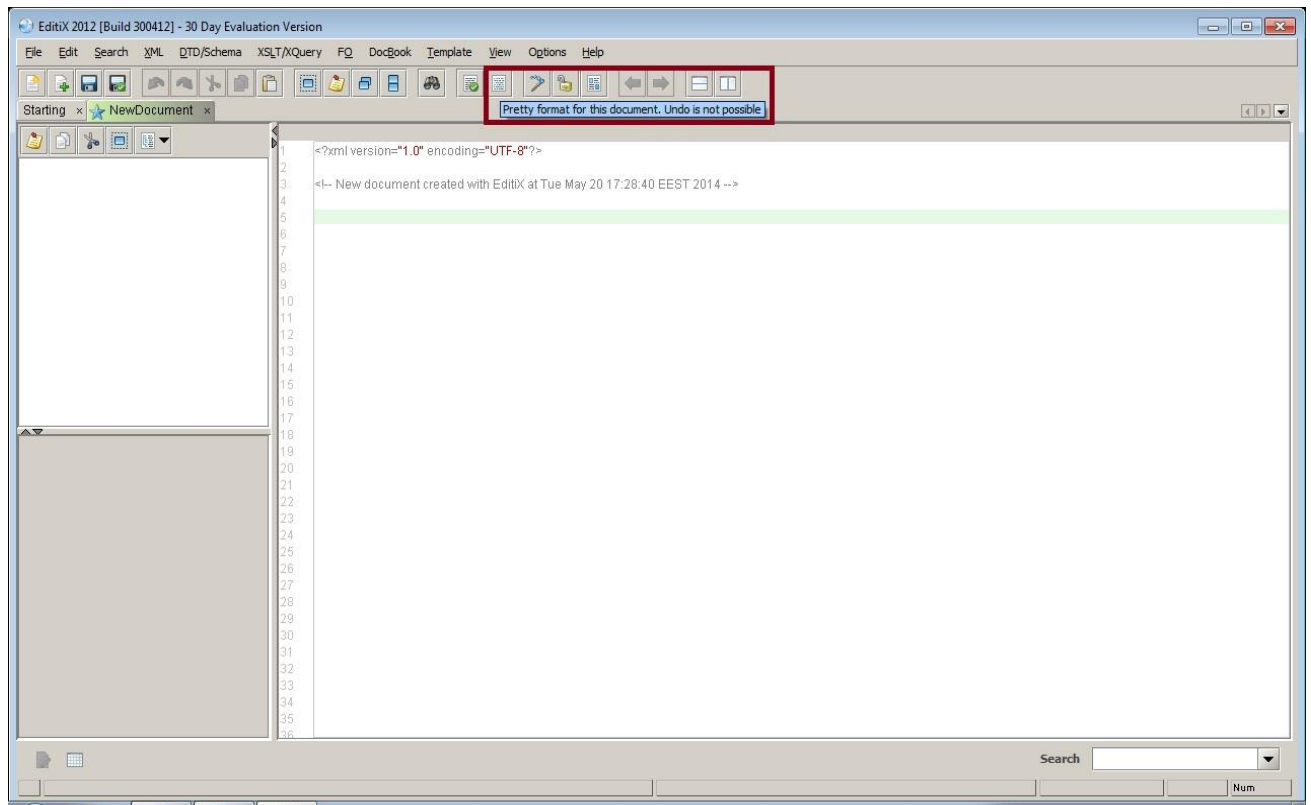
Εικόνα 6 Επιλέγουμε File New



Εικόνα 7 Επιλέγουμε Standard Xml document



Εικόνα 8 Εισαγωγή κώδικα



Εικόνα 9 Μορφοποίηση κώδικα

3.7 PhpMyAdmin

Το PhpMyAdmin είναι ένα δωρεάν open source εργαλείο το οποίο είναι γραμμένο σε php. Σκοπός του είναι να διαχειρίζεται την βάση της MySQL με την χρήση ενός προγράμματος περιήγησης web. Το πρόγραμμα έχει τη δυνατότητα να εκτελεί διάφορες εργασίες, όπως για παράδειγμα, τη δημιουργία, τη τροποποίηση ή τη διαγραφή δεδομένων, πεδίων, πίνακες, γραμμές ή και την διαχείριση των χρηστών και τα δικαιώματα.

3.7.1 PhpMyAdmin

Το πρόγραμμα περιλαμβάνει μια σειρά από δυνατότητες οι οποίες είναι οι εξής:

1. Διασύνδεση Web
2. Διαχείρισης βάσεων δεδομένων MySQL
3. Εισαγωγή δεδομένων από CSV και SQL

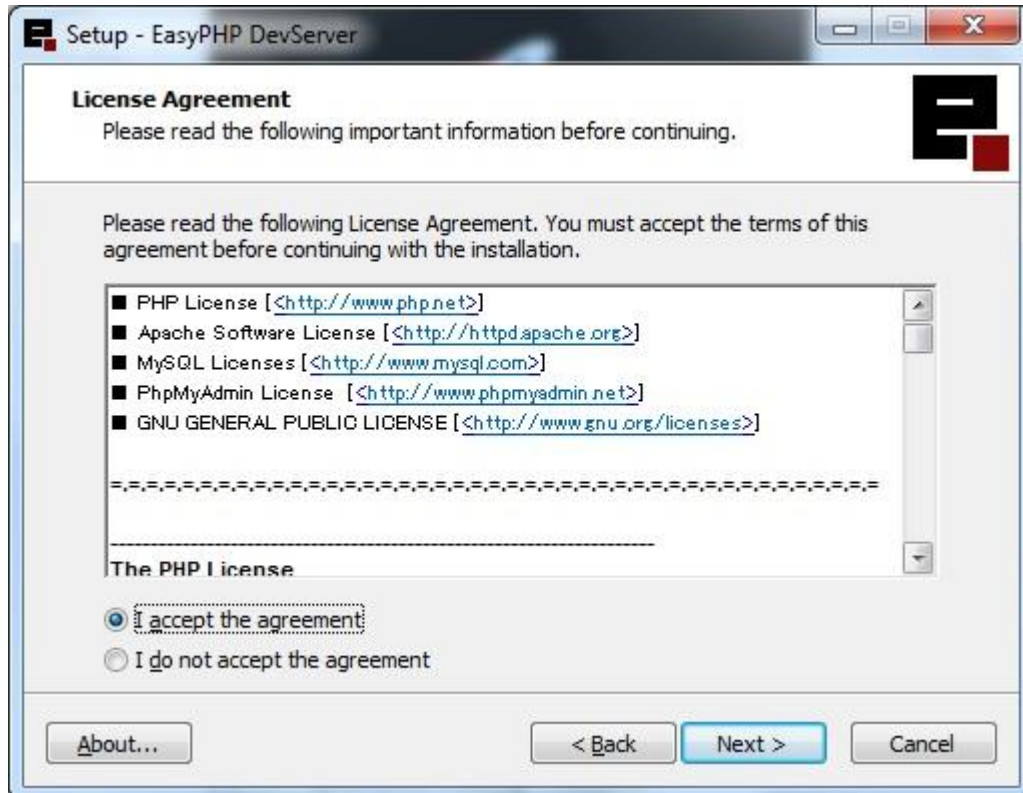
4. Εξαγωγή δεδομένων σε διάφορες μορφές: CSV , SQL , XML , PDF (μέσω του TCPDF βιβλιοθήκη), ISO / IEC 26300 - OpenDocument Text και υπολογιστικού φύλλου, το Word, το Excel, LaTeX και άλλοι
5. Η χορήγηση πολλαπλών servers
6. Δημιουργία γραφικών PDF της διάταξης βάσης δεδομένων
7. Δημιουργία σύνθετων αναζητήσεων χρησιμοποιώντας το Query-by-Παράδειγμα (QBE)
8. Αναζήτηση σε παγκόσμιο επίπεδο σε μια βάση δεδομένων ή ένα υποσύνολο αυτό
9. Μετασχηματισμός αποθηκευμένων δεδομένων σε οποιαδήποτε μορφή, χρησιμοποιώντας ένα σύνολο προκαθορισμένων λειτουργιών, όπως η εμφάνιση BLOB δεδομένων ως εικόνα ή download-link
10. Ζωντανά διαγράμματα για την παρακολούθηση της δραστηριότητας του διακομιστή MySQL, όπως συνδέσεις, τις διαδικασίες, CPU / χρήση της μνήμης, κλπ.



