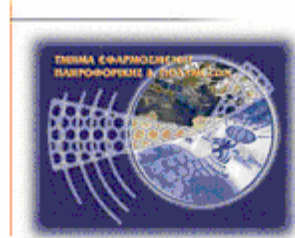




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Τίτλος:

Σύστημα για online δημοπρασίες & πωλήσεις

Λεβετσοβίτης Πέτρος (ΑΜ: 2103)

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

ΗΡΑΚΛΕΙΟ

2015

Ευχαριστίες

Διεκπεραιώνοντας την πτυχιακή εργασία θα ήθελα να ευχαριστήσω ιδιαίτερα τον καθηγητή μου κ. Παπαδάκη Νικόλαο για την βοήθεια και την καθοδήγηση που μου πρόσφερε καθ' όλη τη διάρκεια εκπόνησης της πτυχιακή εργασίας, όπως επίσης και ένα μεγάλο ευχαριστώ σε όλους όσους με στήριξαν και με συμβούλεψαν αυτό το διάστημα. Τέλος, θα ήθελα να ευχαριστήσω θερμά τους γονείς μου οι οποίοι μου στάθηκαν στις δύσκολες στιγμές και πάντα με παρότρυναν στις επιλογές μου.

Abstract

The objective of this dissertation involves the design, development and implementation of an online auction and sales system. The implementation of this system has been succeeded through the design of an online store, while satisfying a different data storing model in online shops and services. Specifically, the system is based on XML documents technology for storing of valuable data, and in combination with PHP programming language, it became feasible the dynamic recovery, processing and content display of a XML document. In addition, the use of JavaScript script language and AJAX technique provides much more interactivity in specific recovery procedures and data display, while also increasing the functionality of the online store.

Σύνοψη

Θεματικό αντικείμενο της παρούσας πτυχιακής εργασίας, είναι η σχεδίαση, η ανάπτυξη και η υλοποίηση ενός συστήματος για online δημοπρασίες και πωλήσεις. Η προσέγγιση υλοποίησης του συστήματος που ακολουθείται, γίνεται μέσω δημιουργίας ενός ηλεκτρονικού καταστήματος που ικανοποιεί ένα διαφορετικό μοντέλο αποθήκευσης δεδομένων στον τομέα των ηλεκτρονικών καταστημάτων και υπηρεσιών. Συγκεκριμένα, το σύστημα βασίστηκε πάνω στην τεχνολογία των XML εγγράφων για την αποθήκευση των χρήσιμων δεδομένων και σε συνδυασμό με την γλώσσα προγραμματισμού PHP, έγινε εφικτή η δυναμική ανάκτηση, επεξεργασία και εμφάνιση περιεχομένου από ένα έγγραφο XML. Επίσης, η χρήση της γλώσσας σεναρίων Javascript και της τεχνικής AJAX, παρέχει περισσότερη διαδραστικότητα σε συγκεκριμένες διαδικασίες ανάκτησης και εμφάνισης δεδομένων, αυξάνοντας την λειτουργικότητα του ηλεκτρονικού καταστήματος.

Πίνακας Περιεχομένων

Ευχαριστίες	ii
Abstract	iii
Σύνοψη	iv
Πίνακας Περιεχομένων	ii
Πίνακας Εικόνων.....	ii
Λίστα Πινάκων.....	ii
ΚΕΦΑΛΑΙΟ 1. Εισαγωγή	11
1.1 Περίληψη.....	11
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	11
1.3 Σκοπός και στόχοι εργασίας.....	11
1.4 Δομή πτυχιακής εργασίας	12
ΚΕΦΑΛΑΙΟ 2. Παρεμφερείς τεχνολογίες.....	13
2.1 Βάση δεδομένων(Database System)	13
2.2 Σύστημα Διαχείρισης Βάσεων Δεδομένων	13
2.2.1 MySQL.....	13
2.2.2 PostgreSQL	14
2.2.3 SQLite	14
2.2.4 Microsoft SQL Server	15
2.2.5 FileMaker	15
2.2.6 Oracle Database.....	15
2.3 Γλώσσες προγραμματισμού	16
2.3.1 Java.....	16
2.3.2 ASP και ASP.NET	17
2.3.3 JSP.....	17
2.3.4 Perl	18
2.3.5 Ruby	18
2.3.6 Python.....	19
ΚΕΦΑΛΑΙΟ 3. Ανάλυση τεχνολογικών απαιτήσεων σε θεωρητικό επίπεδο	20
3.1 Σύντομη περιγραφή τεχνολογικών απαιτήσεων.....	20
3.2 Η Γλωσσά Σήμανσης XML.....	21
3.2.1 Ιστορία της XML	21
3.2.2 Τι είναι η γλωσσά XML	21
3.2.3 Ευρεία χρήση της XML	22
3.2.4 Σύνταξη και δομή της XML.....	22
3.2.5 Πρόλογος (Prolog)	24
3.2.6 Στοιχείο ρίζα(Root element)	24
3.2.7 Στοιχείο (Element)	25
3.2.8 Ετικέτες (Tags) και Σήμανση (Markup).....	26
3.2.9 Χαρακτηριστικό (Attribute)	26
3.2.10 Δενδρική δομή και αναπαράσταση ενός XML εγγράφου.....	28
3.2.11 Προκαθορισμένες Οντότητες	29
3.2.12 Περιεχόμενα στοιχείου.....	29
3.2.13 Χώροι ονοματοδοσίας(Namespaces)	30
3.2.14 Οντότητες(entities).....	31
3.2.15 Σχόλια(Comments).....	32
3.2.16 Well-formed και valid XML έγγραφα.....	32
3.2.17 XML Parsers (αναλυτές).....	33
3.3 Τεχνολογίες Μοντελοποίησης XML εγγράφων.....	33
3.4 XML Schema	34
3.4.1 Σύνταξη και δομή ενός εγγράφου XML Schema.....	35

3.4.2 Βασικά δομικά στοιχεία των XML Schema εγγράφων.....	35
3.4.3 Συχνότητα εμφάνισης στοιχείων(minOccurs & maxOccurs).....	37
3.4.4 Δήλωση χαρακτηριστικών(attribute).....	38
3.4.5 Σχόλια σε ένα έγγραφο XML Schema	38
3.5 Η τεχνολογία μετασχηματισμού και μορφοποίησης XSL.....	39
3.5.1 Η γλώσσα μετασχηματισμού XSLT.....	40
3.5.2 Σύνταξη XSLT φύλλων στυλ.....	41
3.5.3 Άλλα στοιχεία που συνθέτουν τα XSLT φύλλα στυλ.....	44
3.6 Η γλώσσα ερωτημάτων XPath.....	45
3.7 Η γλώσσα προγραμματισμού PHP.....	47
3.7.1 Ιστορία της PHP.....	47
3.7.2 Πλεονεκτήματα της PHP.....	48
3.7.3 Η επέκταση simpleXML.....	48
3.7.4 Η επέκταση DOM στην γλώσσα PHP.....	50
3.7.5 Η έκδοση PHP 5.3 που χρησιμοποιήθηκε.....	50
3.8 Η γλώσσα JavaScript.....	51
3.8.1 Ιστορία της JavaScript.....	51
3.9 Η τεχνολογία Ajax	52
3.10 Η γλώσσα σήμανσης υπερκειμένου HTML.....	52
3.10.1 Η ιστορία της HTML	53
3.11 Η γλώσσα φύλλων στυλ CSS.....	53
3.11.1 Ιστορία της γλώσσας CSS.....	54
ΚΕΦΑΛΑΙΟ 4. Υλοποίηση συστήματος για online δημοπρασίες και πωλήσεις.....	55
4.1 Εργαλεία υλοποίησης.....	56
4.1.1 WampServer.....	56
4.1.2 Adobe Dreamweaver.....	57
4.2 Δημιουργία κατάλληλων εγγράφων XML	57
4.3 Δημιουργία κατάλληλων εγγράφων XML Schema.....	73
4.4 Δυνατότητες χρήστη.....	96
4.4.1 Εγγραφή χρήστη.....	96
4.4.2 Συμμετοχή χρήστη σε μια δημοπρασία.....	97
4.4.3 Διαδικασία αναζήτησης δημοπρασιών.....	104
4.5 Δυνατότητες διαχειριστή συστήματος.....	105
4.5.1 Εισαγωγή νέων προϊόντων	106
4.5.2 Εισαγωγή νέων δημοπρασιών	107
4.5.3 Συγκεντρωτικές αναφορές.....	109
4.6 Αρχεία συστήματος.....	110
ΚΕΦΑΛΑΙΟ 5.....	112
Βιβλιογραφία.....	113
ΠΑΡΑΡΤΗΜΑ ΠΤΥΧΙΑΚΗΣ.....	116
ΠΑΡΑΡΤΗΜΑ Α. Εγκατάσταση προγραμμάτων και αρχείων συστήματος	117
1.1 Εγκατάσταση WampServer 2.2.....	117
1.2 Εγκατάσταση των αρχείων συστήματος στο διακομιστή.....	123
ΠΑΡΑΡΤΗΜΑ Β. Διαφάνειες παρουσίασης.....	125
1.1 Διαφάνειες Παρουσίασης.....	125

Πίνακας Εικόνων

Κεφάλαιο 2

Εικόνα 2-1 : Λογότυπο MySQL.....	13
Εικόνα 2-2 : Λογότυπο PostgreSQL	14
Εικόνα 2-3 : Λογότυπο SQLite	14
Εικόνα 2-4 : Λογότυπο SQL Server.....	15
Εικόνα 2-5 : Λογότυπο FileMaker	15
Εικόνα 2-6 : Λογότυπο Oracle Database	15
Εικόνα 2-7 : Λογότυπο Java.....	15
Εικόνα 2-8 : Λογότυπο ASP.NET.....	16
Εικόνα 2-9 : Λογότυπο JSP.....	17
Εικόνα 2-10 : Λογότυπο Perl.....	18
Εικόνα 2-11 : Λογότυπο Ruby	18

Κεφάλαιο 3

Εικόνα 3-1 : Γλώσσα Σήμανσης XML.....	21
Εικόνα 3-2 : Βασική δομή XML εγγράφου.....	23
Εικόνα 3-3 : Παράδειγμα δομής XML εγγράφου ενός πελάτη	23
Εικόνα 3-4 : Σύνταξη ετικέτας XML	26
Εικόνα 3-5 : Σύνταξη χαρακτηριστικού σε ένα XML έγγραφο	26
Εικόνα 3-6 : Σύνταξη χαρακτηριστικών σε ένα XML έγγραφο.....	27
Εικόνα 3-7 : Παράδειγμα χρήσης χαρακτηριστικών XML	27
Εικόνα 3-8 : Παράδειγμα χρήσης στοιχείων XML	27
Εικόνα 3-9 : Δεντρική αναπαράσταση βασικής δομής XML εγγράφου	28
Εικόνα 3-10 : Λανθασμένη χρήση προκαθορισμένων οντοτήτων της XML	29
Εικόνα 3-11 : Ορθή χρήση προκαθορισμένων οντοτήτων της XML	29
Εικόνα 3-12 : Σύνταξη τμήματος CDATA δεδομένων	29
Εικόνα 3-13 : Σύνταξη χώρου ονοματοδοσίας με πρόθεμα	30
Εικόνα 3-14 : Σύνταξη προκαθορισμένου χώρου ονοματοδοσίας	30
Εικόνα 3-15 : Παράδειγμα χρήσης χώρου ονοματοδοσίας με πρόθεμα.....	30
Εικόνα 3-16 : Παράδειγμα χρήσης προκαθορισμένου χώρου ονοματοδοσίας	31
Εικόνα 3-17 : Σύνταξη δήλωσης οντότητας σε ένα έγγραφο XML	31
Εικόνα 3-18 : Σύνταξη αναφοράς οντότητας σε ένα XML έγγραφο	32
Εικόνα 3-19 : Σύνταξη σχόλιου σε ένα έγγραφο XML.....	32
Εικόνα 3-20 : Παράδειγμα ενός XML εγγράφου	35
Εικόνα 3-21 : Παράδειγμα ενός XML Schema εγγράφου.....	36
Εικόνα 3-22 : Παράδειγμα σύνταξης minOccurs και maxOccurs μέσα σε ένα στοιχείο	37
Εικόνα 3-23 : Παράδειγμα σύνταξης χαρακτηριστικού XML Schema.....	38
Εικόνα 3-24 : Σύνταξη σχολίων σε ένα XML Schema	38
Εικόνα 3-25 : Μη μορφοποιημένη εμφάνιση εγγράφου XML.....	39
Εικόνα 3-26 : Διαδικασία μετασχηματισμού XML εγγράφου	40
Εικόνα 3-27 : XML έγγραφο για τους μαθητές μιας τάξης.....	41
Εικόνα 3-28 : Παράδειγμα σύνταξης XSLT εγγράφου	42
Εικόνα 3-29 : Μετασχηματισμός XML εγγράφου σε HTML μορφή.....	42
Εικόνα 3-30 : XML έγγραφο για προσωπική λίστα βιβλίων.....	46
Εικόνα 3-31 : PHP Λογότυπο.....	47
Εικόνα 3-32 : JavaScript Λογότυπο	51
Εικόνα 3-33 : Ajax Λογότυπο	52
Εικόνα 3-34 : HTML Λογότυπο.....	52
Εικόνα 3-35 : Ορισμός σύνταξης HTML ετικέτας	53
Εικόνα 3-36 : CSS Λογότυπο	53