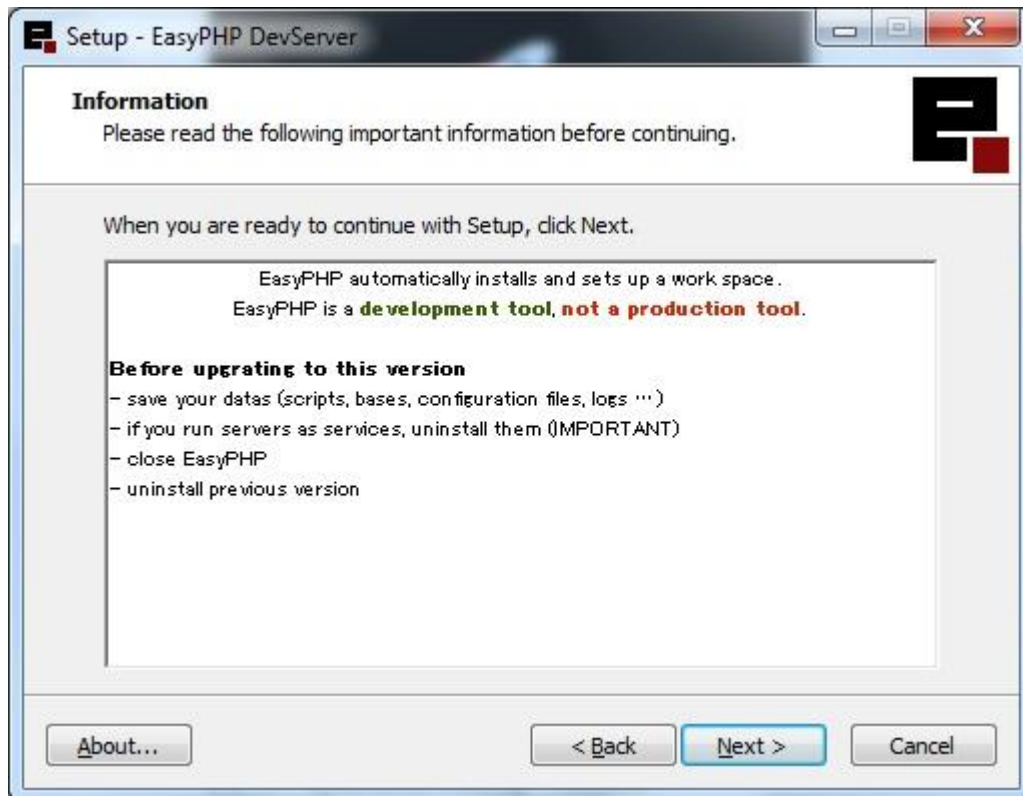
Εικόνα 10 *phpmyadmin install*



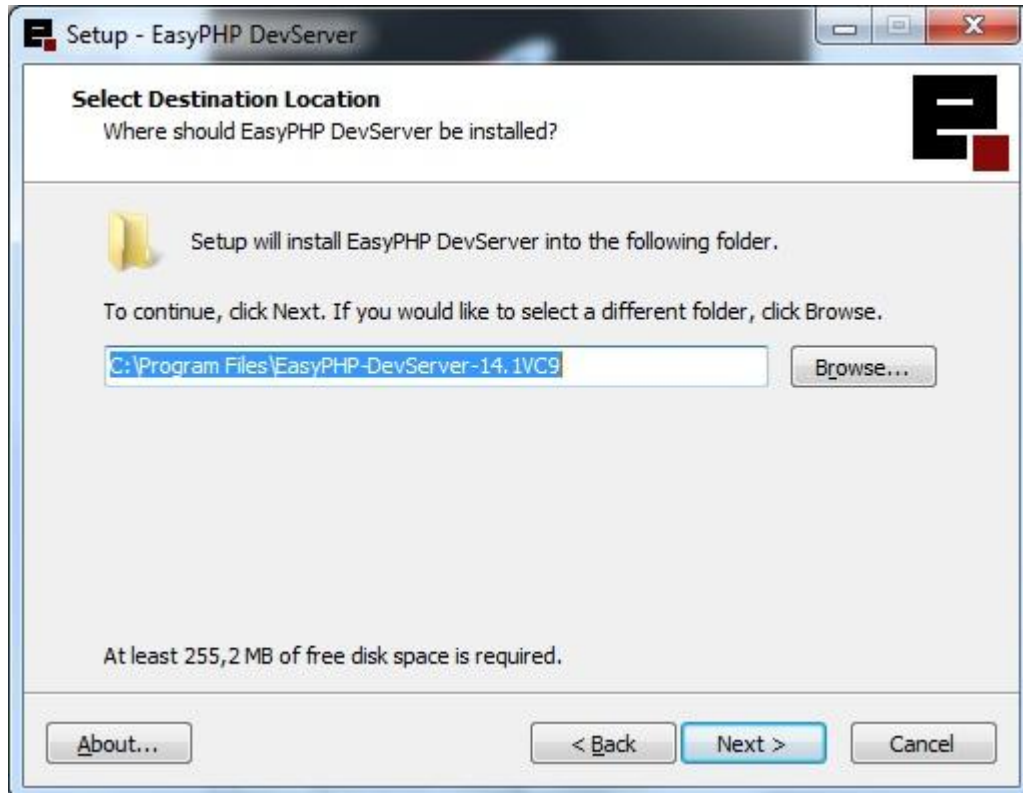
Εικόνα 11 Phpmyadmin install



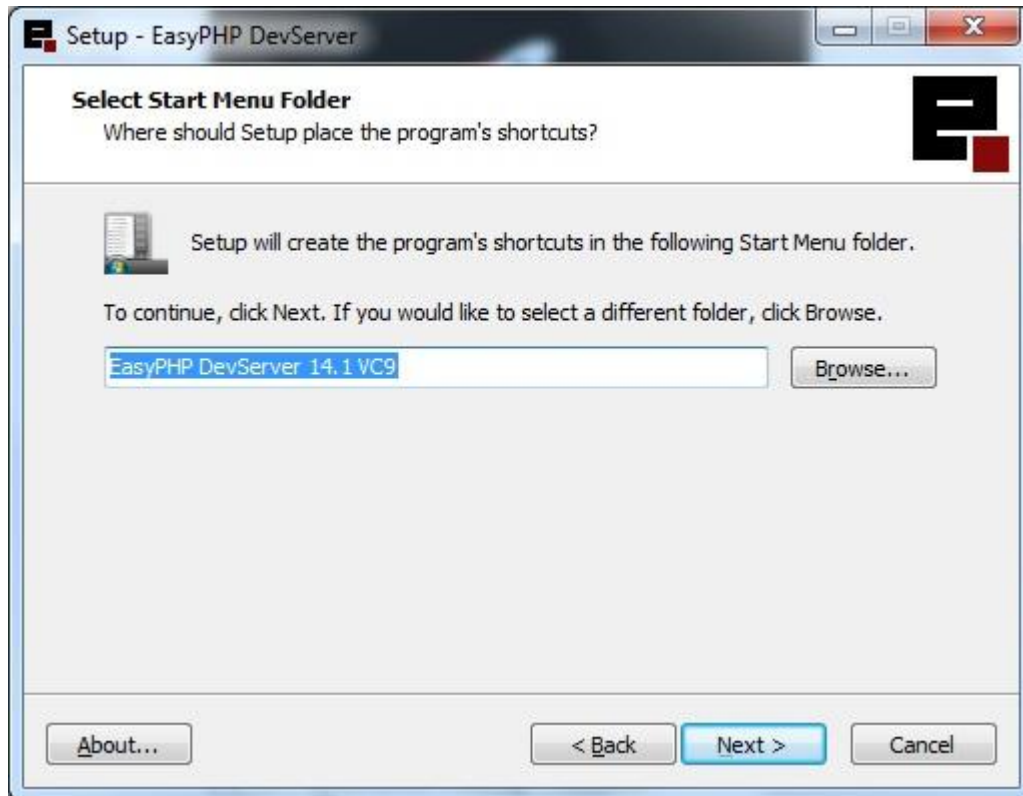
Εικόνα 12 phpmyadmin install



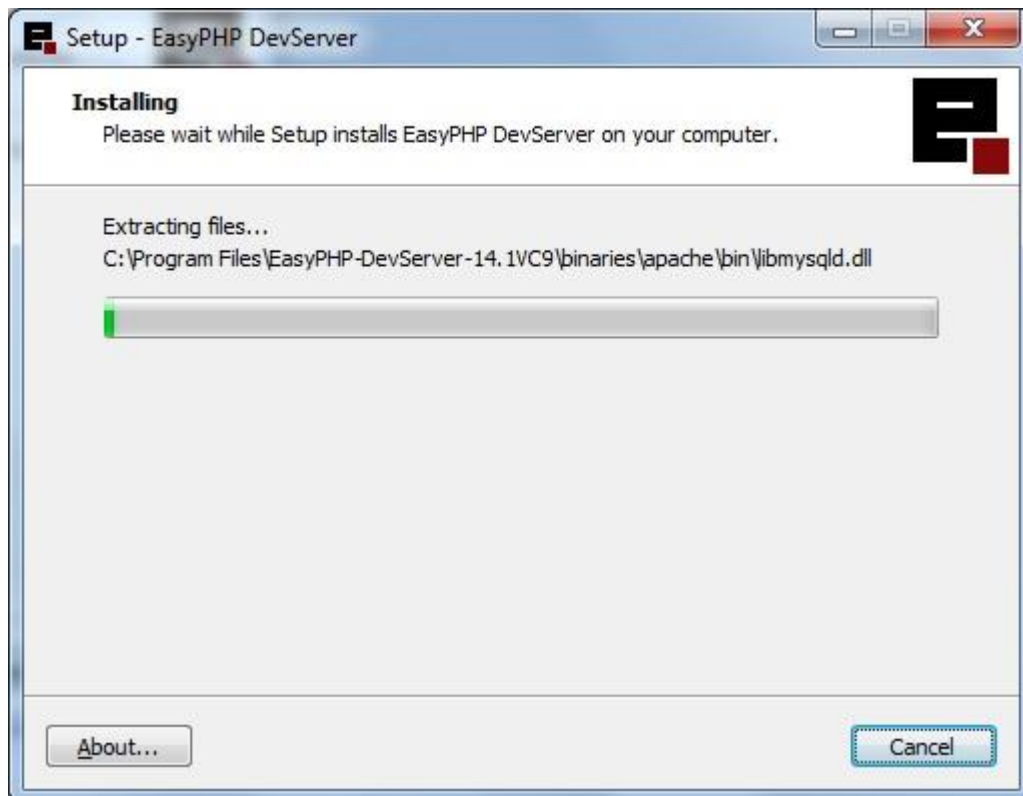
Εικόνα 13 phpmadmin install



Εικόνα 14 αποθήκευση στο C



Εικόνα 15 install



Εικόνα 16 install



Εικόνα 17 ολοκλήρωση εγκατάστασης

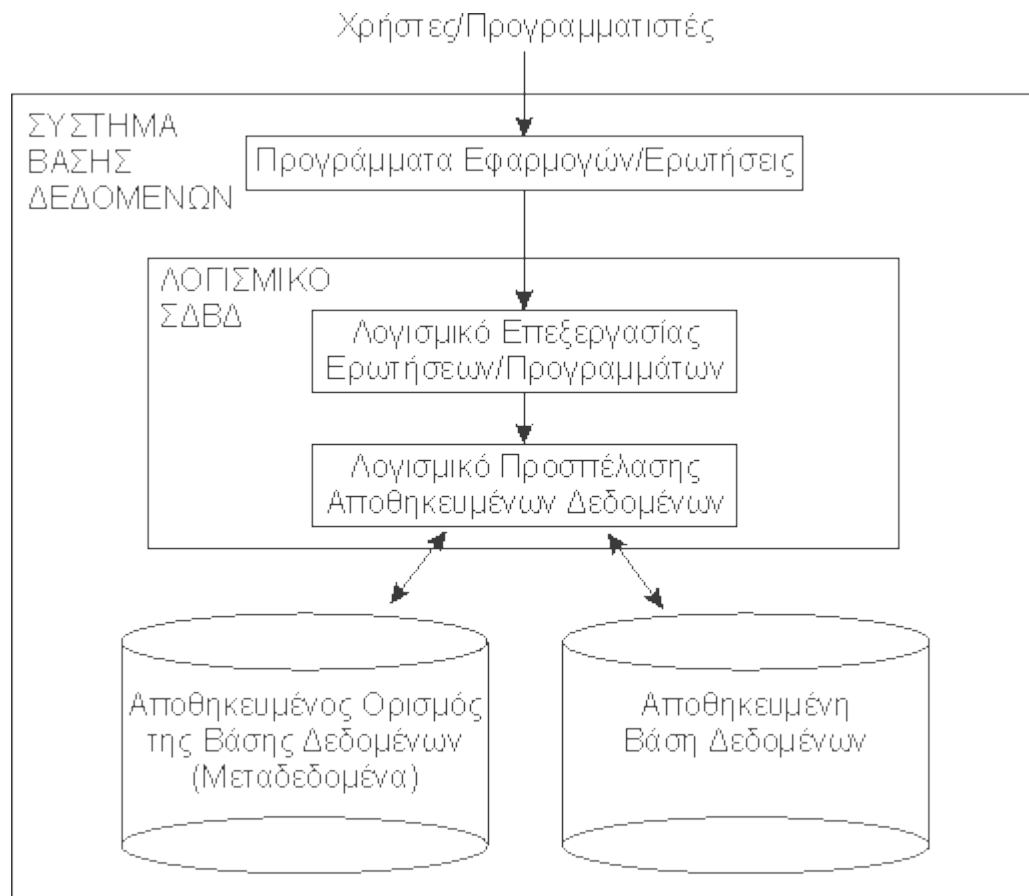
4 Σχεδιασμός Βάσης-Περιγραφή

Μια Βάση Δεδομένων(Database) είναι ένας οργανωμένος τρόπος αποθήκευσης πληροφοριών και πρόσβασή τους με πολλούς τρόπους, με διάφορα προγράμματα. Μια βάση δεδομένων είναι κάτι παραπάνω από μια απλή συλλογή αποθηκευμένων στοιχείων. Σαν βάση δεδομένων, θα μπορούσε επίσης να χαρακτηριστεί ένα ολοκληρωμένο σύστημα που αποτελείται από δεδομένα (data) και από το κατάλληλο λογισμικό(software), τα οποία χρησιμοποιώντας το υλικό (hardware) βοηθούν στην ενημέρωση και πληροφόρηση των χρηστών.

Ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων αποκαλείται Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, DataBase Management System) και με την βοήθειά του μπορούμε να αποθηκεύσουμε, προσθέσουμε, τροποποιήσουμε, εμφανίσουμε ή και διαγράψουμε τα αποθηκευμένα δεδομένα.

Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι :

- Ολοκληρωμένα (Integrated), δηλ. τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός (redundancy) των ίδιων στοιχείων.
- Καταμεριζόμενα (Shared), δηλ. να μπορούν περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.



4.1 Τι είναι το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) ή Database Management System (DBMS);

ΣΔΒΔ καλείται ένα πρόγραμμα λογιστικού υπολογιστών που σχεδιάζεται για να διαχειρίζονται όλες οι βάσεις δεδομένων που υπάρχουν σε σκληρό δίσκο ή ένα δίκτυο συστημάτων. Εξάλλου, μπορεί να οριστεί ως ένα λογισμικό όπου μέσω προγράμματα υπολογιστών που ελέγχουν τη δημιουργία, τη συντήρηση, και τη χρήση μιας βάσης δεδομένων. Επιτρέπει σε οργανισμούς την ανάπτυξη εύχρηστων βάσεων δεδομένων για τις διάφορες εφαρμογές από τους διαχειριστές των.

4.2 Στόχοι μιας Βάσης Δεδομένων

Οι στόχοι μιας βάσης δεδομένων είναι οι εξής :

- Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
- Ο καταμερισμός δεδομένων.
- Η επιβολή κανόνων ασφαλείας (security).
- Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
- Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

- (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
- Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.

4.3 Εφαρμογές συστημάτων βάσεων δεδομένων

Οι βάσεις δεδομένων χρησιμοποιούνται ευρέως όπως για παράδειγμα:

- Τράπεζες: Για πληροφορίες πελατών, λογαριασμών και δανείων και τραπεζικών συναλλαγών.
- Αεροπορικές εταιρίες: Για κρατήσεις θέσεων και πληροφορίες πτήσεων
- Πανεπιστήμια: Για πληροφορίες φοιτητών, εγγραφές σε μαθήματα και βαθμούς
- Συναλλαγές πιστωτικών καρτών: Για αγορές μέσω πιστωτικών καρτών και δημιουργία μηνιαίων κινήσεων
- Τηλεπικοινωνίες: Για διατήρηση των κλήσεων, δημιουργία μηνιαίων λογαριασμών, διατήρηση του υπολοίπου για τις προπληρωμένες κάρτες κλήσης και αποθήκευση πληροφοριών για τα δίκτυα επικοινωνιών
- Χρηματοδοτήσεις: Για αποθήκευση πληροφοριών σχετικά με πωλήσεις και αγορές οικονομικών στοιχείων, όπως μετοχών και ομολόγων
- Πωλήσεις: Για πληροφορίες πελατών, προϊόντων και πωλήσεων
- Βιομηχανία: Για διαχείριση της αλυσίδας προμηθειών και την παρακολούθηση της παραγωγής των προϊόντων σε εργοστάσια, των προϊόντων σε μεγάλες αποθήκες και σε καταστήματα και των παραγγελιών των προϊόντων.
- Ανθρώπινοι πόροι: Για πληροφορίες για εργαζόμενους, μισθούς, φόρους μισθοδοσίας, παροχές και για πληρωμές μισθών.

Η επανάσταση του internet στα τέλη της δεκαετίας του ενενήντα, αύξησε σαφώς την άμεση πρόσβαση του χρήστη στις βάσεις δεδομένων. Οι εταιρίες μετέτρεψαν πολλές από τις διασυνδέσεις τηλεφώνων με βάσεις δεδομένων, σε web διασυνδέσεις και έκαναν διαθέσιμες πολλές υπηρεσίες και πληροφορίες online. Για παράδειγμα, όταν ένας χρήστης έχει πρόσβαση σε ένα online βιβλιοπωλείο και κοιτάζει μια συλλογή βιβλίων ή μουσικής, αυτόματα προσπελαίνει δεδομένα που είναι αποθηκευμένα σε μια βάση δεδομένων. Ακόμα όταν δίνει μια παραγγελία online η παραγγελία του αποθηκεύεται σε μια βάση δεδομένων.

4.4 Περιγραφή συστήματος

Η υλοποίηση του online τουριστικού οδηγού της Ναυπάκτου σχεδιάστηκε με σκοπό να εξυπηρετήσει τις ανάγκες των χρηστών που θέλουν να ενημερωθούν για την ίδια τη πόλη, αλλά και για τις παροχές που τους προσφέρει. Προκειμένου να σχεδιαστεί ένα τέτοιο σύστημα χρειάστηκε να γίνουν μια σειρά από ενέργειες. Αρχικά, σχεδιάσαμε τη βάση δεδομένων σε XML με τη βοήθεια του Editix. Για να στηθεί ένα δυναμικό Web Site απαραίτητη προϋπόθεση είναι η δημιουργία της βάσης, όπου εκεί θα γίνεται η καταγραφή των δεδομένων, καθώς και η διεκπεραίωση συγκεκριμένων διεργασιών. Για παράδειγμα, ο χρήστης μέσα από τον τουριστικό οδηγό θα έχει τη δυνατότητα να κάνει μια online κράτηση σε κάποιο ξενοδοχείο της αρεσκείας του, να ενοικιάσει κάποιο όχημα ή να τα ακυρώσει.

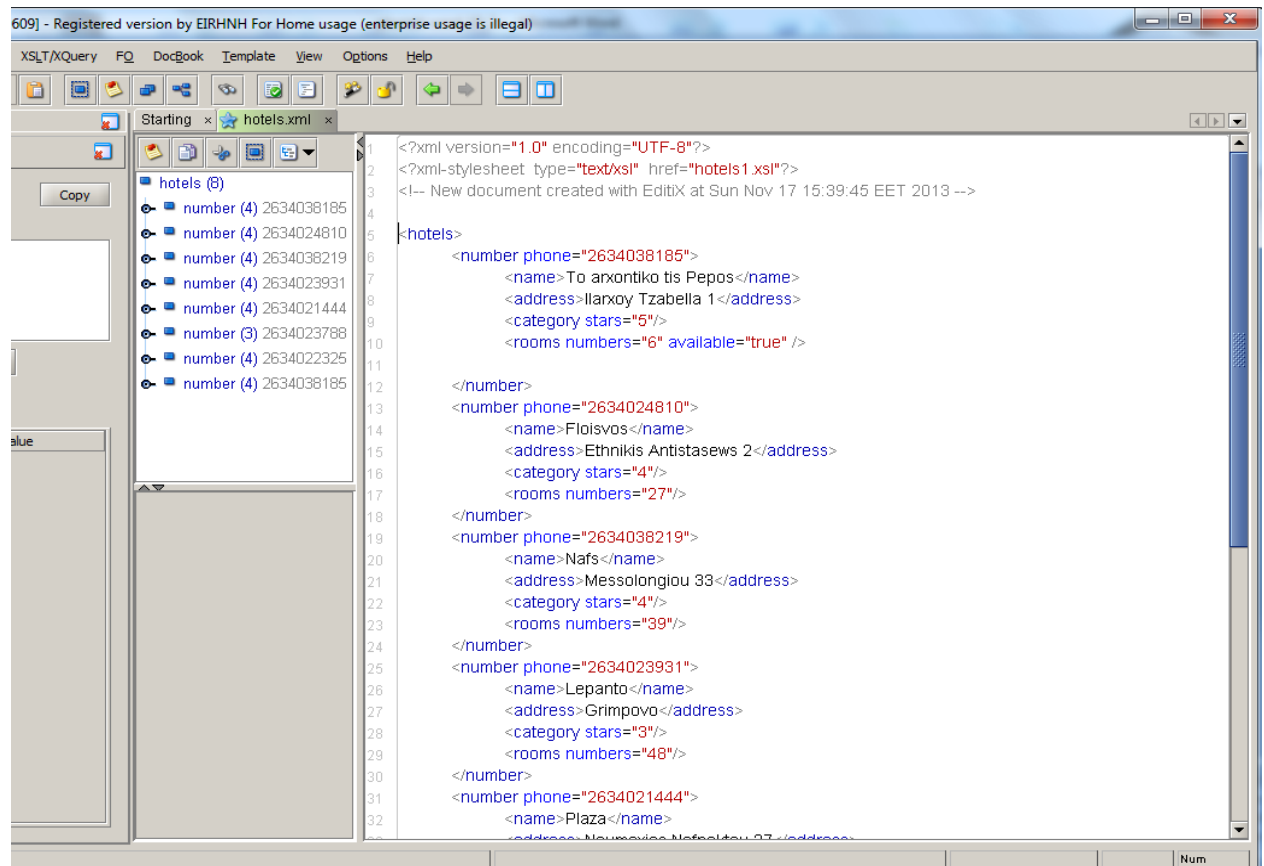
Πρώτα από όλα, δημιουργήσαμε ξεχωριστά αρχεία για τη κάθε κατηγορία, ούτως ώστε να είναι πιο ευανάγνωστα και πιο εύχρηστα για τον προγραμματιστή. Με τη βοήθεια του Editix δημιουργήσαμε το αρχείο hotels.xml όπου μέσα περιλαμβάνεται όλη η πληροφορία για

τα ξενοδοχεία της περιοχής. Αντίστοιχα το ίδιο κάναμε και για τις υπόλοιπες κατηγορίες, δηλαδή για τα ενοικιαζόμενα αυτοκίνητα έχουμε το αρχείο cars_motors.xml, για τα αξιοθέατα χρησιμοποιούμε το αρχείο attractions.xml, για τα δρομολόγια φτιάξαμε το αρχείο dromologia.xml, για τα μουσεία της Ναυπάκτου έχουμε το αρχείο museums.xml και τέλος, για τα ταξιδιωτικά γραφεία έχουμε το αρχείο offices..xml .