Κεφάλαιο 4

Εικόνα 4-1 : WampServer Λογότυπο	56
Εικόνα 4-2 : Adobe Dreamweaver λογότυπο	57
Εικόνα 4-3 : Στιγμιότυπο του αρχείου products.xml.....	58
Εικόνα 4-4 : Στιγμιότυπο του αρχείου pelates.xml	60
Εικόνα 4-5 : Στιγμιότυπο του αρχείου auctions.xml	61
Εικόνα 4-6 : Στιγμιότυπο του αρχείου offers.xml	62
Εικόνα 4-7 : Στιγμιότυπο του αρχείου finished_auctions.xml	64
Εικόνα 4-8 : Στιγμιότυπο XSLT μετασχηματισμού του εγγράφου auction_products.xml	66
Εικόνα 4-9 : Στιγμιότυπο του αρχείου auction_products.xml	67
Εικόνα 4-10 : Κριτήρια αναζήτησης δημοπρασιών	68
Εικόνα 4-11 : Στιγμιότυπο του αρχείου search.xml	68
Εικόνα 4-12 : Αποτελέσματα αναζήτησης με βάση την εταιρία προϊόντος.....	70
Εικόνα 4-13 : Αποτελέσματα αναζήτησης με βάση την κατηγορία προϊόντος.....	70
Εικόνα 4-14 : Αποτελέσματα αναζήτησης με βάση ένα συγκεκριμένο προϊόν	71
Εικόνα 4-15 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση την εταιρία προϊόντος	71
Εικόνα 4-16 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση την κατηγορία προϊόντος	71
Εικόνα 4-17 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση ένα συγκεκριμένο προϊόν.....	72
Εικόνα 4-18 : Στιγμιότυπο αναφοράς των δημοπρασιών που έληξαν άγονες.....	73
Εικόνα 4-19 : Στιγμιότυπο του αρχείου report_barren.xml.....	73
Εικόνα 4-20 : Σχηματική αποτύπωση του αρχείου products.xsd	74
Εικόνα 4-21 : Περιγραφή του στοιχείου-ρίζα products. Αρχείο products.xsd	74
Εικόνα 4-22 : Περιγραφή του στοιχείου product. Αρχείο products.xsd.....	75
Εικόνα 4-23 : Περιγραφή του στοιχείου product_id. Αρχείο products.xsd.....	76
Εικόνα 4-24 : Περιγραφή του στοιχείου price. Αρχείο products.xsd	76
Εικόνα 4-25 : Περιγραφή του στοιχείου product_name. Αρχείο products.xsd	77
Εικόνα 4-26 : Περιγραφή του στοιχείου category. Αρχείο products.xsd	78
Εικόνα 4-27 : Περιγραφή του στοιχείου corporation. Αρχείο products.xsd.....	79
Εικόνα 4-28 : Σχηματική αποτύπωση του αρχείου pelates.xsd.....	79
Εικόνα 4-29 : Περιγραφή του στοιχείου-ρίζα pelates. Αρχείο pelates.xsd	80
Εικόνα 4-30 : Περιγραφή του στοιχείου pelaths. Αρχείο pelates.xsd	80
Εικόνα 4-31 : Περιγραφή του στοιχείου user_id. Αρχείο pelates.xsd.....	81
Εικόνα 4-32 : Περιγραφή του στοιχείου username. Αρχείο pelates.xsd	82
Εικόνα 4-33 : Περιγραφή του στοιχείου password. Αρχείο pelates.xsd.....	82
Εικόνα 4-34 : Περιγραφή του στοιχείου name. Αρχείο pelates.xsd	83
Εικόνα 4-35 : Περιγραφή του στοιχείου address. Αρχείο pelates.xsd.....	83
Εικόνα 4-36 : Περιγραφή του στοιχείου phone. Αρχείο pelates.xsd.....	84
Εικόνα 4-37 : Περιγραφή του στοιχείου credit_card_number. Αρχείο pelates.xsd	85
Εικόνα 4-38 : Περιγραφή του στοιχείου type. Αρχείο pelates.xsd.....	85
Εικόνα 4-39 : Σχηματική αποτύπωση του αρχείου auctions.xsd.....	86
Εικόνα 4-40 : Περιγραφή του στοιχείου-ρίζα auctions. Αρχείο auctions.xsd	87
Εικόνα 4-41 : Περιγραφή του στοιχείου auction. Αρχείο auctions.xsd.....	87
Εικόνα 4-42 : Περιγραφή του στοιχείου auction_id. Αρχείο auctions.xsd.....	88
Εικόνα 4-43 : Περιγραφή του στοιχείου time_start. Αρχείο auctions.xsd	88
Εικόνα 4-44 : Στιγμιότυπο πίνακα δημοπρασιών	89
Εικόνα 4-45 : Δημιουργία pattern για το στοιχείο time_start. Αρχείο auctions.xsd	89
Εικόνα 4-46 : Περιγραφή του στοιχείου start_price. Αρχείο auctions.xsd.....	90
Εικόνα 4-47 : Περιγραφή του στοιχείου auction_status. Αρχείο auctions.xsd.....	91
Εικόνα 4-48 : Σχηματική αποτύπωση του αρχείου offers.xsd.....	91
Εικόνα 4-49 : Περιγραφή του στοιχείου-ρίζα offers. Αρχείο offers.xsd	92
Εικόνα 4-50 : Περιγραφή του στοιχείου offer. Αρχείο offers.xsd.....	92
Εικόνα 4-51 : Περιγραφή του στοιχείου offer_id. Αρχείο offers.xsd.....	93
Εικόνα 4-52 : Περιγραφή του στοιχείου price. Αρχείο offers.xsd	93
Εικόνα 4-53 : Περιγραφή του στοιχείου id_product. Αρχείο offers.xsd	94
Εικόνα 4-54 : Περιγραφή του στοιχείου id_user. Αρχείο offers.xsd.....	94

Εικόνα 4-55 : Περιγραφή του στοιχείου offer_time. Αρχείο offers.xsd.....	95
Εικόνα 4-56 : Ιστοσελίδα εγγραφής χρήστη	96
Εικόνα 4-57 : Ιστοσελίδα παρουσίασης δημοπρασιών	98
Εικόνα 4-58 : URL διεύθυνση μιας συγκεκριμένης δημοπρασίας.....	98
Εικόνα 4-59 : Στιγμιότυπο δημοπρασίας πριν την έναρξη της.....	99
Εικόνα 4-60 : Στιγμιότυπο δημοπρασίας μετά την έναρξη. Χρονόμετρο δέκα λεπτών.....	102
Εικόνα 4-61 : Παράταση χρόνου δημοπρασίας. Χρονόμετρο πέντε λεπτών	103
Εικόνα 4-62 : Στιγμιότυπο λήξης μίας δημοπρασίας	103
Εικόνα 4-63 : Ιστοσελίδα αναζήτησης δημοπρασιών	104
Εικόνα 4-64 : Αποτελέσματα μίας αναζήτησης δημοπρασιών	105
Εικόνα 4-65 : Ιστοσελίδα εισαγωγής νέων προϊόντων. Δικαιώματα διαχειριστή.....	106
Εικόνα 4-66 : Επιτυχής εισαγωγή νέου προϊόντος στο αρχείο products.xml.....	107
Εικόνα 4-67 : Ιστοσελίδα εισαγωγής νέων δημοπρασιών. Δικαιώματα διαχειριστή	108
Εικόνα 4-68 : Επιτυχής εισαγωγή μίας νέας δημοπρασίας στο αρχείο auction.xml	109
Εικόνα 4-69 : Παρουσίαση συγκεντρωτικών αναφορών συστήματος.....	109

Λίστα Πινάκων

Κεφάλαιο 3

Πίνακας 3-1 : Αντιστοίχιση τιμών minOccurs και maxOccurs.....	37
---	----

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η σχεδίαση, ανάπτυξη και υλοποίηση ενός συστήματος για online δημοπρασίες και πωλήσεις. Για την υλοποίηση του συστήματος αυτού, χρησιμοποιήθηκαν συνδυαστικά διαφορές τεχνολογίες και γλώσσες προγραμματισμού, ενώ για την αποθήκευση των δεδομένων χρησιμοποιήθηκε εξολοκλήρου η γλώσσα σήμανσης XML(Extensible Markup Language).

Συγκεκριμένα η τεχνολογία μοντελοποίησης για την περιγραφή των XML εγγράφων που εφαρμόστηκε είναι η τεχνολογία XML schemas, ενώ για την μορφοποίηση και εμφάνιση περιεχομένου πληροφορίας από ένα αρχείο XML χρησιμοποιήθηκε η γλώσσα μετασχηματισμού XSLT(Extensible Stylesheet Language Transformations). Για την διαχείριση των δεδομένων, το σχεδιασμό και τις διαδραστικές δυνατότητες της ιστοσελίδας χρησιμοποιήθηκαν οι γλώσσες προγραμματισμού PHP(server-side scripting language), JAVASCRIPT(client-side), η τεχνική AJAX(Asynchronous Javascript and XML) καθώς και η γλώσσα σήμανσης υπερκειμένου HTML. Η μορφοποίηση και ο καθορισμός του τρόπου εμφάνισης κάθε ιστοσελίδας γίνεται με χρήση CSS(Cascading Style Sheets).

Σαν τελικό αποτέλεσμα ένας χρήστης θα έχει τη δυνατότητα να γραφτεί στην ιστοσελίδα, να κάνει αναζήτηση προϊόντων που αντιστοιχούν σε μια δημοπρασία, καθώς και να συμμετέχει σε μια πώληση προϊόντος μέσω δημοπρασίας σε πραγματικό χρόνο.

1.2 Κίνητρο για την διεξαγωγή της εργασίας

Λόγο της ραγδαίας εξάπλωσης των ηλεκτρονικών καταστημάτων και υπηρεσιών εμπορίου, τα τελευταία χρόνια έγινε αναγκαία η μελέτη και η ερευνά τόσο σε νέες όσο και σε παλιές τεχνολογίες, αποσκοπώντας η πληροφορία που μεταφέρετε να είναι πιο ασφαλής, πιο ευέλικτη ανάμεσα σε διαφορετικές τεχνολογίες και πιο γρήγορη κατά τη μεταφορά της.

Κίνητρο για τη διεξαγωγή της εργασίας αυτής είναι η προσέγγιση υλοποίησης μιας εφαρμογής ηλεκτρονικών δημοπρασιών και πωλήσεων χρησιμοποιώντας την γλώσσα σήμανσης XML ως υπόβαθρο που θα κρατά αποθηκευμένες όλες τις χρήσιμες πληροφορίες. Η πλειοψηφία αυτών των εφαρμογών για την αποθήκευση, την ανάκτηση και την ενημέρωση δεδομένων, χρησιμοποιούν ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, όπως η MySQL, για να μπορέσουν να διαχειριστούν τα δεδομένα τους και να είναι αυτόνομες ως προς τις γλώσσες προγραμματισμού.

Στόχος μας μέσα από την υλοποίηση αυτή, είναι να αξιολογήσουμε τις δυνατότητες της γλώσσας σήμανσης XML, την επεκτασιμότητα της για νέους τρόπους υλοποίησης καθώς και ποιες σύγχρονες τεχνολογίες και γλώσσες προγραμματισμού μπορούν να υποστηριχτούν για την ανάπτυξη μιας διαδικτυακής εφαρμογής ηλεκτρονικών δημοπρασιών και πωλήσεων χωρίς τη χρήση κάποιου συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων.

1.3 Σκοπός και στόχοι εργασίας

Σκοπός της εργασίας αυτής, είναι η σχεδίαση η ανάπτυξη και η υλοποίηση ενός συστήματος για διαδικτυακές πωλήσεις προϊόντων μέσω δημοπρασιών. Για την αποθήκευση και την ανάκτηση πληροφορίας, χρησιμοποιείτε τη γλώσσα σήμανσης XML σε συνδυασμό με τις τεχνολογίες που την υποστηρίζουν(XML Schema και XSL). Η διαχείριση, επεξεργασία και εμφάνιση της πληροφορίας αυτής σε ένα χρήστη γίνεται με τη χρήση των τεχνολογιών PHP, JAVASCRIPT, AJAX, HTML, CSS.

Μέσο αυτής της υλοποίησης στόχος μας είναι να αξιοποιήσουμε όσο περισσότερο μπορούμε την γλώσσα σήμανσης XML και των τεχνολογιών που την υποστηρίζουν(XSL, XML schemas, DTD).

1.4 Δομή πτυχιακής εργασίας

Στο πρώτο κεφάλαιο γίνεται μια σύντομη περίληψη ως προς τη δομή, το σκοπό, και τους στόχους της παρούσας πτυχιακής εργασίας.

Στο δεύτερο κεφάλαιο γίνεται μια σύντομη αναφορά σε παρεμφερείς τεχνολογίες που μπορούν να υλοποιηθούν με άλλο σχεδιασμό το σύστημα για online δημοπρασίες και πωλήσεις.

Στο τρίτο κεφάλαιο γίνεται μία ανάλυση σε θεωρητικό επίπεδο, όλων των τεχνολογιών που χρησιμοποιήθηκαν στην παρούσα πτυχιακή εργασία.

Στο τέταρτο κεφάλαιο παρουσιάζονται τα βήματα υλοποίησης του συστήματος για online δημοπρασίες και πωλήσεις.

Στο πέμπτο κεφάλαιο αναφέρονται, τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας πτυχιακής εργασίας καθώς και οι σκέψεις για μελλοντικές επεκτάσεις του συστήματος.

ΚΕΦΑΛΑΙΟ 2

Παρεμφερείς τεχνολογίες

Η επιλογή των τεχνολογιών XML, PHP, JAVASCRIPT, AJAX και HTML δεν αποτελεί τον βέλτιστο και μοναδικό τρόπο για την υλοποίηση του θέματος της πτυχιακής εργασίας. Στο κεφάλαιο αυτό θα κάνουμε μια σύντομη αναφορά σε παρεμφερείς τεχνολογίες που μπορούν κάλλιστα να υλοποιήσουν το σύστημα για online δημοπρασίες και πωλήσεις.

2.1 Βάση δεδομένων(Database System)

Μια βάση δεδομένων είναι το σύνολο των πληροφοριών, δεδομένων τα οποία έχουν οργανωθεί με τέτοιο τρόπο ώστε η χρήση τους να είναι γρήγορη και αποτελεσματική. Στον κόσμο των ηλεκτρονικών συστημάτων και πληροφοριών ορίζεται ως η οργανωμένη συλλογή λογικά συσχετιζόμενων δεδομένων (πληροφορίες). Τα πιο σημαντικά πλεονεκτήματα των βάσεων δεδομένων είναι ο μη πλεονασμός της πληροφορία καθώς και η δυνατότητα πρόσβασης τους από πολλούς χρήστες με διαφόρους τρόπους και προγράμματα.

2.2 Σύστημα Διαχείρισης Βάσεων Δεδομένων

Ένα σύστημα διαχείρισης βάσεων δεδομένων(Database Management System) είναι ένα λογισμικό(software) που επιτρέπει σε ένα χρήστη να διαχειριστεί μια βάση δεδομένων. Ο χρήστης έχει την δυνατότητα να δημιουργήσει, να τροποποιήσει, να διαγράψει και γενικά να επεξεργαστεί μια βάση με ασφαλή τρόπο παρέχοντας έλεγχο πρόσβασης σε αυτή

Το σύστημα διαχείρισης βάσεων δεδομένων βρίσκεται στην ουσία ανάμεσα σε μια βάση και στην εφαρμογή που χρησιμοποιεί ένας χρήστης ώστε να γίνει δυνατή η επικοινωνία με τη βάση. Ένα ΣΔΒΔ μπορεί να λειτουργεί σε έναν υπολογιστή ή σε ένα δίκτυο υπολογιστών και μπορεί να το διαχειριστεί ένας χρήστης ή και περισσότεροι. Παρακάτω θα κάνουμε μια σύντομη ανάπτυξη των πιο σημαντικών και δημοφιλέστερων συστημάτων διαχείρισης βάσεων δεδομένων.

2.2.1 MySQL

Σχεδιαστής της MySQL είναι ο Monty Widenius ο οποίος έδωσε την συγκεκριμένη επωνυμία από το όνομα της κόρης του My. Το δελφίνι που εμφανίζεται στο λογότυπο της MySQL ονομάζεται Sakila και πήρε την ονομασία του μετά από διαγωνισμό στον οποίο πήραν μέρος πολλοί χρήστες. Μέσα από μια τεράστια λίστα ονομάτων, νικητής ήταν ο ανοιχτού κώδικα προγραμματιστής λογισμικού Ambrose Twebaze από την Σουαζιλάνδη, Αφρική. Σύμφωνα με τον Ambrose Twebaze το όνομα Sakila είναι γένους θηλυκού και έχει ρίζες από τη SiSwati, την τοπική διάλεκτο της Σουαζιλάνδης.



Εικόνα 2-1 : Λογότυπο MySQL

Αρχικά η όλη ιδέα σχεδίασης της MySQL βασίστηκε στη δυνατότητα να μπορεί το MySQL σύστημα βάσεων δεδομένων να συνδεθεί με πίνακες χρησιμοποιώντας τις χαμηλού επιπέδου ρουτίνες (ISAM). Μετά από ένα σύνολο δοκιμών, αποφασίστηκε ότι το MySQL δεν ήταν αρκετά γρήγορο και ευέλικτο για την εξυπηρέτηση των αναγκών σχεδίασης της MySQL. Αυτό είχε σαν αποτέλεσμα την δημιουργία ενός νέου περιβάλλοντος διεπαφής της SQL(Structure Query Language) στην βάση δεδομένων της MySQL με σχεδόν την ίδια διεπαφή API της MSQL. Το API σχεδιάστηκε για την ενεργοποίηση “third party” κώδικα που γράφτηκε για χρήση με MSQL και δυνατότητα μεταφοράς για χρήση με MySQL.

Το MySQL είναι ένα από τα δημοφιλέστερα σύστημα ανοιχτού κώδικα (Open source) διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο αναπτύσσεται, υποστηρίζεται και διανέμεται από τη MySQL A.B. Το σύστημα αυτό είναι πολύ γρήγορο και ευέλικτο και επιτρέπει την αποθήκευση,

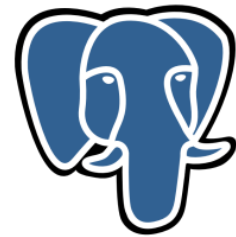
ταξινόμηση, αναζήτηση και ανάκληση δεδομένων αποτελεσματικά και αξιόπιστα. Χρησιμοποιεί την SQL γλώσσα ερωταπαντήσεων για βάσεις δεδομένων και μέσω του MySQL διακομιστή μπορεί να ελέγχει την πρόσβαση στα δεδομένα ώστε να υπάρχει η δυνατότητα να δουλεύουν πολλοί χρήστες ταυτόχρονα, διασφαλίζοντας γρήγορη πρόσβαση μόνο σε πιστοποιημένους χρήστες. Όπως προαναφέραμε είναι ανοιχτού κώδικα σύστημα βάσεων δεδομένων το οποίο επιτρέπει στον οποιονδήποτε να την χρησιμοποιήσει και να την διαμορφώσει ανάλογα με τις ανάγκες του, πάντα μέσα στα πλαίσια των αδειών που έχουν καθιερωθεί.

Σήμερα η MySQL ανήκει στην Oracle Corporation και είναι ευρέως διαδομένη βάση δεδομένων για ιστοσελίδες και διαδικτυακά προγράμματα. Προτιμάτε από μεγάλες εταιρίες υπηρεσιών όπως η Google, Wikipedia, Facebook, Twitter, Youtube και πολλές άλλες. Μπορεί να λειτουργήσει σε περιβάλλοντα Windows, Linux, Unix και Mac.

2.2.2 PostgreSQL

Η PostgreSQL η οποία αρχικά ονομαζόταν Postgres αναπτύχθηκε στο πανεπιστήμιο του Berkley στην Καλιφόρνια από τον καθηγητή επιστήμης υπολογιστών Michael Stonebraker. Η εφαρμογή της Postgres ξεκίνησε το 1986 και βασίστηκε στην δημιουργία βάσεων δεδομένων σύμφωνα με τους κανόνες των συστημάτων διαχείρισης σχεσιακών βάσεων δεδομένων(RDBMS). Από το 1986 έως και το 1994 τροποποιήθηκε με διάφορες επεκτάσεις και υλοποιήσεις πολλών και διαφορετικών εφαρμογών έρευνας όπως σε τεχνολογίες " Object relational " προσπαθώντας να φέρει νέες προοπτικές στην ιδέα των βάσεων δεδομένων. Το 1995 δύο μαθητές από το εργαστήριο του Michael Stonebraker ο Andrew Yu και Jolly Chen αντικατέστησαν την Postgres με POSTQUEL προσθέτοντας την επέκταση της. Το νέο σύστημα μετονομάστηκε σε Postgres95. Στη συνέχεια το 1996 για να γίνει πιο ορατή η σχέση της Postgres με την SQL, έγινε αντικατάσταση του ονόματος σε PostgreSQL και ορίστηκε η αρίθμηση των εκδόσεων να ξεκινά από το νούμερο 6.0 ώστε να συμβαδίζει με την αρίθμηση του αρχικού σχεδίου της Postgres του πανεπιστημίου του Berkley.

Σήμερα η PostgreSQL είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων προσφέροντας πολλές δυνατότητες στους χρήστες τόσο σε επίπεδο επεκτασιμότητας όσο και σε επίπεδο ασφάλειας και γρήγορης διαχείρισης. Είναι γραμμένη σε γλώσσα προγραμματισμού C και είναι παραγωγή της PostgreSQL Development Group. Η PostgreSQL μπορεί να λειτουργήσει με όλα τα λειτουργικά συστήματα όπως Windows, Mac, Linux, Unix. Είναι ACID συμβατή και διαθέτει εγγενείς διεπαφές προγραμματισμού (native programming interfaces) για Java, C/C++, Python, Ruby, Perl, ODBC κ.α. Ο πηγαίος κώδικας της είναι ελεύθερος κάτω από την άδεια ανοιχτού κώδικα(open source) για τον οποιονδήποτε θέλει να τη χρησιμοποιήσει.