Δεδομένου ότι η Xml είναι μια Markup γλώσσα παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις. Συνεπώς όλα τα αρχεία που δημιουργήθηκαν στη βάση περιέχουν δομημένη όλη τη πληροφορία και είναι χωρισμένα σε tags. Ανοίγοντας το αρχείο hotels.xml βλέπουμε ότι υπάρχει ένα μεγάλο tag με το όνομα <hotels> όπου μέσα σ αυτό υπάρχουν μικρότερα κατηγορίες χωρισμένες και αυτές σε tags. Μέσα σ αυτά τα tags εμπεριέχεται όλη η πληροφορία για τα ξενοδοχεία, όπως το όνομα του κάθε ξενοδοχείου με το αντίστοιχο tag </name>, η διεύθυνση με tag </address>, η κατηγορία με tag </category> και τα δωμάτια με tag </rooms >. Με τον ίδιο ακριβώς τρόπο δημιουργήθηκαν και τα υπόλοιπα αρχεία.

Για το αρχείο offices.xml χρειάστηκε να χρησιμοποιήσουμε την εντολή enumeration η οποία περιορίζει το περιεχόμενο ενός στοιχείου XML σε ένα σύνολο αποδεκτών τιμών. Η εντολή αυτή μας συγκρίνει και μορφοποιεί τα ίδιου τύπου δεδομένα μας. Όπως και στα υπόλοιπα αρχεία, έτσι και σε αυτό η πληροφορία συμπεριλαμβάνεται σε δομημένα tags. Επομένως έχουμε ένα μεγάλο tag με όνομα <office/>, όπου περιλαμβάνει το όνομα του γραφείου με tag <name/> και μέσα έχουμε το τηλέφωνο με tag <tel/>. Έχουμε μέσα άλλο ένα μεγάλο tag εξίσου μεγάλο όπου μέσα έχουμε όλη την πληροφορία για τον προορισμό, τις ημέρες αναχώρησης, την ώρα και το αντίτιμο του εισιτηρίου.

Παρακάτω θα προσθέσουμε εικόνες για να δούμε πως δημιουργήθηκε η βάση μας, καθώς επίσης, και τα Xsl αρχεία.



Εικόνα 18 Δημιουργία κώδικα για τα hotels

Ο κώδικας για τα hotels.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet type="text/xsl" href="hotels1.xsl"?>
```

```
<!-- New document created with EditiX at Sun Nov 17 15:39:45 EET 2013 -->
```

```
<hotels> //1
```

```
  <number phone="2634038185"> //2
```

```
    <name>To arxontiko tis Pepos</name> //3
```

```
    <address>Ilarxoy Tzabella 1</address> //4
```

```
    <category stars="5"/> //5
```

```
    <rooms numbers="6" available="true" /> //6
```

```
  </number> //7
```

```
  <number phone="2634024810"> //8
```

```
    <name>Floisvos</name> //9
```

```
    <address>Ethnikis Antistasews 2</address> //10
```

```

        <category stars="4"/> //11
        <rooms numbers="27"/> //12
</number> //13
<number phone="2634038219"> //14
        <name>Nafs</name> //15
        <address>Messolongiou 33</address> //16
        <category stars="4"/> //17
        <rooms numbers="39"/> //18
</number> //19
<number phone="2634023931"> //20
        <name>Lepanto</name> //21
        <address>Grimpovo</address> //23
        <category stars="3"/> //24
        <rooms numbers="48"/> //25
</number> //26
<number phone="2634021444"> //27
        <name>Plaza</name> //28
        <address>Naumaxias Nafpaktou 37</address> //29
        <category stars="3"/> //30
        <rooms numbers="30"/> //31
</number> //32
<number phone="2634023788"> //33
        <name>Xrisi Akti</name> //34
        <address>Naumaxias Nafpaktou 29</address> //35
        <category stars="3"/> //36
</number> //37
<number phone="2634022325"> //38
        <name>Nafpaktos</name> //39
        <address>Korrydalews 4</address> //40
        <category stars="3"/> //41
        <rooms numbers="50"/> //42
</number> //43

```

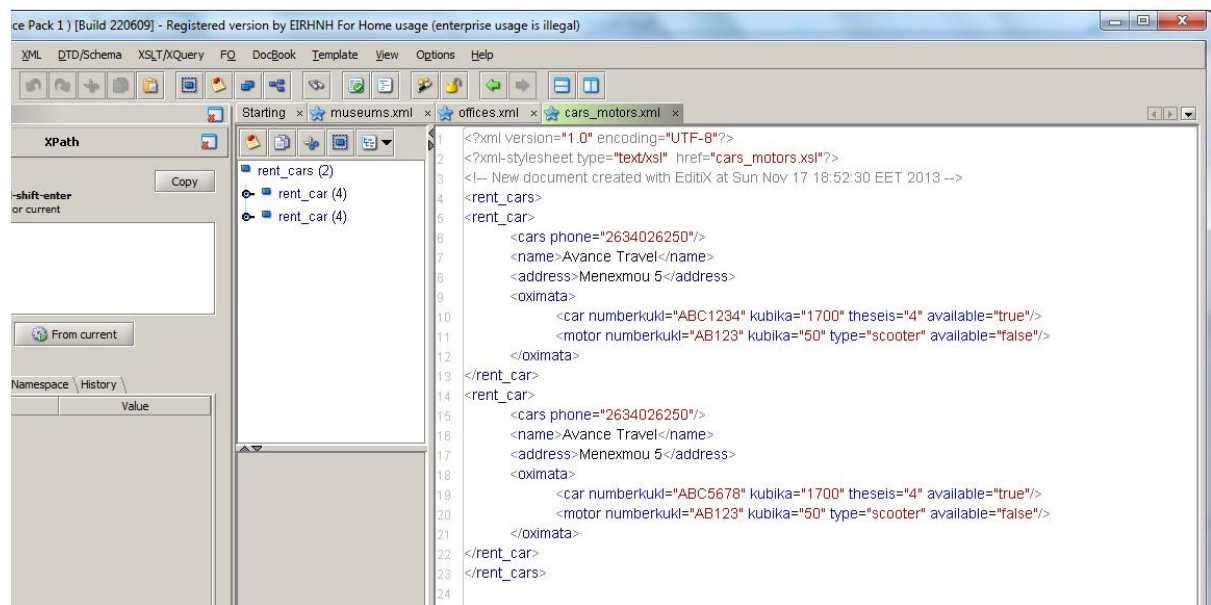
```

<number phone="2634038185"> //44
    <name>Akti</name> //45
    <address>Grimpono</address> //46
    <category stars="2"/> //47
    <rooms numbers="61"/> //48
</number> //49
</hotels> //50

```

Επεξήγηση του κώδικα Xml

Στη πρώτη γραμμή του κώδικα δημιουργούμε το μεγάλο tag με όνομα <hotels> όπου μέσα σ' αυτό εσωκλείονται τα elements και τα attributes για το κάθε ξενοδοχείο ξεχωριστά. Συνεχίζουμε με τη δεύτερη γραμμή του κώδικα όπου δημιουργούμε το αμέσως επόμενο μεγάλο tag με το όνομα <number> το οποίο μέσα περιλαμβάνει το attribute phone, καθώς αυτό παίρνει τον αριθμό του τηλεφώνου του ξενοδοχείου, το element name, address, category με το αντίστοιχο attribute stars που δηλώνει πόσα αστέρια είναι και τέλος, το element rooms με attribute το numbers που δηλώνει πόσα δωμάτια περιλαμβάνει το συγκεκριμένο ξενοδοχείο. Με τον ίδιο τρόπο δημιουργήθηκαν και τα υπόλοιπα 7 ξενοδοχεία.



Εικόνα 19 Ο Κώδικας για τα cars_motors

Ο κώδικας για τα cars_motors.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet type="text/xsl" href="cars_motors.xsl"?>
<!-- New document created with EditiX at Sun Nov 17 18:52:30 EET 2013 -->
<rent_cars> //1
<rent_car> //2
    <cars phone="2634026250"/> //3
    <name>Avance Travel</name> //4
    <address>Menexmou 5</address> //5
    <oximata> //6
        <car numberkukl="ABC1234" kubika="1700" theseis="4" available="true"/>
//7
        <motor numberkukl="AB123" kubika="50" type="scooter"
available="false"/> //8
    </oximata> //9
</rent_car> //10
<rent_car> //12
    <cars phone="2634026250"/> //13
    <name>Avance Travel</name> //14
    <address>Menexmou 5</address> //15
    <oximata> //16
        <car numberkukl="ABC5678" kubika="1700" theseis="4" available="true"/>
//17
        <motor numberkukl="AB123" kubika="50" type="scooter"
available="true"/> //18
    </oximata> //19
</rent_car> //20
</rent_cars> //21
```


Επεξήγηση του κώδικα Xml

Όπως και στα hotels.xml έτσι και στα cars_motors έχουμε δημιουργήσει ένα μεγάλο tag με το όνομα <rent_cars> όπου μέσα έχουμε φτιάξει ένα πιο μικρό tag με το όνομα <rent_car>, το οποίο περιλαμβάνει όλα τα elements και attributes που αφορούν τις παροχές του μαγαζιού. Από τη δεύτερη γραμμή του κώδικα και κάτω περιλαμβάνονται τα elements cars με το attribute phone, το element name που περιέχει το όνομα του μαγαζιού όπου ο χρήστης μπορεί να νοικιάσει οποιοδήποτε όχημα, αυτοκίνητο ή μηχανάκι και τη διεύθυνση που και αυτή είναι element. Στη συνέχεια, στην έκτη γραμμή του κώδικα έχουμε δημιουργήσει ένα πιο μικρό tag με το όνομα <oximata> όπου μέσα έχουμε συμπεριλάβει τα elements car και motor όπου για το car έχουμε τα εξής attributes: α) numberkucl το οποίο παίρνει σαν τιμή τη πινακίδα του οχήματος καθώς αυτό λειτουργεί και σαν μοναδικό κλειδί για τη βάση μας, β) kubika που παίρνει τον αριθμό των κυβικών, γ) theseis που παίρνει σαν τιμή τον τύπο του αυτοκινήτου, δηλαδή αν είναι τετραθέσιο ή πενταθέσιο κλπ, και τέλος έχουμε βάλει ένα attribute με όνομα available το οποίο παίρνει μια Boolean τιμή που δηλώνει αν η κράτηση έχει γίνει ή όχι. Για το motor έχουμε τα ίδια elements με τη μόνη διαφορά ότι εδώ έχουμε το element type που παίρνει ένα attribute το οποίο δηλώνει τον τύπο του δικύκλου, δηλαδή αν είναι scooter ή chooper ή suber bike αντί για το element theseis. Όσον αφορά το element available αν η τιμή του είναι "true" τότε ο χρήστης έχει τη δυνατότητα να κάνει τη κράτηση, μόλις πατήσει το κουμπί "Κράτηση" τότε αυτομάτως η μεταβλητή αλλάζει σε false και δεν μπορεί πλέον να κάνει άλλη κράτηση, καθώς με αυτό το τρόπο δηλώνεται ότι το όχημα δεν είναι διαθέσιμο .

Ενοικιαζόμενα Οχήματα Ναυπάκτου

Avance Travel

Phone: 2634026250

Address: Menexμου 5

Αυτοκίνητα:

Κράτηση ABC1234, 1700 κυβικά, 4 θέσεις

Μηχανές:

Κράτηση AB123, 50 κυβικά, τύπος scooter

Avance Travel

Phone: 2634026250

Address: Menexμου 5

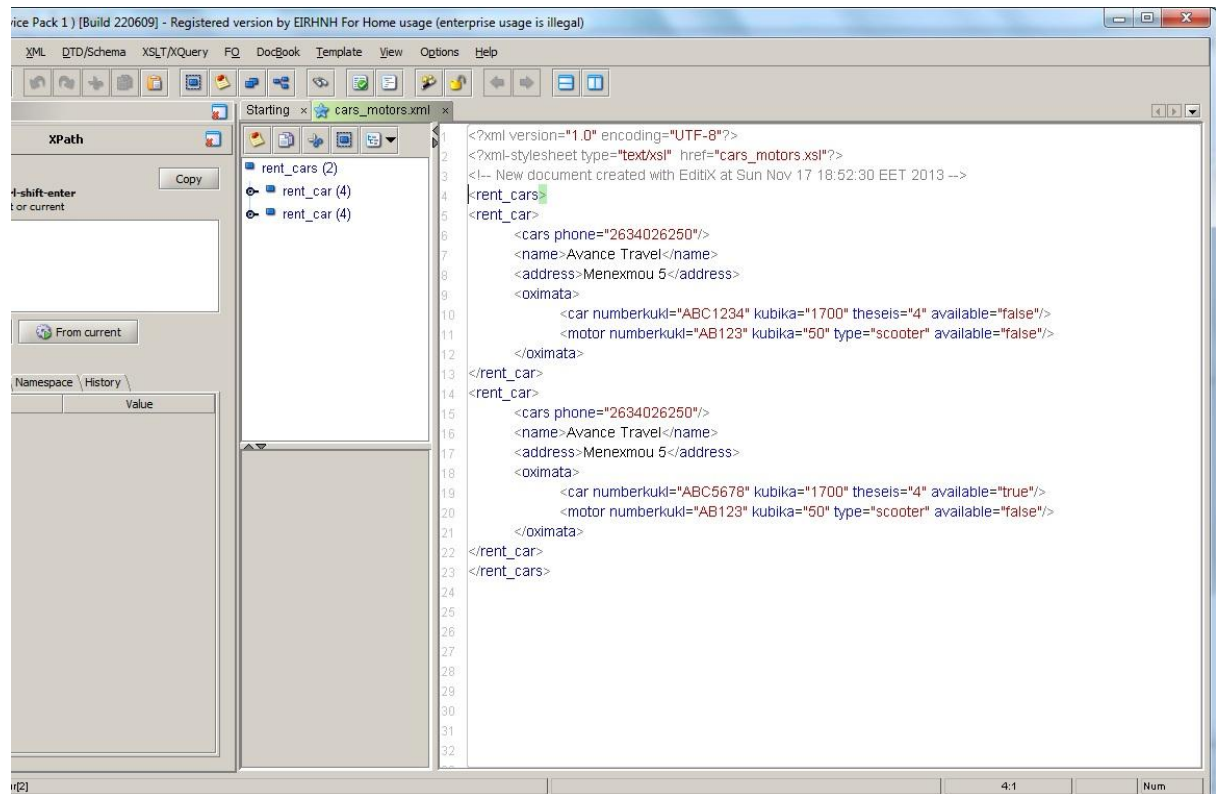
Αυτοκίνητα:

Κράτηση ABC5678, 1700 κυβικά, 4 θέσεις

Μηχανές:

Κράτηση AB123, 50 κυβικά, τύπος scooter

Εικόνα 20 Δυνατότητα κράτησης αν το element available ="true"