PostgreSQL

Εικόνα 2-2 : Λογότυπο PostgreSQL

2.2.3 SQLite

Η SQLite είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και σχεδιάστηκε από τον D. Richard Hipp το 2000. Εκείνη την περίοδο D. Richard Hipp εργαζόταν για την αεροπορική εταιρία άμυνας General Dynamics με συμβόλαιο που είχε με το ναυτικό των Ηνωμένων Πολιτειών. Η αρχική ιδέα σχεδίασης ήταν να μπορεί ένα πρόγραμμα να χειριστεί την SQLite χωρίς να χρειάζεται να γίνει εγκατάσταση κάποιου συστήματος διαχειριστής βάσεων δεδομένων ούτε να υπάρχει κάποιος διαχειριστής βάσης.

Η σχεσιακή βάση δεδομένων SQLite είναι γραμμένη σε γλώσσα προγραμματισμού C και σε αντίθεση με τα υπόλοιπα συστήματα διαχείρισης βάσεων δεδομένων δεν είναι ένα σύστημα πελάτη-εξυπηρετητή(client-server) αλλά είναι ενσωματωμένη στο τέλος του προγράμματος.



Εικόνα 2-3 : Λογότυπο SQLite

Σήμερα είναι ευρέως διαδομένη και χρησιμοποιείται από τους πιο δημοφιλείς περιηγητές όπως ο Mozilla Firefox, Safari, Google Chrome, Skype.

2.2.4 Microsoft SQL Server

Ο SQL Server είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που σχεδιαστικό και αναπτύχθηκε από την Microsoft. Κυκλοφόρησε για πρώτη φορά το 1989 σε συνεργασία με την Sybase. Είναι γραμμένη σε γλώσσα προγραμματισμού C και σήμερα βγαίνει σε διάφορες εκδόσεις που εξυπηρετούν διαφορετικούς σκοπούς. Χρησιμοποιεί σαν κυρία γλώσσα ερωταπαντήσεων (Query Language) την T-SQL και ANSI SQL. Η βασική του λειτουργία είναι να αποθηκεύει και να ανακτά δεδομένα όπως του ζητούνται από διάφορα προγράμματα που μπορεί να είναι στον ίδιο υπολογιστή ή σε κάποιον άλλον υπολογιστή που βρίσκεται στο διαδίκτυο.



Εικόνα 2-4 : Λογότυπο SQL Server

2.2.5 FileMaker

Ο FileMaker είναι ένα ανεξάρτητης πλατφόρμας σύστημα σχεσιακών βάσεων δεδομένων που διανέμεται από την FileMaker Inc, θυγατρική της Apple. Η ιστορία της ξεκινά στις αρχές του 1980 όπου αρχικά είχε σχεδιαστεί από την Nashoba Systems ως πρόγραμμα σε MS-DOS και ονομαζόταν Nutshell. Με την εισαγωγή της Macintosh τα επόμενα χρόνια η Nashoba Systems συνδύασε τη βασική μηχανή δεδομένων με ένα γραφικό περιβάλλον χρήστη (GUI). Το νέο πρόγραμμα ονομάστηκε FileMaker και κυκλοφόρησε τον Απρίλιο του 1985 με την υποστήριξη της πλατφόρμας Apple Macintosh.



Εικόνα 2-5 : Λογότυπο FileMaker

Σήμερα είναι διαθέσιμη για Microsoft Windows (Windows 7sp1, Windows 8) και Macintosh OS ενσωματώνοντας ένα σύστημα βάσεων δεδομένων με ένα γραφικό περιβάλλον και δυνατότητες ασφάλειας, δίνοντας πολύ εύκολα την δυνατότητα σε ένα χρήστη να παραμετροποίηση την βάση σύροντας νέα στοιχεία στα σχεδιαγράμματα ή στις φόρμες.

2.2.6 Oracle Database

Η Oracle Database γνωστή ως Oracle RDBMS ή πιο απλά Oracle σχεδιάστηκε το 1977 από τον Larry Ellison και τους δύο πρώην συναδέλφους του Bob Miner και Ed Oates. Το όνομα Oracle προήλθε από τον κωδικό ονόματος ενός κεφαλαιοποιητικού σχεδίου πάνω στο οποίο είχε δουλέψει ο Larry Ellison όταν είχε εργαστεί στην εταιρία Ampex. Η Oracle είναι ένα αντικείμενο-σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (object-relational database management system) που αποτελείται από δεδομένα αποθηκευμένα σε αρχεία που η προσπέλασή τους γίνεται μέσω του στιγμιότυπου(instance) και από διαδικασίες που αλληλεπιδρούν με τα δεδομένα των αρχείων.

Σήμερα η Oracle Database ανήκει στην εταιρία Oracle Corporation και είναι ένα από τα πιο διαδεδομένα ΣΔΒΔ που αποτελεί ένα σύνολο από συνεργαζόμενα λογισμικά και υπηρεσίες. Υποστηρίζει σχεδόν όλες τις πλατφόρμες και είναι ιδιαίτερα ευέλικτη. Ένα σημαντικό πλεονέκτημα είναι ο συνδυασμός της με την γλώσσα προγραμματισμού Java, όπου στην περίπτωση αυτή οι εφαρμογές είναι πιο αξιόπιστες, γρήγορες και ασφαλέστερες. Η σύνδεση της βάσης με την γλώσσα προγραμματισμού Java γίνεται μέσω του JDBC. Το JDBC είναι μια διαπλατφορμική διεπιφάνεια που



Εικόνα 2-6 : Λογότυπο Oracle Database

έχει καθιερωθεί ως πρότυπο API για την πρόσβαση και την ανάκτηση δεδομένων από σχεσιακές βάσεις δεδομένων όπως είναι η Oracle, MySQL, IBM κ.α.

2.3 Γλώσσες προγραμματισμού

Για την υλοποίηση του συστήματος για online δημοπρασίες και πωλήσεις θα μπορούσαν να χρησιμοποιηθούν και να συνδυαστούν διαφορές γλώσσες προγραμματισμού. Η δημιουργία και ανάπτυξη του συστήματος μπορεί να γίνει με μια εφαρμογή γραμμένη σε μια ή περισσότερες γλώσσες προγραμματισμού που θα τρέχει σε ένα φυλλομετρητή ή μπορεί να είναι μια εφαρμογή που εγκαθιστά ο χρήστης στον υπολογιστή του για να μπορέσει να την λειτουργήσει. Οι πιο διαδεδομένες γλώσσες για την δημιουργία τέτοιων εφαρμογών, που υποστηρίζουν την επεξεργασία και προσπέλαση XML εγγράφων περιγράφονται παρακάτω.

2.3.1 Java

Η Java σχεδιάστηκε και αναπτύχθηκε από τον James Gosling και τη Sun Microsystems. Το 1991 ο James Gosling, Mike Sheridan, και Patrick Naughton ξεκίνησαν την ανάπτυξη της Java που αρχικά είχε σχεδιαστεί για διαδραστική τηλεόραση. Η αρχική της ονομασία ήταν Oak και είχε πάρει το όνομα της από ένα δέντρο βελανιδιάς που βρισκόταν έξω από το σπίτι του James Gosling. Για εκείνη τη χρονική στιγμή σαν γλώσσα ήταν πολύ εξελιγμένη για υπηρεσίες καλωδιακής τηλεόρασης και μέσο ενός νέου σχεδίου που ονομάστηκε Green πήρε το τελικό της όνομα Java το οποίο έχει παραμείνει και μέχρι σήμερα. Η σύνταξη της προέρχεται από τη C και την C++ που ήταν τα μόνα εργαλεία της εποχής, επομένως έχουν πολλά κοινά μεταξύ τους με την διάφορα ότι η Java έχει λιγότερες εγκαταστάσεις χαμηλού επιπέδου. Το όνομα Java προήλθε από την ονομασία της αγαπημένης ποικιλίας καφέ που προτιμούσαν να πίνουν οι πρώτοι σχεδιαστές της. Το 1995 η Sun Microsystems κυκλοφόρησε τη Java 1.0 ως πυρήνα της πλατφόρμας Java, με τον πρώτο μεταγλωττιστή να είναι γραμμένος σε γλώσσά C++ από τον James Gosling. Τα επόμενα χρόνια η εξέλιξη της γλώσσας ήταν πολύ σημαντική στον κλάδο της πληροφορικής και του προγραμματισμού εισάγοντας τη δυνατότητα υποστήριξης κίνησης, ήχου, βίντεο ακόμα και παιχνιδιών σε μια ιστοσελίδα. Ένα ακόμα πλεονέκτημα που πρόσφερε ήταν η δυνατότητα αλληλεπίδρασης των χρηστών με την ιστοσελίδα και ως εκ τούτου με την ίδια την εφαρμογή.

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που παρέχει ανεξαρτησία ως προς το λειτουργικό σύστημα και την πλατφόρμα που λειτουργεί. Με αυτό τον τρόπο ο κώδικας Java που γράφεται από τους προγραμματιστές, μπορεί να λειτουργήσει σε οποιαδήποτε πλατφόρμα και λειτουργικό σύστημα (Microsoft Windows, Mac, Linux, Unix) που υποστηρίζει την τεχνολογία της Java χωρίς να χρειάζεται επαναμεταγλώττιση για την επαναχρησιμοποίηση του σε κάποια άλλη συσκευή (write once, run anywhere). Οι εφαρμογές της γλώσσας αυτής τυπικά μεταγλωττίζονται σε Java bytecode που μπορεί να λειτουργήσει σε οποιοδήποτε Java virtual machine (JVM) που είναι γραμμένο για τη συσκευή που τρέχει, ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Η τεχνική αυτή παρά την πλειάδα των πλεονεκτημάτων που παρέχει στη γλώσσα δημιουργεί ένα σημαντικό πρόβλημα. Η εικονική μηχανή (VM) που τρέχει στη μνήμη του συστήματος κάνει τις εφαρμογές να αποκρίνονται πιο αργά. Μετά το 1997 το πρόβλημα ελαττώθηκε με την προσθήκη της σύνταξης just-in-time αλλά παραμένει ένα σημαντικό μειονέκτημα για την βελτίωση της γλώσσας.

Σήμερα είναι από τις δημοφιλέστερες γλώσσες προγραμματισμού, ιδίως για client-server εφαρμογές και ανήκει στην Oracle Corporation. Η εξαγορά της από την Sun Microsystems έγινε τον Απριλίου 2010 οπότε η Oracle απέκτησε όλα τα πνευματικά δικαιώματα και πατέντες.



Εικόνα 2-7 :
Λογότυπο Java

2.3.2 ASP και ASP.NET

Η ASP(Active Server Pages) γνωστή και ως ASP Classic σχεδιάστηκε και αναπτύχθηκε από τον Scott Guthrie της Microsoft στα μέσα του 1990. Αρχικά είχε κυκλοφορήσει ως επιπρόσθετη δυνατότητα(add-on), για Internet Information Services (IIS) μέσω των Windows NT 4.0 Option Pack το 1996 και ήταν η πρώτη server-side script (πλευρά διακομιστή) μηχανή της Microsoft για δημιουργία δυναμικών ιστοσελίδων. Αργότερα το 2000 συμπεριλήφθηκε σαν δωρεάν component στα Windows 2000 Server όπου ήταν και η τελευταία κυκλοφορία της.

Η νέα έκδοση της γλώσσας 1.0 με όνομα ASP.NET κυκλοφόρησε το 2002 αντικαθιστώντας την ASP Classic. Η ASP.NET είναι ανοιχτού κώδικα και δημιουργήθηκε για διαδικτυακό προγραμματισμό βελτιώνοντας κατά πολύ την αρχική της υλοποίηση. Με την καινούργια έκδοση της ASP.NET ο κώδικας βρίσκεται σε διαφορετικό αρχείο από το κώδικα της HTML παρέχοντας νέες δυνατότητες και καλύτερη ασφάλεια στη γλώσσα. Επίσης προστέθηκε η υποστήριξη Web forms που είναι τα κύρια μπλοκ πάνω στα οποία αναπτύσσονται οι εφαρμογές σε ASP.NET καθώς και η δυνατότητα στους προγραμματιστές να δημιουργήσουν με εύκολο τρόπο διαδικτυακές εφαρμογές και υπηρεσίες. Ένα βασικό πλεονέκτημά της, είναι ότι βασίστηκε πάνω σε Common Language Runtime (CLR), επιτρέποντας στους προγραμματιστές να γράψουν ASP.NET κώδικα χρησιμοποιώντας οποιαδήποτε υποστηριζόμενη .NET γλώσσα. Το μεγαλύτερο μειονέκτημα στη χρήση της ASP.NET είναι εξάρτηση της από την πλατφόρμα της Microsoft με αποτέλεσμα πολλοί Server στο διαδίκτυο να μην μπορούν να την χρησιμοποιήσουν.

Το κομμάτι .NET της ASP είναι ένα λογισμικό framework της Microsoft που απευθύνετε κυρίως σε λειτουργικά συστήματα Windows. Περιλαμβάνει μια μεγάλη κλάση βιβλιοθήκης (FCL) και παρέχει διαλειτουργικότητα στη γλώσσα δηλαδή την δυνατότητα ο κώδικας που έχει γραφτεί σε μια γλώσσα προγραμματισμού που υποστηρίζει .NET να γραφτεί σε διάφορες άλλες γλώσσες. Τα προγράμματα που έχουν γραφτεί σε .NET framework εκτελούνται σε ένα περιβάλλον λογισμικού γνωστό ως Common Language Runtime (CLR) όπως προαναφέραμε που είναι μια εφαρμογή εικονικής μηχανής, παρέχοντας υπηρεσίες ασφάλειας, διαχείρισης μνήμης και χειρισμό εξαιρέσεων (exception handling). Στην ουσία το FCL και το CLR μαζί αποτελούν το .NET Framework. Μέσω αυτής της αρχιτεκτονικής οι εφαρμογές των Windows είναι πιο αξιόπιστες βελτιώνοντας την ασφάλεια τους, οι web εφαρμογές και υπηρεσίες είναι απλούστερες και οι βιβλιοθήκες μπορούν να λειτουργήσουν σε πολλές γλώσσες προγραμματισμού.

Σήμερα η ASP.NET βρίσκεται στη beta έκδοση 5.0 επιτρέποντας στον κώδικά της μέσω της ανάπτυξης OmniSharp Cross platform .NET να λειτουργήσει σε συστήματα Mac και Linux.



ASP.NET

Εικόνα 2-8 : Λογότυπο ASP.NET

2.3.3 JSP

Η JSP(JavaServer Pages) είναι μία web-scripting τεχνολογία βασισμένη στη Java που επιτρέπει τον προγραμματισμό δυναμικά παραγόμενων ιστοσελίδων βασισμένων σε HTML, XML η οποιουδήποτε άλλου τύπου έγγραφου (Document Type) και λειτουργεί από την πλευρά του διακομιστή(server-side). Για πρώτη φορά κυκλοφόρησε το 1999 από την Sun Microsystems και αποτελεί τη βελτιωμένη έκδοση των JS(Java Servlets). Η JSP επιτρέπει στο κώδικα της Java και σε ορισμένες προκαθορισμένες ενέργειες να εμπλέκονται με το στατικό περιεχόμενων των ιστοσελίδων, με τα αποτελέσματα της σελίδας να μεταφράζονται και να εκτελούνται από την πλευρά του διακομιστή ο οποίος είναι υπεύθυνος για την αποστολή του επεξεργασμένου έγγραφου.



Εικόνα 2-9 : Λογότυπο JSP

Αρχιτεκτονικά η JSP μπορεί να θεωρηθεί ως ένα επίπεδο αφαιρετικότητας των Servlets της Java. Όταν λοιπόν ο χρήστης ζητά μια JSP σελίδα ο διακομιστής αφού τη λάβει την προωθεί στη μηχανή JSP η οποία εντοπίζει τον Java κώδικα στο αρχείο και τον μεταφράζει μέσω του JSP Compiler, δημιουργώντας ένα Servlet. Το Servlet στη συνέχεια παράγει την HTML σελίδα που θα επιστρέφει μέσω του διακομιστή στο χρήστη.

Η τεχνολογία JSP υποστηρίζει με έμπρακτο προγραμματιστικό τρόπο την φιλοσοφία της Sun Microsystems για δημιουργία τεχνολογιών write once, run anywhere, προσφέροντας ανεξαρτησία πλατφόρμας και web server. Μερικά από το ποιο σημαντικά πλεονεκτήματα που παρέχει η τεχνολογία αυτή συνοψίζονται παρακάτω.

- Ανεξαρτησία από τον διακομιστή
- Ανεξαρτησία λειτουργικού συστήματος, πλατφόρμας
- Δυνατότητα επαναχρησιμοποίησης κώδικα
- Επιτρέπει την χρήση Java κώδικα, και όλων των βιβλιοθηκών της
- Υποστηρίζει τις τεχνολογίες JavaBeans και Enterprise JavaBeans

2.3.4 Perl

Η τεχνολογία Perl (Practical Extraction and Report Language) είναι μια οικογένεια υψηλού επιπέδου, γενικής χρήσεως, διεργασιακή και δυναμική γλώσσα προγραμματισμού. Σχεδιάστηκε από τον Larry Wall το 1987 καθώς εργαζόταν ως προγραμματιστής στην εταιρία Unisys. Ο αρχικός της σκοπός ήταν να χρησιμοποιηθεί ως UNIX scripting γλώσσα για να μπορέσει να κάνει ευκολότερη την επεξεργασία εκθέσεων σε περιβάλλον UNIX. Κυκλοφόρησε το Δεκέμβριο του 1987 με την έκδοση 1.0 και τα επόμενα χρόνια υπήρξε ραγδαία εξέλιξη και ανάπτυξη λόγω των προγραμματιστικών δυνατοτήτων επεξεργασίας κειμένου που παρέχει.

Η γλώσσα Perl δανείζετε χαρακτηριστικά από άλλες γλώσσες προγραμματισμού όπως είναι η C, η shell scripting (sh), η sed και η AWK. Με την έκδοση 5 τον Οκτώβριο του 1994 και μέχρι τα τέλη του 1990 η Perl έγινε μία από τις πιο δημοφιλείς γλώσσες για την συγγραφή CGI scripts^[66] και αυτό οφείλετε στην αξεπέραστη ικανότητα της για string parsing και regular expression. Παρέχει ισχυρή επεξεργασία κειμένου χωρίς όρια στο μήκος των δεδομένων όπως υποστήριζαν πολλά σύγχρονα εργαλεία γραμμής εντολών UNIX, διευκολύνοντας το χειρισμό αρχείων κειμένου. Επιπρόσθετα με το CGI η Perl 5 χρησιμοποιείτε για τον προγραμματισμό γραφικών, για τη διαχείριση συστημάτων, για προγραμματισμό δικτύου, για βιοπληροφορική και για άλλες εφαρμογές.

Σήμερα η γλώσσα Perl βρίσκεται στην έκδοση 5.22 και υποστηρίζετε από πολλές πλατφόρμες. Παραμένει μια δημοφιλή γλώσσα για τη παραγωγή ιστοσελίδων παρέχοντας μια δυνατή μηχανή επεξεργασίας συμβολοσειρών και αχρείων. Είναι διαθέσιμη στο διαδίκτυο δωρεάν και διανέμεται με την αδεία GNU.

2.3.5 Ruby

Η Ruby είναι μια γενικής χρήσης, ανακλαστική, δυναμική, αντικειμενοστραφή γλώσσα προγραμματισμού η οποία είναι επηρεασμένη από την σύνταξη της Perl, της Smalltalk, της Eiffel, της Ada, και της Lisp. Σχεδιάστηκε και αναπτύχθηκε στα μέσα του 1990 από τον Yukihiro "Matz" Matsumoto στην Ιαπωνία και υποστηρίζει πολλαπλά πρότυπα προγραμματισμού συμπεριλαμβανομένων των δυνατοτήτων για αντικειμενοστραφή προγραμματισμό, συναρτησιακό



Εικόνα 2-10 :
Λογότυπο Perl



Ruby
A Programmer's Best Friend

Εικόνα 2-11 : Λογότυπο Ruby

προγραμματισμό, προστακτικό προγραμματισμό και ο ανακλαστικό (reflective) προγραμματισμό. Διαθέτει επίσης ένα δυναμικό τύπο συστήματος και ικανότητα για αυτόματη διαχείριση μνήμης.

Η αρχική ιδέα για τη Ruby προέκυψε στις 24 Φεβρουαρίου του 1993 από το Yukihiro Matsumoto που ήθελε να σχεδιάσει μια νέα γλώσσα που θα ήταν πιο ισχυρή από την Perl, πιο αντικειμενοστραφής από την Python και θα συνδυάζει με παραγωγικότητα και ευχάριστο τρόπο όλες τις αρχές μιας διεπαφής χρήστη. Η πρώτη δημόσια έκδοση της Ruby 0.95 ανακοινώθηκε σε κάποιες ομάδες της Ιαπωνίας στις 21 Δεκεμβρίου 1995 όπου μετά το διάστημα δυο ημερών δημοσιεύτηκαν άλλες τρεις εκδόσεις. Η έκδοση 1.0 κυκλοφόρησε στις 25 Δεκεμβρίου 1996. Μια σημαντική χρονιά για τη προώθηση και την επιτυχία της Ruby ήταν το 2005 όπου ο συνδυασμός της με το Ruby on Rails την έκανε πολύ γνωστή στο ευρύ κοινό. Το Ruby on Rails είναι ένα πλαίσιο ανάπτυξης Web εφαρμογών (web application framework) γραμμένο σε γλώσσα προγραμματισμού Ruby το οποίο προσφέρει την δυνατότητα για ταχεία ανάπτυξη λογισμικού (rapid application development). Το όνομα της Ruby αποφασίστηκε στη διάρκεια μιας ηλεκτρονικής συνομιλίας μεταξύ του Matsumoto και του Keiju Ishitsuka στις 24 Φεβρουαρίου του 1993. Ο Matsumoto αργότερα είχε δηλώσει ότι σημαντικό ρόλο στην επιλογή αυτού του ονόματος είχε συμβάλει ένα ρουμπίνι που είχε δοθεί σαν θρησκευτικό δώρο σε ένα συνάδελφο του.

Σήμερα η γλώσσα προγραμματισμού Ruby βρίσκεται στην έκδοση 2.2.2 όπου έχει εισάγει σημαντικές αλλαγές στο χειρισμό της μνήμης και στις δυνατότητες του αντικειμενοστραφή προγραμματισμού.

2.3.6 Python

Η Python είναι μια γενικής χρήσης, ανοιχτού κώδικα, υψηλού επιπέδου γλώσσα προγραμματισμού. Σχεδιάστηκε το 1989 από τον Guido van Rossum στο εθνικό ινστιτούτο έρευνας CWI(Centrum Wiskunde & Informatica) στην Ολλανδία. Αρχικά σχεδιάστηκε ως μία γλώσσα σεναρίων για το λειτουργικό σύστημα Amoeba, ικανή να διαχειριστεί εξαιρέσεις και κλήσεις συστήματος. Η αρχική πηγή έμπνευσης για την σχεδίαση της Python ήταν η γλώσσα ABC ενώ το όνομα της το πήρε από την αγαπημένη τηλεοπτική εκπομπή του Guido van Rossum, το ιπτάμενο τσίρκο των Monty Python's.