Εικόνα 21 Αλλαγή μεταβλητής

127.0.0.1/naupaktos/reserve_car.php

Ενοικιαζόμενα Οχήματα Ναυπάκτου

Avance Travel
Phone: 2634026250
Address: Μενεχμου 5

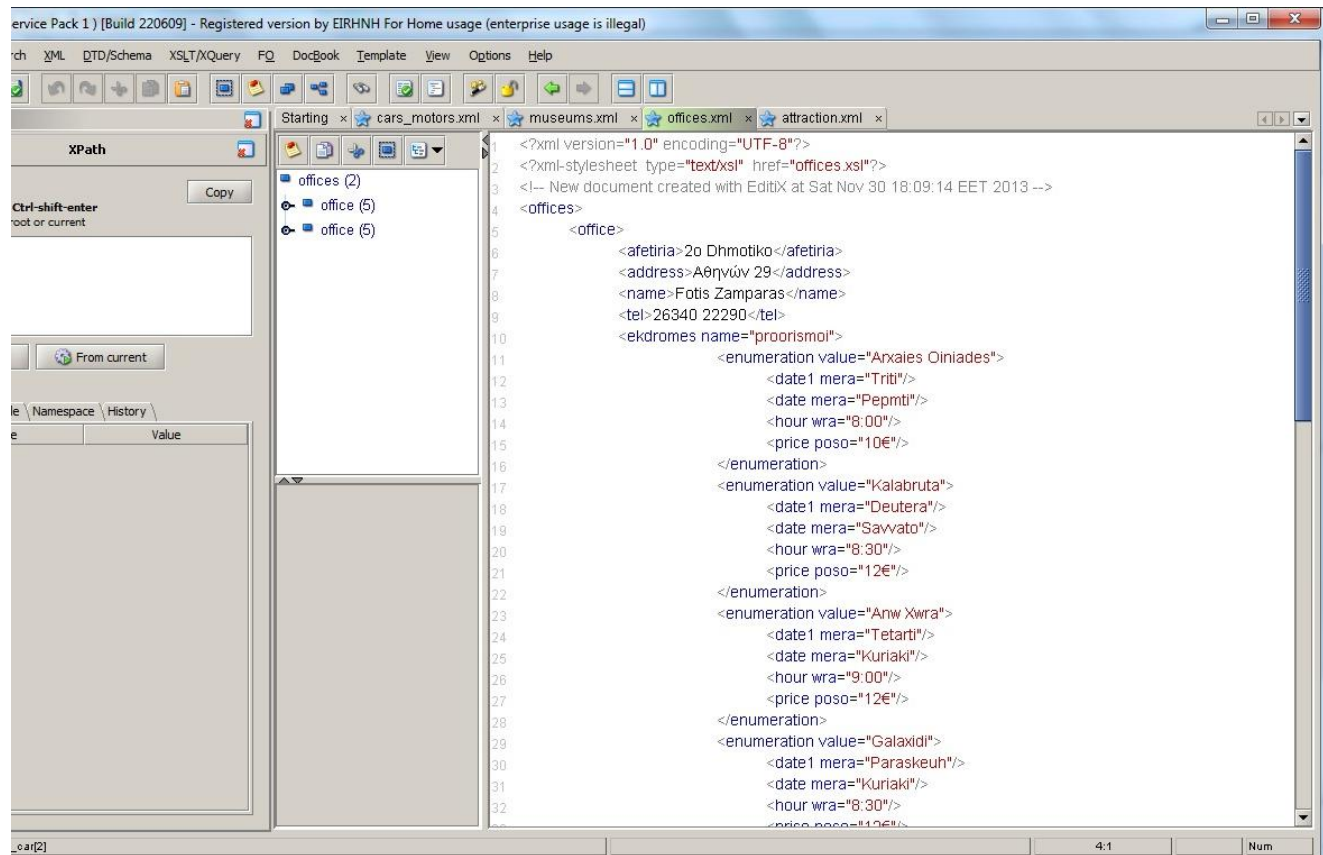
Αυτοκίνητα:
ABC1234, 1700 κυβικά, 4 θέσεις
Μηχανές:
AB123, 50 κυβικά, τύπος scooter

Avance Travel
Phone: 2634026250
Address: Μενεχμου 5

Αυτοκίνητα:
 ABC5678, 1700 κυβικά, 4 θέσεις
Μηχανές:
AB123, 50 κυβικά, τύπος scooter

2014 Vanessa Douni - Eirini Loukou

Εικόνα 22 Το κουμπί κράτηση δεν υπάρχει για τις τρεις επιλογές



Εικόνα 23 Ο κώδικας για τα offices

Ο κώδικας για τα office.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="offices.xsl"?>
<!-- New document created with EditiX at Sat Nov 30 18:09:14 EET 2013 -->
<offices>
  <office>
    <afetiria>2o Dhmotiko</afetiria>
    <address>Αθηνών 29</address>
    <name>Fotis Zamparas</name>
    <tel>26340 22290</tel>
    <ekdromes name="proorismoι">
      <enumeration value="Arxaies Oiniades">
        <date1 mera="Triti"/>
        <date mera="Pepmti"/>

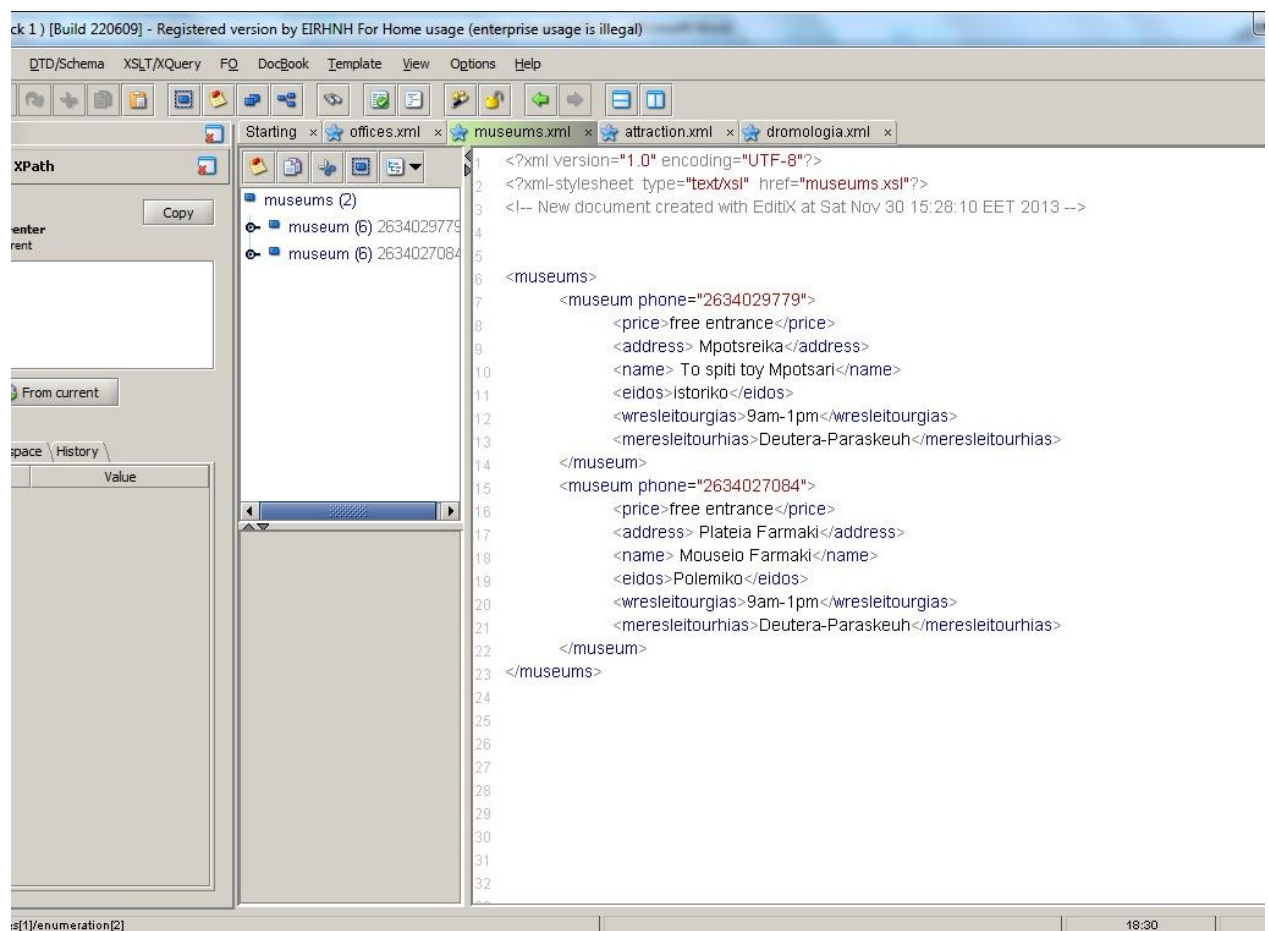
```

```
<hour wra="8:00"/>
<price poso="10€"/>
</enumeration>
<enumeration value="Kalabruta">
  <date1 mera="Deutera"/>
  <date mera="Savvato"/>
  <hour wra="8:30"/>
  <price poso="12€"/>
</enumeration>
<enumeration value="Anw Xwra">
  <date1 mera="Tetarti"/>
  <date mera="Kuriaki"/>
  <hour wra="9:00"/>
  <price poso="12€"/>
</enumeration>
<enumeration value="Galaxidi">
  <date1 mera="Paraskeuh"/>
  <date mera="Kuriaki"/>
  <hour wra="8:30"/>
  <price poso="12€"/>
</enumeration>
<enumeration value="Amfiloxia-Bonitsa">
  <date1 mera="Triti"/>
  <date mera="Kuriaki"/>
  <hour wra="8:00"/>
  <price poso="15€"/>
</enumeration>
</ekdromes>
</office>
```

```
<office>
  <afetiria>Papaxaralampeio Ghpedo</afetiria>
  <ekdromes name="proorismoι">
    <enumeration value="Leukada">
      <date1 mera="Tetarti"/>
      <date mera="Kuriaki"/>
      <hour wra="7:30"/>
      <price poso="20€"/>
    </enumeration>
    <enumeration value="Ampelakiwtissa-Elatou">
      <date1 mera="Deutera"/>
      <date mera="Tetarti"/>
      <hour wra="8:00"/>
      <price poso="10€"/>
    </enumeration>
    <enumeration value="Thermo-trixwnida">
      <date1 mera="Pempti"/>
      <date mera="Kuriaki"/>
      <hour wra="9:00"/>
      <price poso="12€"/>
    </enumeration>
    <enumeration value="Araxwba">
      <date1 mera="Paraskeuh"/>
      <date mera="Savvatos"/>
      <hour wra="9:30"/>
      <price poso="15€"/>
    </enumeration>
  </ekdromes>
  <address>Αθηνών 67</address>
  <name>Ntotsikas</name>
  <tel>26340 25338</tel>
</office>
</offices>
```

Επεξήγηση του κώδικα Xml

Και σε αυτό το αρχείο δημιουργήσαμε, όπως, ορίζει ο κανόνας της Xml ένα μεγάλο tag με το όνομα <offices> που μέσα βάλαμε άλλο ένα μεγάλο tag με όνομα <office> το οποίο περιλαμβάνει όλη τη πληροφορία για τα ταξιδιωτικά γραφεία της περιοχής. Πιο συγκεκριμένα, μέσα στο tag <office> υπάρχει το element <afetiria>, το address όπου μέσα σ' αυτό περιλαμβάνεται η διεύθυνση του συγκεκριμένου γραφείου, το <name>, το <thl>, καθώς επίσης και το <ekdromes> με το δικό του attribute name, το οποίο παίρνει σαν τιμή τους προορισμούς που διοργανώνει το κάθε γραφείο. Μέσα στο element <ekromes> συμπεριλάβαμε το element <enumeration> με το attribute του <value>, το οποίο το μεν, μας συγκρίνει και μορφοποιεί τα ίδιου τύπου δεδομένα μας και το δε, παίρνει το όνομα του γραφείου της περιοχής. Μέσα στο element <enumeration> περιλαμβάνονται τα elements <date 1> και <date> με τα attributes <mera> που παίρνουν σαν τιμή την ημέρα που διοργανώνεται η εκάστοτε εκδρομή. Επίσης, μέσα στο elements <enumeration> έχουμε συμπεριλάβει και τα επιπλέον elements, <hour> και <price> που μέσα σ' αυτά περιλαμβάνονται τα αντίστοιχα attributes <wra> και <poso>. Το πρώτο παίρνει σαν τιμή την ώρα αναχώρησης και το δεύτερο το ποσό που πρέπει να πληρώσει ο ενδιαφερόμενος πελάτης.



Εικόνα 24 Δημιουργία κώδικα για τα μουσεία

Ο κώδικας για τα museum.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="museums.xsl"?>
<!-- New document created with EditiX at Sat Nov 30 15:28:10 EET 2013 -->

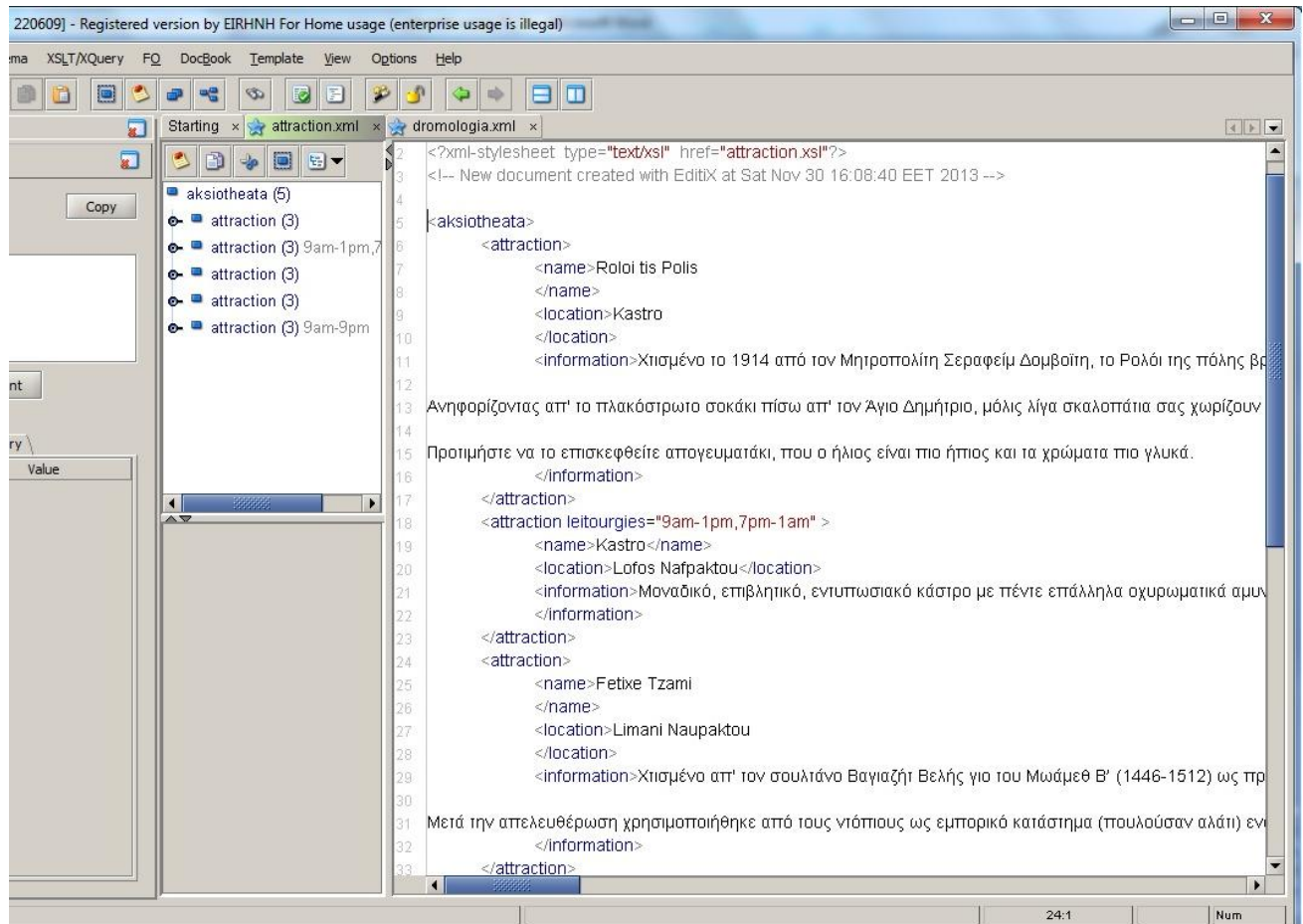
<museums> //1
  <museum phone="2634029779"> //2
    <price>free entrance</price> //3
    <address> Mpotsreika</address> //4
    <name> To spiti toy Mpotsari</name> //5
    <eidος>istoriko</eidος> //6
    <wresleitourgias>9am-1pm</wresleitourgias> //7
    <meresleitourhias>Deutera-Paraskeuh</meresleitourhias> //8
  </museum> //9
  <museum phone="2634027084"> //10
    <price>free entrance</price> //11
    <address> Plateia Farmaki</address> //12
    <name> Mouseio Farmaki</name> //13
    <eidος>Polemiko</eidος> //14
    <wresleitourgias>9am-1pm</wresleitourgias> //15
    <meresleitourhias>Deutera-Paraskeuh</meresleitourhias> //16
  </museum> //17
</museums> //18

```

Επεξήγηση του κώδικα Xml

Όπως σ’ όλα τα αρχεία έτσι και σε αυτό έχουμε δημιουργήσει ένα μεγάλο tag με όνομα <museums> όπου μέσα σ’ αυτό περιλαμβάνεται όλη η πληροφορία για τα μουσεία της περιοχής της Ναυπάκτου. Στη δεύτερη γραμμή του κώδικα έχουμε φτιάξει ένα πιο μικρό tag το <museum> και μέσα σ’ αυτό έχουμε προσθέσει όλη τη πληροφορία που χρειαζόμαστε για να είναι λειτουργική η βάση μας. Από τη τρίτη έως την όγδοη γραμμή του κώδικα έχουμε βάλει όλα τα elements που είναι απαραίτητα καθώς και το text που χρειαζόμαστε για να εμφανίζεται η πληροφορία στη σελίδα μας. Πιο συγκεκριμένα και βλέποντας τον παραπάνω κώδικα το element <price> περιλαμβάνει το text “free entrance”, το <address> περιλαμβάνει το text “Plateia Farmaki” “Mpotsreika”, το <name> το text “ To spiti toy Mpotsari ”, το <eidος> το text “istoriko”, το element <wresleitourgias> περιλαμβάνει το text “ 9am-1pm” και τέλος το <meresleitourhias> εσωκλείει το text “Deutera-Paraskeuh”. Στην ένατη γραμμή του

κώδικα κλείνει το tag <museum> και με ακριβώς την ίδια λογική δημιουργούμε ένα άλλο για το δεύτερο μουσείο της περιοχής.