Η Python είναι μια αλληλεπιδραστική (interactive), διερμηνευτική (interpreted), αντικειμενοστραφής γλώσσα προγραμματισμού και χρησιμοποιεί μεταγλωττιστή για την δημιουργία του εκτελέσιμου κώδικα της. Βασικό κομμάτι της προγραμματιστικής της φιλοσοφίας είναι η αναγνωσιμότητα του κώδικά της, καθώς και η ευκολία χρήσης του συντακτικού της, παρέχοντας την δυνατότητα στους προγραμματιστές να προγραμματίσουν και να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα, κάτι το οποίο είναι αδύνατον για γλώσσες όπως η Java και η C. Λόγο της ευκολίας χρήσης του συντακτικού της η Python είναι ιδανική για σενάρια και rapid development. Χρησιμοποιεί ενότητες (modules), τάξεις (classes), εξαιρέσεις (exceptions) καθώς και πολύ υψηλού επιπέδου δυναμικούς τύπους δεδομένων. Οι διερμηνευτές (interpreters) της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα(Windows, Macintosh, Unix) και ο πηγαίος κώδικας (source code) της Python διατίθεται δωρεάν. Χρησιμοποιώντας εργαλεία τρίτων(third parties), όπως το Pyinstaller ή το Py2exe, ο κώδικας της Python μπορεί να πακεταριστεί σε εκτελέσιμα αρχεία τα οποία παρέχουν την αυτονομία σε Python λογισμικό να εκτελεστεί σε λειτουργικά συστήματα χωρίς να απαιτείτε εγκατάσταση του ερμηνευτή της Python.

Σήμερα η Python αναπτύσσεται ως ανοιχτού κώδικα λογισμικό όπου η διαχείριση και ανάπτυξη της γίνεται από το μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας της γλώσσας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Άξιο αναφοράς για το κυρός και τη φήμη της Python είναι ότι μία από τις μεγαλύτερες ανεπτυγμένες εταιρίες όλου του κόσμου η γνωστή σε όλους μας Google χρησιμοποιεί την Python από την αρχή του έργου της.



Εικόνα 2-12 : Λογότυπο Python

ΚΕΦΑΛΑΙΟ 3

Ανάλυση τεχνολογικών απαιτήσεων σε θεωρητικό επίπεδο

Στο κεφάλαιο αυτό, θα κάνουμε μια τεχνολογική ανάλυση απαιτήσεων, στο γνωστικό υπόβαθρο που απαιτείτε για να κατανοήσουμε τον τρόπο με τον οποίο έγινε η υλοποίηση του συστήματος για online δημοπρασίες και πωλήσεις. Η ανάλυση των απαιτήσεων αυτών άφορα το θεωρητικό κομμάτι που πρέπει να γνωρίζουμε για να μπορέσουμε να το εφαρμόσουμε με έμπρακτο τρόπο τις τεχνολογίες που επιλέξαμε.

Το σύστημα για online δημοπρασίες και πωλήσεις υλοποιείται μέσω ενός ιστότοπου που έχει την μορφή ενός ηλεκτρονικού καταστήματος. Ο επισκέπτης μπαίνοντας για πρώτη φορά στην κεντρική ιστοσελίδα θα πρέπει να κάνει εγγραφή νέου χρήστη και στη συνέχεια κάνοντας σύνδεση με το κωδικό και το μοναδικό όνομα του, θα έχει τη δυνατότητα να αναζητήσει προϊόντα η δημοπρασίες που τον ενδιαφέρουν όπως επίσης θα μπορεί να συμμετέχει σε διάφορες πωλήσεις προϊόντων μέσω δημοπρασιών που εξελίσσονται σε πραγματικό χρόνο. Επιπλέον το σύστημα υποστηρίζει έναν διαχειριστή ο οποίος έχει τη δυνατότητα να προσθέσει ένα νέο προϊόν, να δημιουργήσει μια νέα δημοπρασία, και να τυπώσει συγκεντρωτικές αναφορές που σχετίζονται με τις δημοπρασίες που έχουν λάβει τέλος.

Η ανάλυση των απαιτήσεων αυτών, για την ορθή λειτουργία του ηλεκτρονικού καταστήματος οδηγεί στην επιλογή και τον συνδυασμό των παρακάτω τεχνολογιών για την σχεδίαση και υλοποίηση του έργου.

3.1 Σύντομη περιγραφή τεχνολογικών απαιτήσεων

Για την υλοποίηση αυτής της πτυχιακής εργασίας συνδυαστήκαν οι παρακάτω τεχνολογίες:

- XML (Extensible Markup Language)
Η γλώσσα σήμανσης XML κρατά δομημένες σε έγγραφα όλες τις χρήσιμες πληροφορίες που είναι αναγκαίες για τη σωστή λειτουργία του ιστότοπου.
- XML schemas
Τεχνολογία μοντελοποίησης για την περιγραφή και τον καθορισμό κανόνων σε έγγραφα XML.
- XSL(Extensible Stylesheet Language)
Η γλώσσα XSL περιλαμβάνει την γλώσσα μετασχηματισμού XSLT η οποία χρησιμοποιήθηκε για τον καθορισμό του τρόπου εμφάνισης των εγγράφων XML.
- XPath
Η XPath είναι μια γλώσσα εκτέλεσης ερωτημάτων με σκοπό την επιλογή κόμβων(nodes) και χαρακτηριστικών από ένα έγγραφο XML.
- PHP(Hypertext Preprocessor)
Η γλώσσα προγραμματισμού PHP χρησιμοποιήθηκε για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο, παρέχοντας έναν αξιόπιστο μηχανισμό για την διαχείριση και επεξεργασία των XML εγγράφων.
- JavaScript
Η γλώσσα προγραμματισμού JavaScript χρησιμοποιήθηκε για τη παραγωγή δυναμικού περιεχομένου και την προσθήκη διαδραστικότητας στις ιστοσελίδες της εφαρμογής.
- AJAX(Asynchronous Javascript and XML)
Η χρήση της τεχνικής AJAX μας επιτρέπει να ενημερώνουμε ασύγχρονα, μέρη μιας ιστοσελίδας χωρίς να χρειάζεται να ανανεώσουμε ολόκληρη τη σελίδα.

- HTML(HyperText Markup Language)
Η γλώσσα σήμανσης υπερκειμένου HTML μας επιτρέπει τη δημιουργία και τη σωστή δόμηση ιστοσελίδων.
- CSS(Cascading Style Sheets)
Η γλώσσα φύλλων στυλ CSS μας επιτρέπει την μορφοποίηση και τον καθορισμό της εμφάνισης κάθε ιστοσελίδας.

3.2 Η Γλώσσά Σήμανσης XML

Η τεράστια εξάπλωση του παγκοσμίου ιστού και η ανάγκη για σύγχρονες τεχνολογικές τάσεις δημιουργείσαι μεγάλο όγκο πληροφορίας η οποία καθημερινά ανταλλάσσεται μέσω διαδικτυακών εφαρμογών και υπηρεσιών. Για τη διακίνηση του τεράστιου αυτού όγκου δεδομένων είναι αναγκαίο οι πηγές πληροφορίας να είναι εύκολα προσβάσιμες, ευέλικτες ως προς την μεταφορά τους και ανεξάρτητες οποιουδήποτε συστήματος και περιεχομένου. Για την επίλυση τέτοιων προβλημάτων αναπτύχθηκαν οι γλώσσες σήμανσης(markup language).

Μια γλώσσα σήμανσης δεν είναι τίποτα παραπάνω από ένα σύνολο εντολών γνώστες ως ετικέτες(tags) που επιτρέπουν την προσθήκη πληροφορίας στο περιεχόμενο μίας πηγής δεδομένων. Η πληροφορία αυτή περιγράφει το περιεχόμενο σε ένα έγγραφο με τέτοιο τρόπο ώστε συντακτικά να ξεχωρίζει από το υπόλοιπο κείμενο. Μια τέτοια γλώσσα που πληροί τις παραπάνω προϋποθέσεις είναι και η XML.



Εικόνα 3-1 : Γλώσσα Σήμανσης XML

3.2.1 Ιστορία της XML

Η γλώσσα XML αναπτύχθηκε το 1996 από το XML Working Group(γνωστό και ως SGML Editorial Review Board) του οργανισμού προτύπων W3C (World Wide Web Consortium) και συνδυάζει την ισχύ και την επεκτασιμότητα της SGML (Standard Generalized Markup Language). Η SGML δημιουργήθηκε στις αρχές της δεκαετίας του '80 και αποτελεί το διεθνές πρότυπο(ISO 8879:1986) γενικευμένης γλώσσας σήμανσης. Σχεδιάστηκε για να καθορίσει τους διάφορους τρόπους αναπαράστασης της πληροφορία, προσφέροντας ανεξαρτησία συστήματος και δυνατότητες δημιουργίας νέων γλωσσών σήμανσης οι οποίες είναι υποσύνολά της. Η SGML αποτέλεσε την έμπνευση για την ανάπτυξη της γλώσσας HTML το 1990 η οποία με την σειρά της έπαιξε σημαντικό ρολό στο σχεδιασμό της γλώσσας XML(Extensible Markup Language).

Αρχικά η XML σχεδιάστηκε για τη χρήση της SGML στο διαδίκτυο η οποία λόγω του προτύπου επικοινωνίας της και της πολυπλοκότητας της δεν μπορούσε να χρησιμοποιηθεί για ένα τέτοιο σκοπό. Επιπλέον ο σχεδιασμός της XML κατάφερε να ικανοποιήσει πολλαπλές ανάγκες στις οποίες η HTML αδυνατεί να ικανοποιήσει, παρέχοντας περισσότερη προσαρμοστικότητα στη δομή και το στυλ. Ουσιαστικά η γλώσσα σήμανσης XML αποτελεί υποσύνολο της SGML καλύπτοντας τις ατέλειες της HTML και απευθύνεται στη σήμανση εγγράφων που περιέχουν δομημένες πληροφορίες, αποδίδοντας σημασιολογική πληροφορία στα περιεχόμενα ενός εγγράφου.

3.2.2 Τι είναι η γλώσσά XML

Η γλώσσα σήμανσης XML έχει γίνει το επικρατέστερο μέσο για την αναπαράσταση και διακίνηση δεδομένων ανάμεσα σε εφαρμογές και συστήματα στον Παγκόσμιο Ιστό. Αποτελεί μια γλώσσα περιγραφής δεδομένων και ένα από τα σημαντικότερα πλεονεκτήματα της είναι η ευκολία στη χρήση και την κατανόηση της. Αρχικά χρησιμοποιήθηκε για την ανταλλαγή δεδομένων αλλά λόγω της επεκτασιμότητας της δεν περιοριστικέ μονό εκεί.

Η γλώσσα σήμανσης XML είναι ένα σύνολο εντολών, γνώστες ως ετικέτες που επιτρέπουν την προσθήκη πληροφορίας με δομημένο τρόπο στο περιεχόμενο μιας πηγής δεδομένων (έγγραφο). Το περιεχόμενο αυτό μπορεί να είναι κείμενο, μια φωτογραφία, ένα βίντεο ή οποιαδήποτε άλλης μορφής πληροφορίας. Οι ετικέτες δεν είναι προκαθορισμένες και οι χρήστες έχουν τη δυνατότητα να καθορίσουν νέες ετικέτες με βάση τις ανάγκες τους. Με αυτό τον τρόπο μέσω της γλώσσας XML ένας χρήστης μπορεί να δημιουργήσει μια νέα γλωσσά σήμανσης φτιάχνοντας ένα νέο σύνολο από ετικέτες. Ο απεριόριστος αριθμός γλωσσών σήμανσης που μπορούν να παραχθούν μέσω της XML την καθιστά ως μεταγλώσσα (metalanguage).