Εικόνα 25 Δημιουργία κώδικα για τα αξιοθέατα

Ο κώδικας για τα attraction.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="attraction.xsl"?>
<!-- New document created with EditiX at Sat Nov 30 16:08:40 EET 2013 -->
<aksiotheata> //1
  <attraction> //2
    <name>Roloi tis Polis //3
  </name> //4
  <location>Kastro //5
  </location> //6
  <information>Χτισμένο το 1914 από τον Μητροπολίτη Σεραφεϊμ Δαμβοϊτή, το Ρολόι της πόλης βρ
  Ανηφορίζοντας απ' το πλακόστρωτο σοκάκι πίσω απ' τον Άγιο Δημήτριο, μόλις λίγα σκαλοπάτια σας χωρίζουν
  Προιμύση να το επισκεφθείτε απογευματικά, που ο ήλιος είναι πιο ήπιος και τα χρώματα πιο γλυκά.
  </information>
  </attraction>
  <attraction leitourgies="9am-1pm,7pm-1am" >
    <name>Kastro</name>
    <location>Lofos Nafpaktou</location>
    <information>Μοναδικό, επιβλητικό, εντυπωσιακό κάστρο με πέντε επάλληλα οχυρωματικά αμυν
    </information>
  </attraction>
  <attraction>
    <name>Fetixe Tzami
    </name>
    <location>Limani Naupaktou
    </location>
    <information>Χτισμένο απ' τον σουλτάνο Βαγιαζήτ Βελής γιο του Μωάμεθ Β' (1446-1512) ως πρ
  </information>
  </attraction>
  </aksiotheata>
  
```

<information>Χτισμένο το 1914 από τον Μητροπολίτη Σεραφείμ Δομβοΐτη, το Ρολόι της πόλης βρίσκεται σε προνομιακή θέση.Ανηφορίζοντας απ' το πλακόστρωτο σοκάκι πίσω απ' τον Άγιο Δημήτριο, μόλις λίγα σκαλοπάτια σας χωρίζουν απ' την πανοραμική θέα στον κόλπο της Ναυπάκτου. Αν ξεχαστείτε με τη θέα, ο ήχος του ρολογιού- που οι ντόπιοι αποκαλούν και Σεραφείμ προς τιμή του εμπνευστή του- θα σας θυμίσει την ώρα.Προτιμήστε να το επισκεφθείτε απογευματάκι, που ο ήλιος είναι πιο ήπιος και τα χρώματα πιο γλυκά. //7

</information> //8

</attraction> //9

<attraction leitourgies="9am-1pm,7pm-1am" > //10

<name>Kastro</name>

<location>Lofos Nafpaktou</location>

<information>Μοναδικό, επιβλητικό, εντυπωσιακό κάστρο με πέντε επάλληλα οχυρωματικά αμυντικά διαζώματα, από την κορυφή του λόφου (Ιτζ-Καλέ) μέχρι το λιμάνι. Το κάστρο της Ναυπάκτου ποτέ δεν κατακτήθηκε, η παράδοση του ήταν αποτέλεσμα πάντα διαπραγματεύσεων και συνθηκών. Από την αρχαιότητα η Ναύπακτος υπήρξε οχυρωμένη πόλη. Στα χρόνια του κράτους της Ηπείρου χτίστηκε το μεσαιωνικό κάστρο στην πρώτη του μορφή. Στην αλληλοδιαδοχή των κυριάρχων που ακολούθησε - Φράγκοι, Αλβανοί, Ιππότες της Ρόδου και Ενετοί - οι τελευταίοι το 15ο αι. το επισκεύασαν και πήρε τότε τη σημερινή του μορφή.

</information>

</attraction>

<attraction>

<name>Fetixe Tzami

</name>

<location>Limani Naupaktou

</location>

<information>Χτισμένο απ' τον σουλτάνο Βαγιαζήτ Βελής γιο του Μωάμεθ Β' (1446-1512) ως προσφορά στον Αλλάχ για την κατάληψη της Ναυπάκτου το 1499, το Φετιχέ Τζαμί ήταν το πρώτο μουσουλμανικό τέμενος της πόλης. Μετά την απελευθέρωση χρησιμοποιήθηκε από τους ντόπιους ως εμπορικό κατάστημα (πουλούσαν αλάτι) ενώ, σήμερα, ανακαινισμένο από της 22η Εφορεία Αρχαιοτήτων που εδρεύει στην πόλη, είναι επισκέψιμο και χρησιμοποιείται για την διεξαγωγή πολιτιστικών εκδηλώσεων.

</information>

</attraction>

<attraction>

<name>Parko Thervantes

</name>

<location>Limani Naupaktou

</location>

<information>Τι γυρεύει ο μεγάλος Ισπανός λογοτέχνης, ποιητής και συγγραφέας Μιγκέλ ντε Θερβάντες, στη Ναύπακτο; Το Σεπτέμβριο του 1571 υπηρέτησε ως υπαξιωματικός του μεγάλου στόλου που πολέμησε νικηφόρα στη Ναυμαχία της Ναυπάκτου στις 7 Οκτωβρίου, εναντίον του οθωμανικού στόλου, αμφισβητώντας για πρώτη φορά την κυριαρχία του στη Μεσόγειο. Η στάση του Θερβάντες υπήρξε γενναία, αρνούμενος να περιοριστεί στα «μετόπισθεν» παρά το γεγονός πως ήταν προσβεβλημένος από πυρετό. Κατά τη διάρκεια της μάχης τραυματίστηκε δύο φορές στο στήρνο, ενώ ένας τρίτος τραυματισμός του προκάλεσε μόνιμη βλάβη, αχρηστεύοντας το αριστερό του χέρι. Ευτυχώς για την λογοτεχνία ήταν δεξιόχειρας, χαρίζοντας μας τον Δον Κιχώτη.

</information>

</attraction>

<attraction leitourgies="9am-9pm">

<name>Gefyra Riou-Antirriou

</name>

<location>Dimos Naupaktou

</location>

<information>Η Γέφυρα Ρίου-Αντιρρίου (Γέφυρα Χαρίλαος Τρικούπης) είναι καλωδιωτή γέφυρα που ολοκληρώθηκε το 2004 μεταξύ του Ρίου και του Αντιρρίου, που συνδέει την Πελοπόννησο με τη δυτική ηπειρωτική Ελλάδα. Κατασκευάστηκε από την Γαλλική εταιρία Vinci. Το μήκος της γέφυρας που στηρίζεται σε τέσσερις πυλώνες, ανέρχεται στα 2.280 μέτρα, ενώ μαζί με τις προσβάσεις φτάνει γύρω στα 2.880 μέτρα. Η γέφυρα αναπτύχθηκε και ολοκληρώθηκε, παρά τις δυσμενείς περιβαλλοντικές συνθήκες, ενώ το μέγιστο βάθος θεμελίωσης φθάνει τα 65 μέτρα υπό την επιφάνεια της θάλασσας. Πρόκειται για μία καλωδιωτή γέφυρα, οι αντοχές της οποίας είναι εντυπωσιακές. Σχεδιάστηκε και κατασκευάστηκε με προδιαγραφές να αντέξει σε σεισμό μεγαλύτερο από 7,4 της Κλίμακας Ρίχτερ.

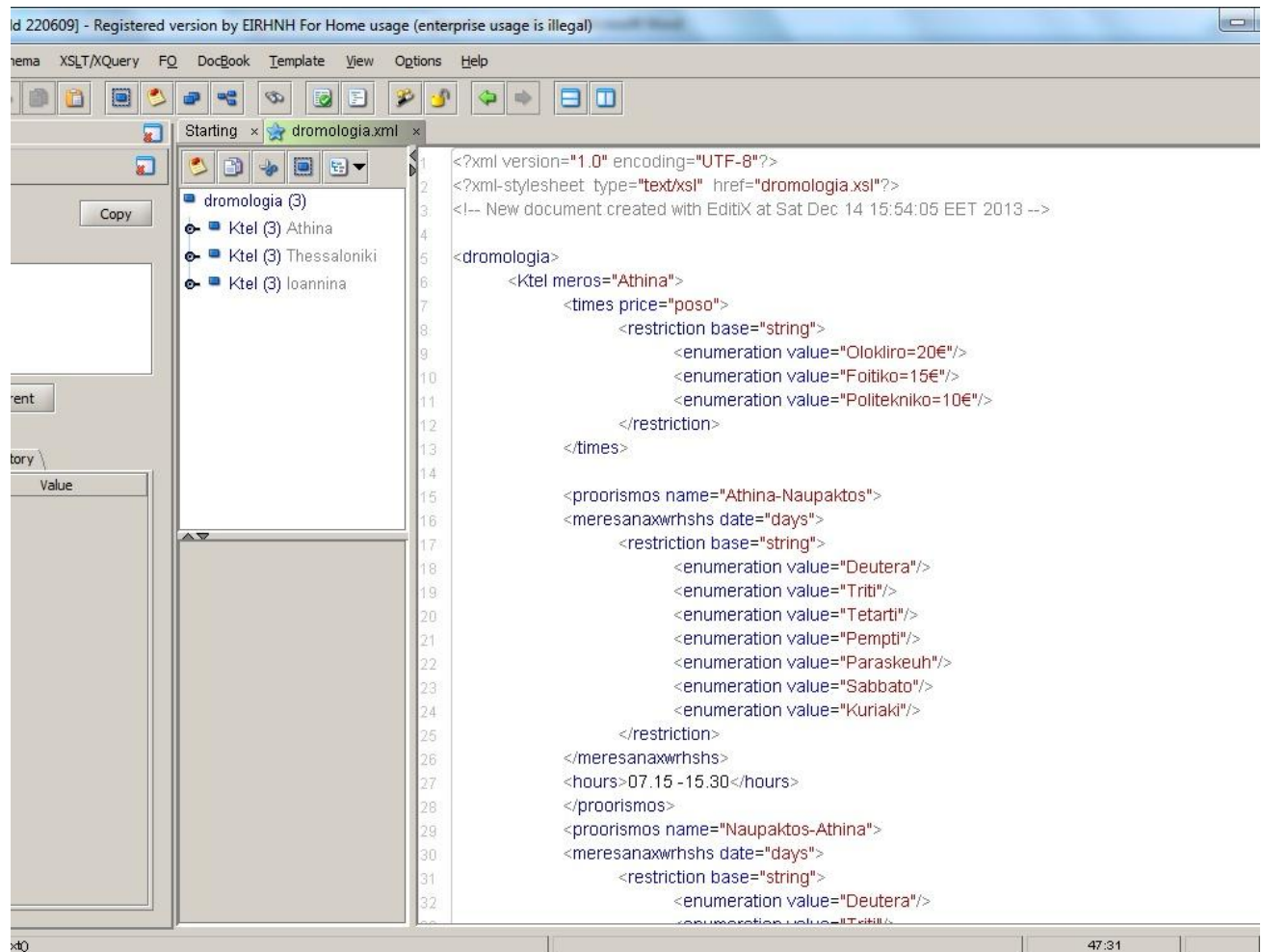
</information>

</attraction>

</aksiotheata>

Επεξήγηση του κώδικα Xml

Στη πρώτη γραμμή του κώδικα έχουμε δημιουργήσει το μεγάλο tag <aksiotheata> και μέσα σ' αυτό έχουμε φτιάξει άλλα έξι elements με το όνομα <attraction> όπου μέσα εκεί περικλείεται όλη η πληροφορία για τα αξιοθέατα της Ναυπάκτου. Από τη τρίτη γραμμή του κώδικα μέχρι την έβδομη έχουμε δημιουργήσει τα elements που χρειαζόμαστε για να περιγράψουμε το κάθε αξιοθέατο ξεχωριστά. Στο element <information> έχουμε βάλει σε text το κείμενο που θέλουμε να εμφανίζεται στην ιστοσελίδα μας. Κατόπιν, κλείνουμε τα elements από αυτό που έχει ανοίξει τελευταίο. Συνεχίζουμε με την ίδια λογική και για τα υπόλοιπα αξιοθέατα.



Εικόνα 26 Δημιουργία κώδικα για τα δρομολόγια

Ο κώδικας για τα dromologia.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="dromologia.xsl"?>
<!-- New document created with EditiX at Sat Dec 14 15:54:05 EET 2013 -->

<dromologia>
  <Ktel meros="Athina">
    <times price="poso">
      <restriction base="string">
        <enumeration value="Olokliro=20€"/>
        <enumeration value="Foitiko=15€"/>
        <enumeration value="Politekniko=10€"/>
      </restriction>
    </times>
  </Ktel>
  <Ktel meros="Naupaktos-Athina">
    <times price="poso">
      <restriction base="string">
        <enumeration value="Deutera"/>
        <enumeration value="Triti"/>
        <enumeration value="Tetarti"/>
        <enumeration value="Pempti"/>
        <enumeration value="Paraskeuh"/>
        <enumeration value="Sabbato"/>
        <enumeration value="Kuriaki"/>
      </restriction>
    </times>
  </Ktel>
</dromologia>

```

```

</times> //9

<proorismos name="Athina-Naupaktos"> //10
<meresanaxwrhshs date="days"> //11
  <restriction base="string"> //12
    <enumeration value="Deutera"/> //13
    <enumeration value="Triti"/> //14
    <enumeration value="Tetarti"/> //15
    <enumeration value="Pempti"/> //16
    <enumeration value="Paraskeuh"/> //17
    <enumeration value="Sabbato"/> //18
    <enumeration value="Kuriaki"/> //19
  </restriction> //20
</meresanaxwrhshs> //21
<hours>07.15 -15.30</hours> //22
</proorismos> //23
<proorismos name="Naupaktos-Athina">
<meresanaxwrhshs date="days">
  <restriction base="string">
    <enumeration value="Deutera"/>
    <enumeration value="Triti"/>
    <enumeration value="Tetarti"/>
    <enumeration value="Pempti"/>
    <enumeration value="Paraskeuh"/>
    <enumeration value="Sabbato"/>
    <enumeration value="Kuriaki"/>
  </restriction>
</meresanaxwrhshs>
<hours>10:00-17:00</hours>
</proorismos>
</Ktel>

```

```
<Ktel meros="Thessaloniki">
<times price="poso">
  <restriction base="string">
    <enumeration value="Olokliro=45€"/>
    <enumeration value="Foitiko=35€"/>
    <enumeration value="Politekniko=23€"/>
  </restriction>
</times>

<proorismos name="Thessaloniki-Naupaktos">
<meresanaxwrhshs date="days">
  <restriction base="string">
    <enumeration value="Deutera"/>
    <enumeration value="Triti"/>
    <enumeration value="Tetarti"/>
    <enumeration value="Pempti"/>
    <enumeration value="Paraskeuh"/>
    <enumeration value="Sabbato"/>
    <enumeration value="Kuriaki"/>
  </restriction>
</meresanaxwrhshs>
<hours>09:30 -15.30</hours>
</proorismos>
<proorismos name="Naupaktos-Thessaloniki">
<meresanaxwrhshs date="days">
  <restriction base="string">
    <enumeration value="Deutera"/>
    <enumeration value="Triti"/>
    <enumeration value="Tetarti"/>
    <enumeration value="Pempti"/>
    <enumeration value="Paraskeuh"/>
```

```
<enumeration value="Sabbato"/>
<enumeration value="Kuriaki"/>
</restriction>
</meresanaxwrhshs>
<hours>08:00-13:00</hours>
</proorismos>
</Ktel>
<Ktel meros="Ioannina">
  <times price="poso">
    <restriction base="string">
      <enumeration value="Olokliro=30€"/>
      <enumeration value="Foitiko=22€"/>
      <enumeration value="Politekniko=15€"/>
    </restriction>
  </times>

  <proorismos name="Ioannina-Naupaktos">
    <meresanaxwrhshs date="days">
      <restriction base="string">
        <enumeration value="Deutera"/>
        <enumeration value="Triti"/>
        <enumeration value="Tetarti"/>
        <enumeration value="Pempti"/>
        <enumeration value="Paraskeuh"/>
        <enumeration value="Kuriaki"/>
      </restriction>
    </meresanaxwrhshs>
    <hours>12:00</hours>
  </proorismos>
  <proorismos name="Naupaktos-Ioannina">
    <meresanaxwrhshs date="days">
      <restriction base="string">
```

```
<enumeration value="Deutera"/>
<enumeration value="Triti"/>
<enumeration value="Tetarti"/>
<enumeration value="Pempti"/>
<enumeration value="Paraskeuh"/>
<enumeration value="Kuriaki"/>

</restriction>

</meresanaxwrhshs>

<hours>10:00
</hours>

</proorismos>

</Ktel>

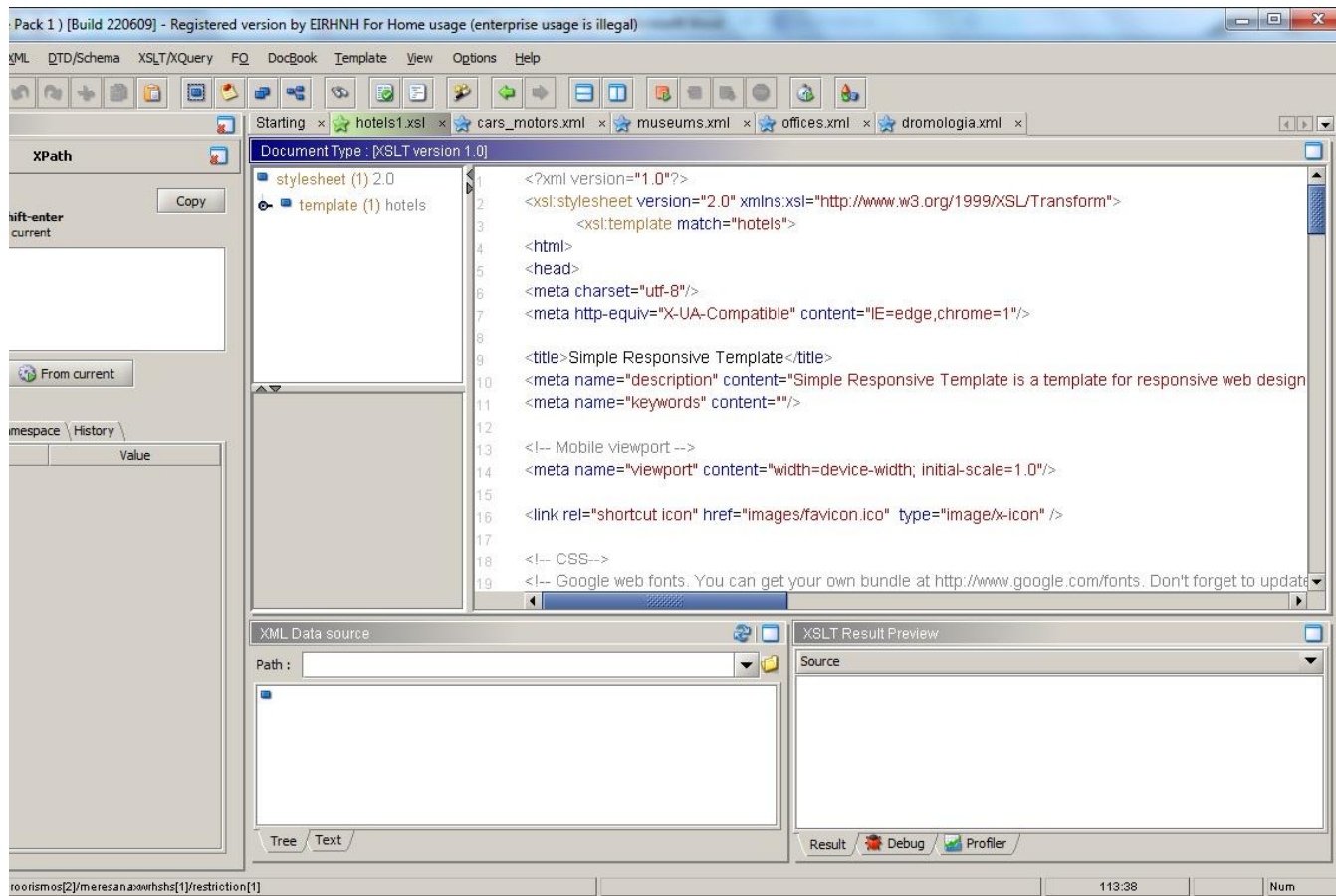
</dromologia>
```

Επεξήγηση του κώδικα Xml

Στη πρώτη γραμμή του κώδικα έχουμε φτιάξει το μεγάλο tag <dromologia> και στη δεύτερη γραμμή έχουμε βάλει το elements <Ktel> και μέσα σ' αυτό το attribute <meros> για να δηλώσουμε το μέρος από όπου ο χρήστης έχει τη δυνατότητα να ταξιδέψει προς και από την Ναύπακτο. Στη τρίτη γραμμή του κώδικα έχουμε βάλει το element <times> με το attribute <price> για να δηλώσουμε τις τιμές των εισιτηρίων. Πιο κάτω στη τέταρτη γραμμή έχουμε βάλει ένα element το οποίο δηλώνει και το περιορισμό που θέλουμε να υπάρχει για τις υπόλοιπες κατηγορίες όπως το <enumeration> το οποίο θέλουμε να παίρνει string, ούτως ώστε να εμφανίζεται στην σελίδα μας ακριβώς το κείμενο που υπάρχει στα αυτάκια(“”). Στη συνέχεια του κώδικά μας και συγκεκριμένα στην δέκατη γραμμή έχουμε βάλει το element <proorismos> με το attribute <name> για να δηλώσουμε από πού μπορεί να ταξιδέψει ο χρήστης και πού μπορεί να πάει. Στην ενδέκατη γραμμή έχουμε δηλώσει με το κατάλληλο element και attribute τις μέρες αναχώρησης και παρακάτω ομαδοποιούμε τις μέρες αναχώρησης. Τέλος, με το element <hours> δηλώνουμε τις ώρες.

4.5 Δημιουργία XSL αρχεία

Στη συνέχεια, αφού ολοκληρώσαμε όλα τα .xml αρχεία μας δημιουργήσαμε για το καθένα ξεχωριστά τα αντίστοιχα .xsl. Τα XSL αρχεία αποτελούν βασική προϋπόθεση για να ενσωματωθεί ο κώδικας XML στον HTML, καθώς και στην εμφάνιση των δεδομένων στη σελίδα μας. Πιο συγκεκριμένα η εντολή που τα κάνει όλα αυτά είναι η **<xsl:template match=" " >** όπου μέσα στα " " μπαίνει το αντίστοιχο αρχείο .xml. Για παράδειγμα, η εντολή **<xsl:template match="hotels">** συνδέει το αρχείο hotels.xml με το αντίστοιχο hotels.xsl ώστε να εμφανίζονται τα δεδομένα του .xml αρχείου στο site.



Κώδικα για τα hotels.xsl

```
<?xml version="1.0"?>
```

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
  <xsl:template match="hotels">
```

```
<html>
```

```
<head>

<meta charset="utf-8"/>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>

<title>Simple Responsive Template</title>

<meta name="description" content="Simple Responsive Template is a template for
responsive web design. Mobile first, responsive grid layout, toggle menu, navigation bar with
unlimited drop downs, responsive slideshow"/>

<meta name="keywords" content=""/>

<!-- Mobile viewport -->

<meta name="viewport" content="width=device-width; initial-scale=1.0"/>

<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />

<!-- CSS-->

<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't
forget to update the CSS accordingly!-->

<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'/>

<link rel="stylesheet" href="css/normalize.css"/>

<link rel="stylesheet" href="js/flexslider/flexslider.css" />

<link rel="stylesheet" href="css/basic-style.css"/>

<!-- end CSS-->

<!-- JS-->

<script src="js/libs/modernizr-2.6.2.min.js"></script>

<!-- end JS-->

</head>

<body id="home">

<!--[if lt IE 7]>

    <p class="chromeframe">You are using an <strong>outdated</strong> browser.
Please <a href="http://browsehappy.com/">upgrade your browser</a> or <a
href="http://www.google.com/chrome/?redirect=true">activate Google Chrome
Frame</a> to improve your experience.</p>

    <![endif]-->
```

```
<!-- header area -->
<header class="wrapper clearfix">

  <div id="banner">
    <!--<div id="logo"><a href="index.html"></a></div> -->
    </div>

  <!-- main navigation -->
  <nav id="topnav" role="navigation">
    <div class="menu-toggle">Menu</div>
    <ul class="srt-menu" id="menu-main-navigation">
      <li><a href="index.html">Αρχική</a></li>
      <li>
        <a href="#">Αξιοθέατα</a>
        <ul>
          <li><a href="attraction.xml">Αξιοθέατα</a></li>
          <li><a href="museums.xml">Μουσεία</a></li>
        </ul>
      </li>
      <li class="current"><a href="hotels.xml">Διαμονή</a></li>
      <li><a href="#">Υπηρεσίες</a>
        <ul>
          <li>
            <a href="offices.xml">Ταξιδιωτικά Γραφεία</a>
          </li>
          <li>
            <a href="cars_motors.xml">Ενοικιαζόμενα
Οχήματα</a>
          </li>
          <li>
            <a href="dromologia.xml">Δρομολόγια</a>
          </li>
        </ul>
      </li>
    </ul>
  </nav>
</div>
```

```
</li>

</ul>

</li>

</ul>

</nav>

</header><!-- end header -->

<section id="page-header" class="clearfix">
<!-- responsive FlexSlider image slideshow -->
<div class="wrapper">
    <h1>Ξενοδοχεία Ναυπάκτου</h1>
</div>

</section>

<!-- main content area -->
<div class="wrapper" id="main">

<!-- content area -->
    <section id="content">
        <xsl:for-each select="number/.">
            <h3><xsl:value-of select="name"/></h3>
            <p>
                <i><xsl:text>Category: </xsl:text></i><xsl:value-of
select="category/@stars"/><br/>
                <i><xsl:text>Address: </xsl:text></i><xsl:value-of
select="address"/><br/>
                <i><xsl:text>Rooms: </xsl:text></i><xsl:value-of
select="rooms/@numbers"/><br/>
            </p>
        </xsl:for-each>
    </section>
</div>
</div>
</div>
```

```

                                <i><xsl:text>Phone: </xsl:text></i><xsl:value-of
select="@phone"/>
                                </p>
                                </xsl:for-each>
</section><!-- #end content area -->
```

```
<!-- left sidebar -->
```

```
<aside>
```

```
    <!--
```

```
    <h2>Secondary Section menu</h2>
```

```
    <nav id="secondary-navigation">
```

```
        <ul>
```

```
            <li><a href="#">menu item</a></li>
```

```
            <li class="current"><a href="#">current menu item</a></li>
```

```
            <li><a href="#">menu item</a></li>
```

```
            <li><a href="#">menu item</a></li>
```

```
            <li><a href="#">menu item</a></li>
```

```
        </ul>
```

```
    </nav>
```

```
    -->
```

```
</aside><!-- #end left sidebar -->
```

```
</div><!-- #end div #main .wrapper -->
```

```
<!-- footer area -->
```

```
<footer>
```

```
    <div id="colophon" class="wrapper clearfix">
```

```
    <p>2014 Vassiliki Douni - Eirini Loukou</p>
```

```
</div>
```

```
<!--You can NOT remove this attribution statement from any page, unless you get the
permission from prowebdesign.ro-->
```

```
<!--
```

```
<div id="attribution" class="wrapper clearfix" style="color:#666; font-
size:11px;">Site built with <a href="http://www.prowebdesign.ro/simple-responsive-
template/" target="_blank" title="Simple Responsive Template is a free software by
www.prowebdesign.ro" style="color:#777;">Simple Responsive Template</a></div>
```

```
-->
```

```
<!--end attribution-->
```

```
</footer><!-- #end footer area -->
```

```
<!-- jQuery -->
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

Ο παραπάνω κώδικας είναι ένας Html κώδικας στον οποίο έχουμε προσαρμόσει με το κατάλληλο κώδικα xml έτσι ώστε η σελίδα μας να είναι πιο δυναμική και να εμφανίζονται τα δεδομένα που υπάρχουν στη βάση μας. Πιο συγκεκριμένα αυτή τη δυνατότητα μας τη δίνει ο παρακάτω κώδικας: <section id="content"> // 1

```
<xsl:for-each select="number/"> //2
```

```
<h3><xsl:value-of select="name"/></h3> //3
```

```
<p> //4
```

```
<i><xsl:text>Category: </xsl:text></i><xsl:value-of
select="category/@stars"/><br/> //5
```

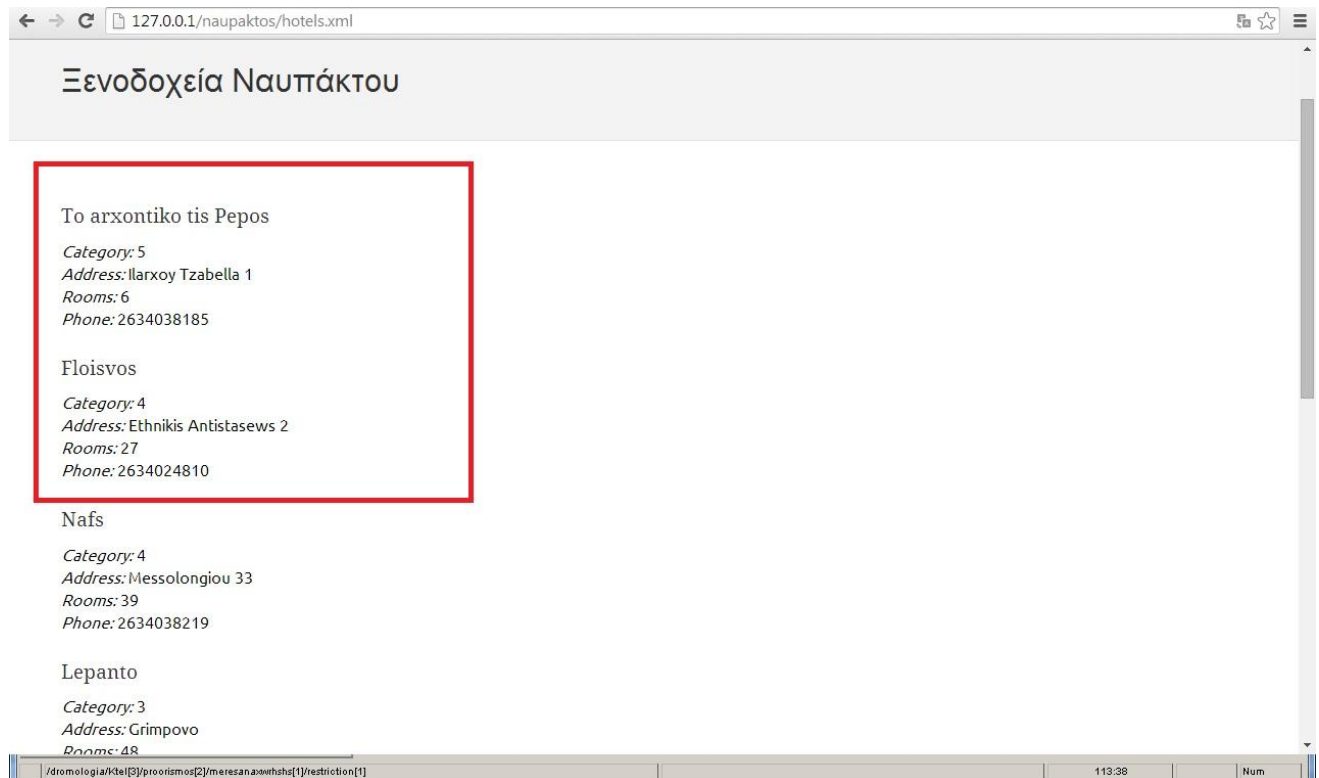
```
<i><xsl:text>Address: </xsl:text></i><xsl:value-of
select="address"/><br/> //6
```

```

        <i><xsl:text>Rooms: </xsl:text></i><xsl:value-of
select="rooms/@numbers"/><br/> //7
        <i><xsl:text>Phone: </xsl:text></i><xsl:value-of
select="@phone"/> //8
    </p>
</xsl:for-each>
</section>

```

Στη δεύτερη γραμμή του κώδικα η έννοια του for-each μπορεί να χρησιμοποιηθεί για την επιλογή κάθε στοιχείου Xml από ένα συγκεκριμένο κόμβο. Στη συγκεκριμένη περίπτωση δηλώνει τον κόμβο από το οποίο θα πάρει τα δεδομένα προς εμφάνιση των ξενοδοχείων. Στη τρίτη γραμμή του κώδικα εξάγουμε τη τιμή που θέλουμε για παράδειγμα με το name θέλουμε να πάρουμε το κείμενο που θα εμφανίζεται στην σελίδα μας. Στις επόμενες γραμμές εξάγουμε τα ονόματα έτσι όπως θέλουμε να εμφανίζεται στο site μας, δηλαδή, το Category, το Address, τα rooms και το Phone. Στη παρακάτω εικόνα φαίνεται το αποτέλεσμα του παραπάνω κώδικα.