Παρακάτω αναφέρονται συνοπτικά μερικά από τα πλεονέκτημα της XML :

- Ανεξαρτησία συστήματος και περιεχομένου
- Πολλαπλοί τρόποι εμφανίσεις των δεδομένων
- Δομημένο κείμενο
- Η δημιουργία XML εγγράφων μπορεί να γίνει με ένα απλό επεξεργαστή κειμένου
- Τα XML έγγραφα είναι ευανάγνωστα
- Δομή καθορισμένη από το χρήστη
- Ανάπτυξη ευέλικτων εφαρμογών
- Δυνατότητες εξέλιξης
- Δυνατότητα μορφοποίησης εμφάνισης
- Ευκολία στη χρήση χωρίς να χρειάζεται ιδιαίτερες γνώσεις
- Υποστηρίζετε από ένα εύρη φάσμα εφαρμογών
- Τα XML έγγραφα διαβάζονται από οποιοδήποτε φυλλομετρητή

3.2.3 Ευρεία χρήση της XML

Η ευρεία χρήση της XML τα τελευταία χρόνια στο παγκόσμιο ιστό επέφερε πολύ σημαντικές αλλαγές στον τρόπο με τον οποίο ανταλλάσσουν τα δεδομένα οι εφαρμογές μεταξύ τους, επιλύοντας ένα μεγάλο πρόβλημα ασυμβατότητας διαφορετικών συστημάτων. Οι εφαρμογές αυτές εξυπηρετούν καθημερινά μέσω υπηρεσιών διαδικτύου πολλές ανάγκες των ανθρώπων παρέχοντας ευκολία στην καθημερινή ζωή. Οι ανάγκες αυτές ποικίλουν και για αυτό η εξέλιξη του παγκόσμιου ιστού στην ζωή του ανθρώπου εξαρτάτε άμεσα από αυτό.

Η γλώσσα σήμανσης XML ήρθε να συμβάλει έμμεσα στην κάλυψη των αναγκών αυτών και να υποστηρίξει τον παγκόσμιο ιστό ως ένα πολύ σημαντικό εργαλείο στη πορεία της εξέλιξης του ανθρώπου. Συνοπτικά η XML έχει ευρεία εξέλιξη στους παρακάτω τομείς :

- Διαθέσιμα Επιχειρηματικά δεδομένα (πωλήσεις, τιμολόγια, παραγγελίες)
- Οικονομικά δεδομένα (οικονομικές και λογιστικές εφαρμογές)
- Στοιχεία από βάσεις δεδομένων (όλα τα σύγχρονα RDBMS υποστηρίζουν εξαγωγή και εισαγωγή από XML)
- Human Resources XML (HR-XML)

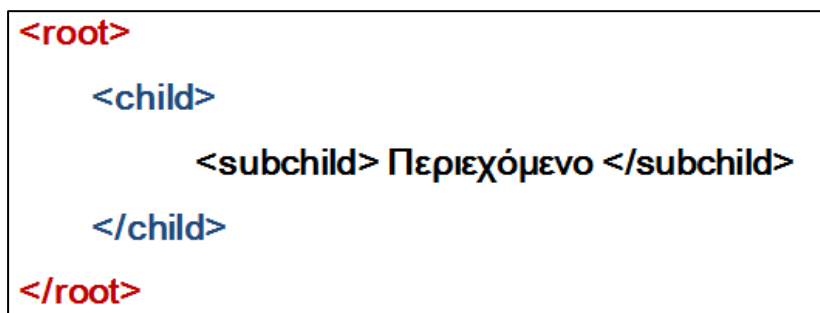
3.2.4 Σύνταξη και δομή της XML

Όπως προαναφέραμε η γλώσσα σήμανσης XML είναι ένα σύνολο εντολών που καθορίζονται από το χρήστη για τη δημιουργία ετικετών που περιγράφουν τα δεδομένα ενός εγγράφου. Τα XML έγγραφα όπως ονομάζονται είναι αυστηρά δομημένα και έχουν δενδρική δομή (βλ. παρακάτω) στην εικονική τους αποτύπωση. Τα αρχεία XML που δημιουργούνται από ένα χρήστη χρησιμοποιούν Unicode κωδικοποίηση χαρακτήρων (UTF-8, UTF-16, ISO 8859-1 κ.α.) προσφέροντας έτσι την δυνατότητα υπάρξεις χαρακτήρων από οποιοδήποτε αλφάβητο.

Παρακάτω αναφέρονται τα βασικά στοιχεία που συνθέτουν τη σωστή δομή ενός XML έγγραφου :

- Πρόλογος (Prolog)
- Στοιχείο Ρίζα (Root element)
- Στοιχείο (Element)
- Σήμανση (<, >)
- Ετικέτες (tags)
- Χαρακτηριστικό (Attribute)

Παράδειγμα βασικής δομής ενός XML έγγραφου :



Εικόνα 3-2 : Βασική δομή XML εγγράφου

Παράδειγμα δομής ενός XML έγγραφου για ένα πελάτη ενός ηλεκτρονικού καταστήματος :



Εικόνα 3-3 : Παράδειγμα δομής XML εγγράφου ενός πελάτη

Όπως βλέπουμε στο παράδειγμα της εικόνας 3-3 ένα XML έγγραφο ξεκινάει με τον πρόλογο που αναφέρετε στη δήλωση της XML έκδοσης (βλ. παρακάτω). Μετά το πρόλογο ακολουθεί το στοιχείο ρίζα με την ετικέτα ανοίγματος <customer> η οποία με τη σειρά της εμφωλεύει μέσα της κάποια αλλά στοιχεία. Τα στοιχεία αυτά έχουν μια ετικέτα ανοίγματος <user_id>, <name>, <surname> κ.ο.κ. και μια ετικέτα κλεισίματος </user_id>, </name>, </surname> κ.ο.κ. Προσοχή η ετικέτα κλεισίματος </customer> του στοιχείου ρίζα <customer> κλείνει στο τέλος του XML έγγραφου δηλώνοντας έτσι ότι μέσα της περικλείει αλλά στοιχεία. Τα ονόματα για κάθε ετικέτα τα ορίζει ο χρήστης και για να είναι ορθή η σύνταξη ενός XML εγγράφου κάθε ετικέτα ανοίγματος πρέπει να έχει και μια ετικέτα κλεισίματος με το ίδιο όνομα. Όπως βλέπουμε και στο παράδειγμα της εικόνας 3-3 μια ετικέτα έχει μια σήμανση ανοίγματος "<" και μια σήμανση κλεισίματος ">" για να είναι ολοκληρωμένη συντακτικά. Ένα στοιχείο με περιεχόμενο όπως είναι το <name> ονομάζετε και text element. Ένα στοιχείο χωρίς περιεχόμενο όπως είναι το <address></address> ονομάζετε κενό στοιχείο και για λογούς συντομογραφίας θα μπορούσε να γραφτεί <address/> χωρίς την συγγραφή ολοκλήρου του συντακτικού <address></address>. Σημαντική παρατήρηση στο παράδειγμα της εικόνας 3-3 είναι ότι η ετικέτα της δήλωσης XML δεν ακολουθείται από ετικέτα κλεισίματος σύμφωνα με τους κανόνες σύνταξης της XML. Μόνο τα στοιχεία που περιγράφουν τη δομή του XML έγγραφου έχουν ετικέτα κλεισίματος.

Για να μπορέσουμε να καταλάβουμε καλύτερα τους βασικούς κανόνες σύνταξης και δομής ενός σωστά γραμμένου XML έγγραφου, πρέπει να κατανοήσουμε τις έννοιες ετικέτα, στοιχείο ρίζα,

στοιχείο, σήμανση και πρόλογος. Αν κατανοήσουμε αυτά τα 5 βασικά στοιχεία τότε μπορούμε να αναγνωρίσουμε σε προσωπικό επίπεδο το πόσο σημασιολογική αξία δίνει η γλώσσα XML στα δεδομένα ενός εγγράφου, οδηγώντας πολύ εύκολα και γρήγορα τον χρήστη να κατασκευάσει με οργανωτική δομή ένα περιγραφικό ως προς το περιεχόμενο του XML έγγραφο.

Στα επόμενα υποκεφάλαια θα γίνει μια πιο λεπτομερής περιγραφή αυτών των 5 βασικών στοιχείων που συνθέτουν ένα XML έγγραφο, για την περαιτέρω κατανόηση και σαφήνεια τους.

3.2.5 Πρόλογος (Prolog)

Σύμφωνα με τους κανόνες του πρότυπου W3C (World Wide Web Consortium) στην πρώτη γραμμή κάθε XML εγγράφου πρέπει να βρίσκετε η δήλωση (declaration) όπως αλλιώς ονομάζετε της XML. Η δήλωση αυτή, είναι στην ουσία μία εντολή προς τον αναλυτή κειμένου (parser βλ. παρακάτω) και έχει σκοπό να υποδείξει τη φύση του εγγράφου. Με αυτό τον τρόπο διαβάζοντας ο αναλυτής την πρώτη γραμμή αναγνωρίζει ότι το έγγραφο είναι τύπου XML.

Όπως βλέπουμε και στο παράδειγμα της εικόνας 3-3 μια δήλωση XML χαρακτηρίζετε από ένα βασικό χαρακτηριστικό και ένα προαιρετικό.

1. Version (έκδοση). Βασικός όρος σε μια δήλωση XML
Παράδειγμα : `version="1.0"`
2. Encoding (κωδικοποίηση). Προαιρετικός όρος
Παράδειγμα : `encoding="utf-8"`

Το χαρακτηριστικό version πρέπει να υπάρχει πάντα μέσα σε μια δήλωση και περιγράφει την έκδοση της XML στο εκάστοτε XML έγγραφο και μέχρι και σήμερα η μια και μοναδική έκδοση είναι η "1.0". Το χαρακτηριστικό encoding χρησιμοποιείτε για να δηλώσουμε σε ένα αναλυτή κειμένου την κωδικοποίηση των χαρακτήρων σε ένα έγγραφο XML. Το encoding είναι μια προαιρετική αναφορά στην δήλωση XML, παρόλα αυτά οι συγγραφείς που θέλουν να χρησιμοποιήσουν χαρακτήρα κωδικοποίησης εκτός του US-ASCII πρέπει να το συμπεριλάβουν μέσα στην δήλωση τους. Άλλο ένα χαρακτηριστικό που είναι προαιρετικό σε μια δήλωση είναι το standalone (π.χ. `standalone="yes"`) και χρησιμοποιείτε όταν ένα XML έγγραφο εξαρτάτε από ένα άλλο XML έγγραφο. Οι τιμές που μπορεί να πάρει ο όρος αυτός είναι "yes" και "no". Στην περίπτωση που η τιμή είναι "yes" το έγγραφο XML στηρίζετε σε άλλο εξωτερικό έγγραφο το οποίο ονομάζετε DTD.

Μια ακόμα σημαντική διευκρίνιση κατά την συγγραφή μιας XML δήλωσης είναι η δομή σύνταξης που ακολουθεί. Πριν κάνουμε αναφορά στους ορούς για τους οποίους μιλήσαμε στην προηγούμενη παράγραφο μια δήλωση πρέπει να ξεκινά με το χαρακτηριστικό "<?" και να τελειώνει με το χαρακτηριστικό "?>" μετά την ορισμό του όρου version, και των προαιρετικών encoding, standalone(π.χ. `<?xml version="1.0" encoding="utf-8" standalone="yes"?>`). Το χαρακτηριστικά "<?" και "?>" αποτελούν εντολές προς τον αναλυτή, για το πώς θα αποκωδικοποιήσει την δομή του XML εγγράφου ενώ αντίστοιχα το αλφαριθμητικό (string) "xml" μέσα στην έκφραση της δήλωσης εκφράζει τη φύση του εγγράφου. Η λέξη XML λόγω της ύπαρξης της μέσα στην δήλωση δεν μπορεί να χρησιμοποιηθεί πουθενά μέσα σε ένα έγγραφο XML ούτε να χρησιμοποιηθεί σαν χαρακτηριστικό(attribute).