Εικόνα 27 Αποτελέσματα XSL

Με τον ίδιο τρόπο δουλεύτηκαν και τα υπόλοιπα αρχεία χρησιμοποιώντας τον ίδιο κώδικα αλλάζοντας κάθε φορά το όνομα του κόμβου που γίνεται το for-each select ="". Έτσι, λοιπόν, για τα cars_motors έχουμε: xsl:for-each select="rent_car">

```
<h3><xsl:value-of select="name"/></h3>

<p>

        <i><xsl:text>Phone: </xsl:text></i><xsl:value-of
select="cars/@phone"/><br/>

        <i><xsl:text>Address: </xsl:text></i><xsl:value-of
select="address"/><br/>

        <form method="post" action="reserve_car.php">

        <i><xsl:text>Αυτοκίνητα: </xsl:text></i><br/>

        <xsl:for-each select="./oximata/car">

                <xsl:if test="@available = 'true'"><input
type="submit" onclick="" value="Κράτηση"
name="{ @numberkukl}"/>&#160;&#160;</xsl:if>

                <xsl:value-of select="@numberkukl"/>, <xsl:value-
of select="@kubika"/> κυβικά, <xsl:value-of select="@theseis"/> θέσεις<br/>

                </xsl:for-each>

        <i><xsl:text>Μηχανές: </xsl:text></i><br/>

        <xsl:for-each select="./oximata/motor">

                <xsl:if test="@available = 'true'"><input
type="submit" value="Κράτηση" name="{ @numberkukl}"/>&#160;&#160;</xsl:if>

                <xsl:value-of select="@numberkukl"/>,
<xsl:value-of select="@kubika"/> κυβικά, τύπος <xsl:value-of select="@type"/><br/>

                </xsl:for-each>
```

Σ' αυτό το κώδικα έχουμε φτιάξει με την εντολή type="submit" onclick="" value="Κράτηση" το κουμπί «Κράτηση» που εμφανίζεται στην ιστοσελίδα μας για να μπορεί ο χρήστης να κάνει τη κράτηση για το όχημα που θέλει.

Για τα offices έχουμε: <xsl:for-each select="office">

```
<h3><xsl:value-of select="name"/></h3>

<p>

        <i><xsl:text>Phone: </xsl:text></i><xsl:value-of
select="tel"/><br/>

        <i><xsl:text>Address: </xsl:text></i><xsl:value-of
select="address"/><br/>

        <i><xsl:text>Starting point:
</xsl:text></i><xsl:value-of select="afetiria"/><br/>

        <i><xsl:text>Εκδρομές </xsl:text></i><br/>

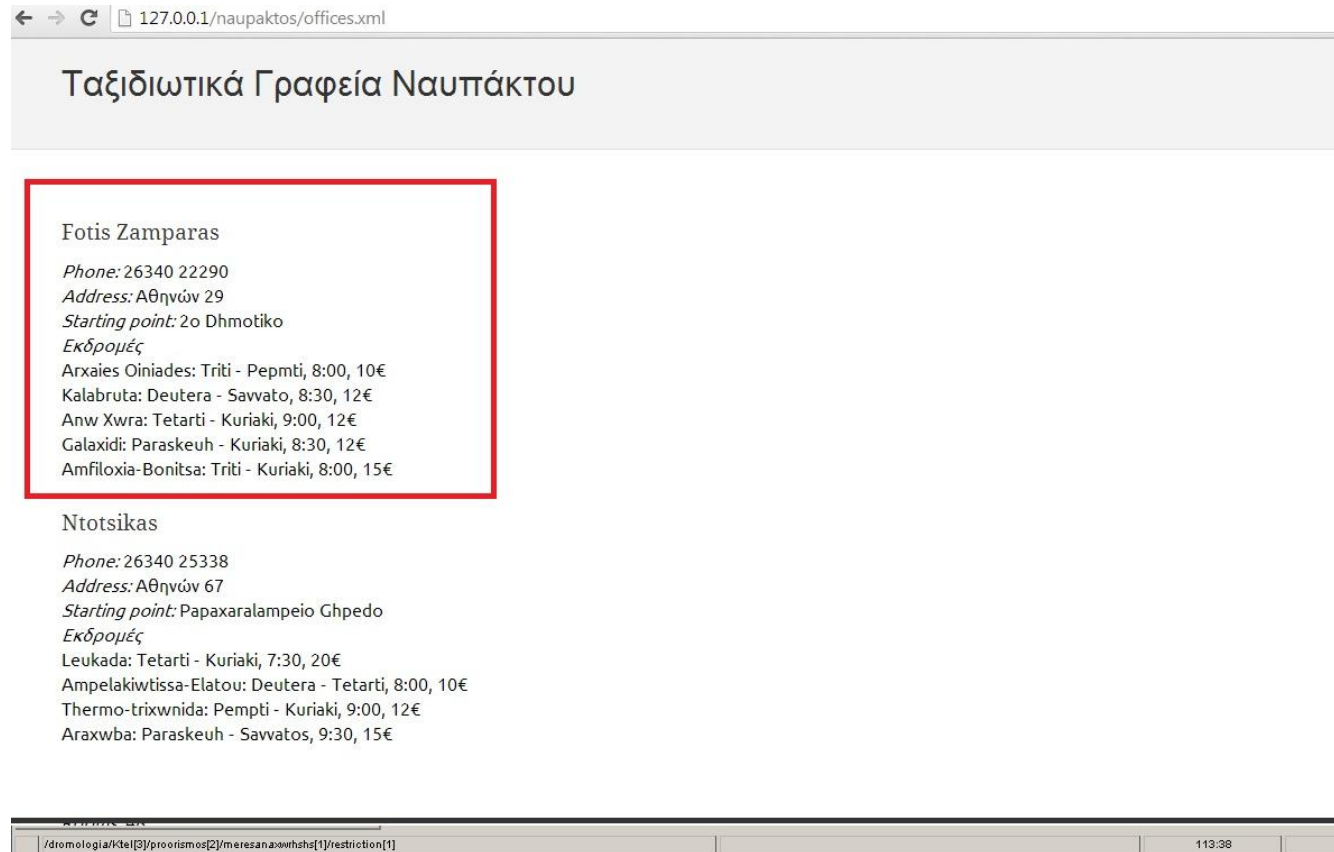
        <xsl:for-each select="./ekdromes/enumeration">
```



```

        <xsl:value-of select="@value"/>: <xsl:value-of
select="./date1/@mera"/> - <xsl:value-of select="./date/@mera"/>, <xsl:value-of
select="./hour/@wra"/>, <xsl:value-of select="./price/@poso"/><br/>
        </xsl:for-each>
    </p>

```



Εικόνα 28 Εμφάνιση αποτελεσμάτων του Xsl

Για τα μουσεία έχουμε: <xsl:for-each select="museum">

```

    <h3><xsl:value-of select="name"/></h3>

```

```

    <p>
    <i><xsl:text>Address: </xsl:text></i><xsl:value-of
select="address"/><br/>

```

```

    <i><xsl:text>Price: </xsl:text></i><xsl:value-of
select="price"/><br/>

```

```

    <i><xsl:text>Category: </xsl:text></i><xsl:value-of
select="eidosis"/><br/>

```

```

    <i><xsl:text>Phone: </xsl:text></i><xsl:value-of
select="./@phone"/><br/>

```

```

    <i><xsl:text>Hours: </xsl:text></i><xsl:value-of
select="wresleitourgias"/><br/>

```

```
<i><xsl:text>Days: </xsl:text></i><xsl:value-of  
select="meresleitourhias"/><br/>  
</p>  
</xsl:for-each>
```

Αρχική

Αξιοθέατα

Διαμονή

Υπηρεσίες

Μουσεία Ναυπάκτου

To spiti toy Mpotsari

Address: Mpotsreika

Price: free entrance

Category: istoriko

Phone: 2634029779

Hours: 9am-1pm

Days: Deutera-Paraskeuh

Mouseio Farmaki

Address: Plateia Farmaki

Price: free entrance

Category: Polemiko

Phone: 2634027084

Hours: 9am-1pm

Days: Deutera-Paraskeuh

Εικόνα 29 Αποτέλεσμα xsl για τα μουσεία

Για τα δρομολόγια έχουμε: <xsl:for-each select="Ktel">

```
<h3>ΚΤΕΛ: <xsl:value-of select="@meros"/></h3>
```

```
<p>
```

```
<i><xsl:text>Τιμή: </xsl:text></i><xsl:for-each  
select="./times/restriction/enumeration"><xsl:value-of select="@value"/>&#160;</xsl:for-  
each><br/>
```

```
<i><xsl:text>Δρομολόγια: </xsl:text></i><br/>
```

```
<xsl:for-each select="./proorismos">
```

```
<xsl:value-of select="@name"/>: <xsl:for-  
each select="./meresanaxwrhshs/restriction/enumeration"><xsl:value-of  
select="@value"/>&#160;</xsl:for-each><br/>
```

```
</xsl:for-each>
```

```
<i><xsl:text>Ωρες: </xsl:text></i><xsl:value-of  
select="./proorismos/hours"/><br/>  
</p>  
</xsl:for-each>
```

Δρομολόγια Ναυπάκτου

ΚΤΕΛ: Athina

Τιμή: Olokliro=20€ Foitiko=15€ Politekniko=10€

Δρομολόγια:

Athina-Naupaktos: Deutera Triti Tetarti Pempti Paraskeuh Sabbato Kuriaki

Naupaktos-Athina: Deutera Triti Tetarti Pempti Paraskeuh Sabbato Kuriaki

Ωρες: 07.15 - 15.30

ΚΤΕΛ: Thessaloniki

Τιμή: Olokliro=45€ Foitiko=35€ Politekniko=23€

Δρομολόγια:

Thessaloniki-Naupaktos: Deutera Triti Tetarti Pempti Paraskeuh Sabbato Kuriaki

Naupaktos-Thessaloniki: Deutera Triti Tetarti Pempti Paraskeuh Sabbato Kuriaki

Ωρες: 09:30 - 15.30

ΚΤΕΛ: Ioannina

Τιμή: Olokliro=30€ Foitiko=22€ Politekniko=15€

Δρομολόγια:

Ioannina-Naupaktos: Deutera Triti Tetarti Pempti Paraskeuh Kuriaki

Naupaktos-Ioannina: Deutera Triti Tetarti Pempti Paraskeuh Kuriaki

Ωρες: 12:00

Εικόνα 30 Αποτέλεσμα xsl για τα δρομολόγια

Για τα αξιοθέατα έχουμε: xsl:for-each select="attraction">

```
<h3><xsl:value-of select="name"/></h3>
```

```
<p>
```

```
<i><xsl:text>Location: </xsl:text></i><xsl:value-of  
select="location"/><br/>
```

```
<i><xsl:text>Open on: </xsl:text></i><xsl:value-of  
select="@leitourgies"/><br/>
```

```
<i><xsl:text>Information: </xsl:text></i><xsl:value-  
of select="information"/><br/>
```

```
</p>
```

```
</xsl:for-each>
```

Αξιοθέατα Ναυπάκτου

Roloi tis Polis

Location: Kastro

Open on:

Information: Χτισμένο το 1914 από τον Μητροπολίτη Σεραφείμ Δομβοΐτη, το Ρολόι της πόλης βρίσκεται σε προνομιακή θέση. Ανηφορίζοντας απ' το πλακόστρωτο σοκάκι πίσω απ' τον Άγιο Δημήτριο, μόλις λίγα σκαλοπάτια σας χωρίζουν απ' την πανοραμική θέα στον κόλπο της Ναυπάκτου. Αν ξεχαστείτε με τη θέα, ο ήχος του ρολογιού- που οι ντόπιοι αποκαλούν και Σεραφείμ προς τιμή του εμπνευστή του- θα σας θυμίσει την ώρα. Προτιμήστε να το επισκεφθείτε απογευματάκι, που ο ήλιος είναι πιο ήπιος και τα χρώματα πιο γλυκά.

Kastro

Location: Lofos Nafpaktou

Open on: 9am-1pm, 7pm-1am

Information: Μοναδικό, επιβλητικό, εντυπωσιακό κάστρο με πέντε επάλληλα οχυρωματικά αμυντικά διαζώματα, από την κορυφή του λόφου (Ιτς-Καλέ) μέχρι το λιμάνι. Το κάστρο της Ναυπάκτου ποτέ δεν κατακτήθηκε, η παράδοση του ήταν αποτέλεσμα πάντα διαπραγματεύσεων και συνθηκών. Από την αρχαιότητα η Ναύπακτος υπήρξε οχυρωμένη πόλη. Στα χρόνια του κράτους της Ηπείρου χτίστηκε το μεσαιωνικό κάστρο στην πρώτη του μορφή. Στην αλληλοδιαδοχή των κυριάρχων που ακολούθησε - Φράγκοι, Αλβανοί, Ιππότες της Ρόδου και Ενετοί - οι τελευταίοι το 15ο αι. το επισκεύασαν και πήρε τότε τη σημερινή του μορφή.

Εικόνα 31 Αποτέλεσμα xsl για τα αξιοθέατα

Τέλος για να δουλέψει η κράτηση για οχήματα χρειάστηκε να φτιάξουμε ένα php αρχείο το οποίο θα ενεργοποιεί την εκάστοτε κράτηση.

Κώδικας PHP

```
<?php
```

```
error_reporting(0);
```

```
function loadFile($xml, $xsl){
```

```
    $xmlDoc = new DOMDocument();
```

```
    $xmlDoc->load($xml);
```

```
    $xslDoc = new DOMDocument();
```

```
    $xslDoc->load($xsl);
```

```
    $proc = new XSLTProcessor();
```

```
    $proc->importStyleSheet($xslDoc);
```

```
    echo $proc->transformToXML($xmlDoc);
```

```
}

function updateFile($xml){
    $xmlLoad = simplexml_load_file($xml);
    $postKeys = array_keys($_POST);

    foreach($xmlLoad->children() as $x){
        foreach($_POST as $key=>$value){
            foreach($x->children() as $y){
                foreach($y->children() as $z){
                    if(strcmp($key , $z->attributes()) == 0){
                        $attr = $z->attributes();
                        $attr['available'] = "false";
                    }
                }
            }
        }
    }

    $xmlLoad->asXML($xml);
    loadFile($xml, "cars_motors.xml");
}

updateFile("cars_motors.xml");
```

?>

Επεξήγηση του κώδικα

Στο παραπάνω κώδικα έχουμε δημιουργήσει δυο μεθόδους την Loadfile και την UpdateFile οι οποίες χρησιμοποιούνται για να πραγματοποιούνται οι κρατήσεις. Πιο συγκεκριμένα, η συνάρτηση LoadFile παίρνει σαν ορίσματα τα αρχεία .xml και .xsl και μορφοποιεί το αρχείο .xml στο αρχείο .xsl. Η UpdateFile είναι η συνάρτηση που στη πραγματικότητα ενεργοποιεί τη κράτηση καθώς αλλάζει το αρχείο .xml. Στο αρχείο cars_motors.xml, το πεδίο available είναι true. Όταν ο χρήστης πατάει το κουμπί «Κράτηση», καλείται η συνάρτηση UpdateFile, έτσι το available αλλάζει από true σε false και μετά ξαναφορτώνει με την LoadFile το καινούργιο xml αρχείο και το αυτοκίνητο δεν είναι πια διαθέσιμο.

Ενοικιαζόμενα Οχήματα Ναυπάκτου

Avance Travel

Phone: 2634026250

Address: Μενεخمου 5

Αυτοκίνητα:

Κράτηση ABC1234, 1700 κυβικά, 4 θέσεις

Μηχανές:

Κράτηση AB123, 50 κυβικά, τύπος scooter

Avance Travel

Phone: 2634026250

Address: Μενεخمου 5

Αυτοκίνητα:

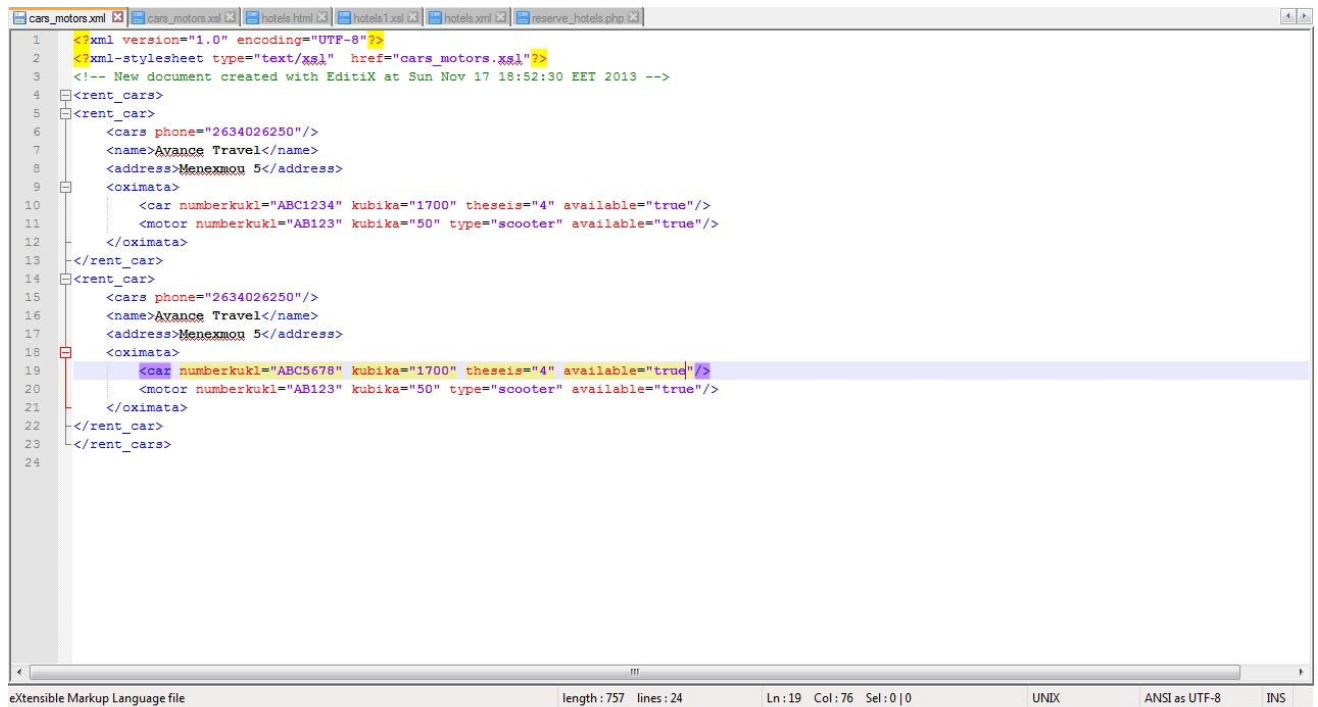
Κράτηση ABC5678, 1700 κυβικά, 4 θέσεις

Μηχανές:

Κράτηση AB123, 50 κυβικά, τύπος scooter

2014 Vanessa Douni - Eirini Loukou

Εικόνα 32 Η κράτηση



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xml-stylesheet type="text/xsl" href="cars_motors.xsl"?>
3 <!-- New document created with EditiX at Sun Nov 17 18:52:30 EET 2013 -->
4 <rent_cars>
5 <rent_car>
6 <cars phone="2634026250"/>
7 <name>Avance Travel</name>
8 <address>Μενεξμου 5</address>
9 <oximata>
10 <car numberkukl="ABC1234" kubika="1700" theseis="4" available="true"/>
11 <motor numberkukl="AB123" kubika="50" type="scooter" available="true"/>
12 </oximata>
13 </rent_car>
14 <rent_car>
15 <cars phone="2634026250"/>
16 <name>Avance Travel</name>
17 <address>Μενεξμου 5</address>
18 <oximata>
19 <car numberkukl="ABC5678" kubika="1700" theseis="4" available="true"/>
20 <motor numberkukl="AB123" kubika="50" type="scooter" available="true"/>
21 </oximata>
22 </rent_car>
23 </rent_cars>
24
```

length : 757 lines : 24 Ln : 19 Col : 76 Sel : 0 | 0 UNIX ANSI as UTF-8 INS

Εικόνα 33 Ο κώδικας πριν τη κράτηση

Ενοικιαζόμενα Οχήματα Ναυπάκτου

Avance Travel
Phone: 2634026250
Address: Μενεξμου 5

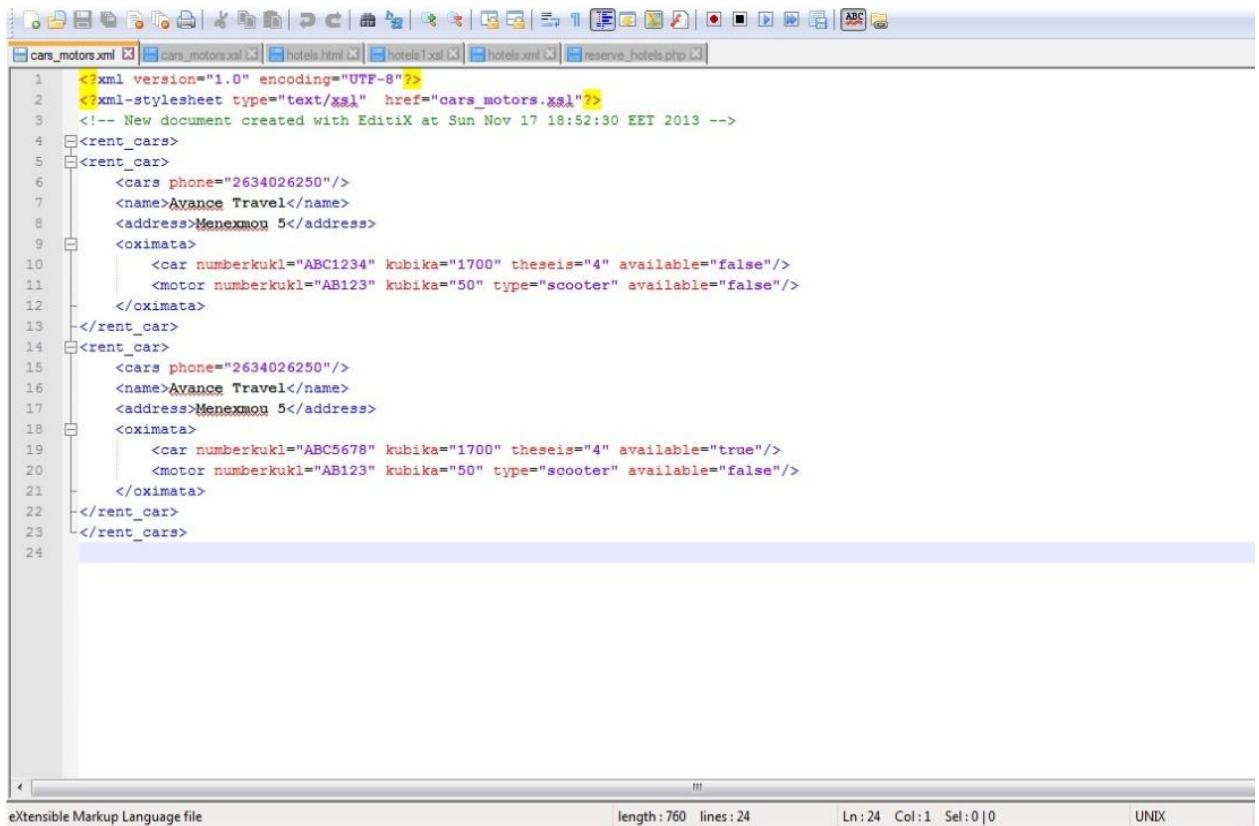
Αυτοκίνητα:
ABC1234, 1700 κυβικά, 4 θέσεις
Μηχανές:
AB123, 50 κυβικά, τύπος scooter

Avance Travel
Phone: 2634026250
Address: Μενεξμου 5

Αυτοκίνητα:
Κράτηση ABC5678, 1700 κυβικά, 4 θέσεις
Μηχανές:
AB123, 50 κυβικά, τύπος scooter

2014 Vanessa Douni - Eirini Loukou

Εικόνα 34 Γίνεται η κράτηση



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="cars_motors.xsl"?>
3 <!-- New document created with EditiX at Sun Nov 17 18:52:30 EET 2013 -->
4 <rent_cars>
5 <rent_car>
6   <cars phone="2634026250"/>
7   <name>Avance Travel</name>
8   <address>Μενεξμου 5</address>
9   <oximata>
10    <car numberkukl="ABC1234" kubika="1700" theseis="4" available="false"/>
11    <motor numberkukl="AB123" kubika="50" type="scooter" available="false"/>
12  </oximata>
13 </rent_car>
14 <rent_car>
15   <cars phone="2634026250"/>
16   <name>Avance Travel</name>
17   <address>Μενεξμου 5</address>
18   <oximata>
19    <car numberkukl="ABC5678" kubika="1700" theseis="4" available="true"/>
20    <motor numberkukl="AB123" kubika="50" type="scooter" available="false"/>
21  </oximata>
22 </rent_car>
23 </rent_cars>
24
```

length : 760 lines : 24 Ln : 24 Col : 1 Sel : 0 | 0 UNIX

Εικόνα 35 Το πεδίο available έχει αλλάξει από true σε false

5. Επίλογος

5.1 Βελτιώσεις

Η βάση που δημιουργήσαμε στο πλαίσιο της συγκεκριμένης πτυχιακής, λειτουργεί σας ένας ολοκληρωμένος τουριστικός οδηγός. Ο χρήστης έχει την δυνατότητα να συλλέξει πληροφορίες για την Ναύπακτο, δηλαδή τι ξενοδοχεία υπάρχουν στην περιοχή, να μάθει πληροφορίες για τα αξιοθέατά της, ποια κοντινά μέρη μπορεί να επισκεφτεί, καθώς επίσης μπορεί μέσα από το site να κάνει κράτηση για ένα μεταφορικό μέσο.

Μέσα στο site θα μπορούσε αργότερα να προστεθεί, πέρα από την δυνατότητα της κράτησης των οχημάτων, να προστεθεί η δυνατότητα κράτησης στα ξενοδοχεία. Πιο συγκεκριμένα ο χρήστης να είναι σε θέση να επιλέξει τι δωμάτιο θέλει, για παράδειγμα μονόκλινο, δίκλινο, με θέα ή χωρίς, σουίτα ή απλό δωμάτιο, αλλά, να έχει την επιλογή της ακύρωσης.

Οι βελτιώσεις που μπορούν να γίνουν μελλοντικά στο site, είναι αρχικά η ασφάλεια, ώστε να είναι πιο προστατευμένο από επιθέσεις. Ακόμα, θα μπορούσε να προστεθεί μια επιπλέον λειτουργία, ώστε να κρατείται ιστορικό έτσι ώστε να γνωρίζει η εκάστοτε εταιρία ποιοι είναι οι παλιοί και ποιοι οι καινούριοι πελάτες. Ένας τέτοιος τρόπος, είναι να προστεθεί στο site η δυνατότητα δημιουργίας λογαριασμού και να γίνεται μέλος ο υποψήφιος πελάτης(member). Μέσω αυτού, θα είναι σε θέση να κάνει τα δικά του σχόλια για το μέρος που επισκέφτηκε, να εκφράζει τις εντυπώσεις του για το ξενοδοχείο στο οποίο διέμεινε και να παροτρύνει νέους πελάτες να επισκεφτούν το μέρος και το ξενοδοχείο.

Τέλος, μέσα από τον τουριστικό οδηγό θα μπορούσαν να επωφεληθούν και τα ξενοδοχεία με μια πιθανή συνεργασία, διότι θα είχαν την δυνατότητα προβολής σε εικοσιτετράωρη βάση, προσθέτοντας μάλιστα τα link για τις ιστοσελίδες τους είτε φωτογραφικό υλικό.

5.2 Συμπεράσματα

Μέσα από αυτήν την εργασία, μπορέσαμε να δημιουργήσουμε ένα ολοκληρωμένο δικτυοκεντρικό σύστημα, κάτι που χρησιμοποιείται ευρέως σήμερα, μιας και όλες οι συναλλαγές – κρατήσεις – αγορές αλλά και απλές ενημερώσεις των πολιτών γίνονται πλέον μέσα από τέτοια συστήματα. Μας δόθηκε η δυνατότητα να μάθουμε και να χρησιμοποιήσουμε σύγχρονα και πολύτιμα εργαλεία για το «στήσιμο» μιας ολοκληρωμένης δουλειάς.

Στόχος μας εξάλλου ήταν μέσα από αυτήν την διαδικασία, πέρα από το να μάθουμε πώς να υλοποιούμε ένα σύστημα από το μηδέν, ήταν και η προβολή του αγαπημένου μας τόπου. Το φωτογραφικό υλικό, οι πληροφορίες αλλά και κάποια μικρά πράγματα για την ιστορία του, αποσκοπούν στο να κάνουν κάποιους ανθρώπους να γνωρίσουν αυτήν την μικρή παραθαλάσσια πόλη και να την επισκεφτούν στο μέλλον.

Τέλος θα πρέπει να το τονίσουμε πως για εμάς ήταν πολύ σημαντικό το ότι ασχοληθήκαμε με ένα πολύ ενδιαφέρον κομμάτι της επιστήμης της πληροφορικής, το οποίο μάλιστα εξελίσσεται ραγδαία και ζητείται έντονα από την αγορά εργασίας. Ευελπιστούμε να μπορέσει μελλοντικά να αξιοποιηθεί και να εμπλουτιστεί με νέα πράγματα που θα ορίσει η εξέλιξη της τεχνολογίας.

5.3 Βιβλιογραφία

- Συστήματα Βάσεων Δεδομένων - Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- Σημειώσεις Θέματα προγραμματισμού Διαδικτύου – Νίκος Παπαδάκης
- <http://www.w3schools.com/>
- <http://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B1%CE%B4%CE%AF%CE%BA%CF%84%CF%85%CE%BF>
- http://users.sch.gr/tsibinos/internet_history/history_1.html
- http://www.cs.unipi.gr/index.php?option=com_content&view=article&id=46%3A2009-04-12-10-30-15&catid=35&lang=el
- <http://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CF%8C%CE%B5%CE%BC%CF%80%CF%8C%CF%81%CE%B9%CE%BF>
- http://en.wikipedia.org/wiki/Internet_booking_engine
- <http://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CF%8C%CE%B5%CE%BC%CF%80%CF%8C%CF%81%CE%B9%CE%BF>
- Πανεπιστημιακές σημειώσεις: Η γλώσσα σήμανσης XML- Δημήτριος Σάμψων :
http://www.fme.aegean.gr/sites/default/files/dsampsom_xml_lectures-notes-dec2003.pdf
- <http://en.wikipedia.org/wiki/XQuery>
- <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-CSS.html>
- <http://el.wikipedia.org/wiki/PHP>