3.2.6 Στοιχείο ρίζα(Root element)

Το στοιχείο ρίζα είναι ένα βασικό στοιχείο μέσα σε ένα XML έγγραφο το οποίο δίνει σημασιολογική και φυσική υπόσταση στα δεδομένα που περικλείει και ως εκ τούτου περιγράφει. Όπως βλέπουμε και στο παράδειγμα της εικόνας 3-3 το στοιχείο ρίζα <customer> περιλαμβάνει όλα τα υπόλοιπα στοιχεία <user_id>, <name>, <surname> κ.ο.κ., περιγράφοντας με φυσικό τρόπο ότι ένας πελάτης ενός ηλεκτρονικού καταστήματος έχει όνομα, επώνυμο, διεύθυνση κ.τ.λ. και μια ταυτότητα(id) που τον κάνει μοναδικό στο κόσμο.

Για να είναι ορθή συντακτικά η δομή ενός XML εγγράφου το στοιχείο ρίζα πρέπει να είναι μοναδικό (έχει μοναδικό όνομα), να δηλώνετε πάντα μετά το πρόλογο της XML και δομικά να

βρίσκετε στην κορυφή της ιεραρχίας, περικλείοντας όλα τα υπόλοιπα στοιχεία. Όλα τα στοιχεία μέσα σε ένα XML έγγραφο συμπεριλαμβανομένου και του στοιχείου ρίζα ακολουθούν μια συγκεκριμένη σύνταξη διαχωρίζοντας τα από την σύνταξη της δήλωσης. Το όνομα κάθε στοιχείου ξεκινάει με την σήμανση “<”, ακολουθεί το όνομα του και τελειώνει με την σήμανση “>”(ετικέτα ανοίγματος). Συγκεκριμένα, σύμφωνα με το παράδειγμα της εικόνας 3-3 το στοιχείο ρίζα ακολουθεί την εξής σύνταξη και δομή :

1. Ξεκινάμε με τον σήμανση <
2. Δίνουμε το όνομα ετικέτας (όνομα στοιχείου ρίζας) <customer
3. Κλείνουμε την ετικέτα με τη σήμανση >
4. Αποτέλεσμα <customer>

Σε αυτό το σημείο έχουμε δημιουργήσει το στοιχείο ρίζα <customer> σύμφωνα με τους κανόνες σύνταξης ενός XML εγγράφου. Ακολουθώντας τους κανόνες δομής της XML, αφού δημιουργήσουμε όλα τα στοιχεία που θα περιλαμβάνει το στοιχείο ρίζα συνεχίζουμε με τα εξής βήματα:

5. Συντάσσουμε την ετικέτα κλεισίματος του στοιχείου ρίζα
6. Ξεκινάμε με την σήμανση </
7. Γράφουμε τη σήμανση / που δηλώνει το σκοπό κλεισίματος της ετικέτας </
8. Γράφουμε το όνομα του στοιχείου ρίζας </customer
9. Η ετικέτα κλεισίματος ολοκληρώνεται με τη σήμανση >
10. Αποτέλεσμα βλ. παράδειγμα εικόνα 3-3

3.2.7 Στοιχείο (Element)

Το στοιχείο η όπως αλλιώς ονομάζετε το στοιχείο δεδομένων μπορεί να περιλαμβάνει αλλά στοιχεία ή μπορεί να έχει μια τιμή. Στο παράδειγμα της εικόνας 3-3, στοιχείο θεωρείτε οτιδήποτε βρίσκετε μέσα στη ρίζα όπως είναι <user_id>, <name>, <surname> κ.ο.κ.. Η τιμή που έχει ένα στοιχείο για παράδειγμα το <user_id> έχει τιμή 4562 δεν θεωρείτε στοιχείο αλλά περιεχόμενο του στοιχείου.

Για τη συγγραφή ενός στοιχείου μέσα σε ένα έγγραφο XML ακολουθούμε την εξής σύνταξη και δομή:

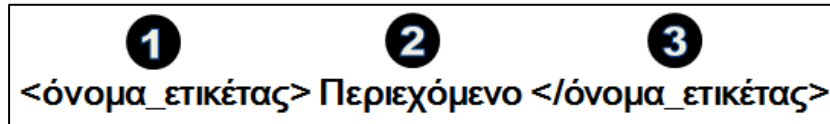
1. Ξεκινάμε με τον σήμανση <
2. Δίνουμε το όνομα ετικέτας (όνομα στοιχείου) <όνομα_στοιχείου
3. Κλείνουμε την ετικέτα με τη σήμανση >.
4. Αποτέλεσμα <όνομα_στοιχείου>
5. Το στοιχείο αυτό μπορεί:
 - Να περιλαμβάνει αλλά στοιχεία (ακολουθούμε τα βήματα 1 έως 10)
 - Να έχει μια τιμή (βλ. παρακάτω)
6. Συντάσσουμε την ετικέτα κλεισίματος του στοιχείου
7. Ξεκινάμε με την σήμανση </
8. Γράφουμε τη σήμανση / που δηλώνει το σκοπό κλεισίματος της ετικέτας </
9. Συνεχίζουμε γράφοντας το όνομα του στοιχείου από το βήμα 2 </όνομα_στοιχείου
10. Η ετικέτα κλεισίματος ολοκληρώνεται με τη σήμανση >

Ορισμός σύνταξης: <όνομα_στοιχείου>Περιεχόμενο ή άλλα στοιχεία</όνομα_στοιχείου>

Το στοιχείο σαν έννοια μέσα σε ένα έγγραφο XML παίζει σημαντικό ρόλο, τόσο στη περιγραφή όσο και την σημασιολογική αξία που δίνει στο περιεχόμενο που περικλείει. Το όνομα ενός στοιχείου επιλέγεται από το χρήστη με βάση τις ανάγκες που θέλει να εξυπηρετεί το εκάστοτε έγγραφο XML. Κατά την διαδικασία ονοματοδοσία πρέπει να είμαστε προσεχτικοί, γιατί παρότι φαίνεται εύκολη υπόθεση, αποτελεί τη βάση για να μπορέσουμε να χτίσουμε οργανωμένα τα δεδομένα μας μέσα σε ένα έγγραφο XML.

3.2.8 Ετικέτες (Tags) και Σήμανση (Markup)

Κοιτάζοντας το παράδειγμα της εικόνας 3-3 ετικέτα θεωρείτε το <customer>, το <customer/>, το <user_id>, το <user_id/> κ.ο.κ. Ο ρόλος της ετικέτας μέσα σε ένα έγγραφο XML είναι πολύ σημαντικός αφού διαχωρίζει το υπόλοιπο κείμενο του εγγράφου από τα στοιχεία. Όπως μπορούμε να καταλάβουμε τα στοιχεία XML έχουν άμεση σχέση με τις ετικέτες και αυτό γιατί οι ετικέτες μαζί με τη σήμανση, είναι τα δομικά συστατικά που συνθέτουν τα στοιχεία ενός εγγράφου. Ο συνδυασμός μιας ετικέτας μαζί με την σήμανση ακολουθεί την εξής σύνταξη και δομή:



Εικόνα 3-4 : Σύνταξη ετικέτας XML

Όπου:

1. Ετικέτα ανοίγματος

Η ετικέτα ανοίγματος αποτελείται από την σήμανση “<”, “>” και το όνομα που θα επιλέξει ο χρήστης με βάση τις ανάγκες που θέλει να υλοποιεί το XML έγγραφο που δημιουργεί.

2. Περιεχόμενο

Το περιεχόμενο μπορεί να είναι ένα κείμενο, άλλα δεδομένα (π.χ. εικόνα), άλλες ετικέτες που είναι εμφωλευμένες ή και ο συνδυασμός αυτών (βλ. παρακάτω).

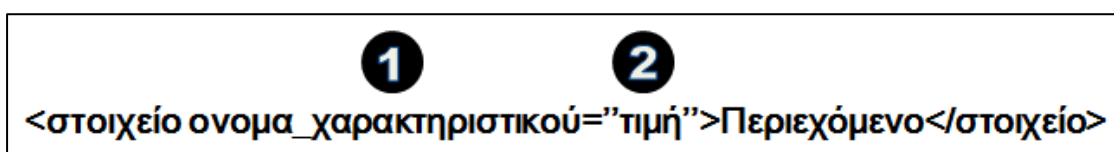
3. Ετικέτα κλεισίματος

Η ετικέτα κλεισίματος διαφοροποιείται από την ετικέτα ανοίγματος με την παρουσία του χαρακτήρα “/”. Ο χαρακτήρας αυτός είναι μια σήμανση που δηλώνει το τέλος της ετικέτας. Δομικά μέσα σε ένα έγγραφο XML μια ετικέτα τέλους μπαίνει πάντα μετά την ετικέτα ανοίγματος και συντακτικά πρέπει να έχει το ίδιο όνομα με την ετικέτα ανοίγματος που συνδυάζετε.

Ακολουθώντας τα παραπάνω βήματα η συγγραφή ετικετών χωρίς συντακτικά λάθη και με σωστή δομή και περιγραφική υπόσταση αποτελεί μια εύκολη υπόθεση.

3.2.9 Χαρακτηριστικό (Attribute)

Το χαρακτηριστικό αποτελεί μια επιπλέον πληροφορία σε ένα στοιχείο δίνοντας του σημασιολογικά μία ιδιότητα. Οι κανόνες σύνταξης ενός χαρακτηριστικού δεν αποκλίνουν από τους βασικούς κανόνες σύνταξης που αναλύσαμε στις προηγούμενες ενότητες του τρίτου κεφαλαίου, με την διαφορά ότι δομικά ένα χαρακτηριστικό είναι εμφωλευμένο μέσα στο στοιχείο που χαρακτηρίζει. Για την συγγραφή ενός χαρακτηριστικού μέσα σε ένα στοιχείο ακολουθούμε την εξής σύνταξη:



Εικόνα 3-5 : Σύνταξη χαρακτηριστικού σε ένα XML έγγραφο

Όπου :

1. Όνομα χαρακτηριστικού

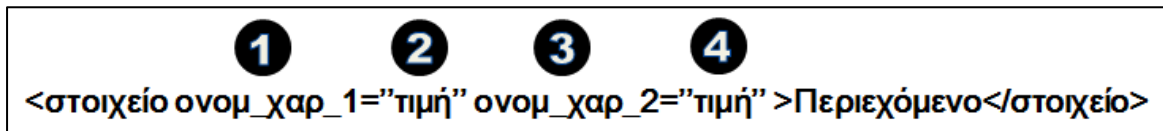
Το όνομα ενός χαρακτηριστικού το επιλέγει ο χρήστης και πρέπει να τηρεί τα εξής κριτήρια για να είναι συντακτικά σωστό:

- Το όνομα του είναι μοναδικό και να μην εμφανίζεται μέσα στο ίδιο στοιχείο παραπάνω από μια φορά.
- Πρέπει να ξεκινάει, ή με ένα γράμμα, ή με ένα αριθμό ή με το χαρακτήρα “_”.
- Μπορεί να περιέχει γράμματα, αριθμούς και συγκεκριμένους χαρακτήρες όπως “_”, “-” και “.”

2. Τιμή χαρακτηριστικού

Η τιμή ενός χαρακτηριστικού συντακτικά περικλείεται με μονά ή διπλά εισαγωγικά και μπορεί να περιλαμβάνει οποιαδήποτε δεδομένα χαρακτήρων. Η τιμή αποδίδεται μετά το χαρακτήρα της ισότητας “=” ξεχωρίζοντας με αυτό τον τρόπο το όνομα από την τιμή του χαρακτηριστικού.

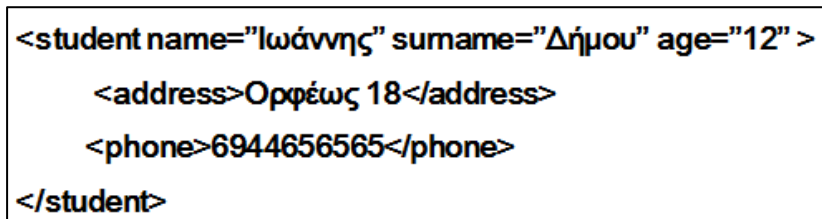
Μέσα σε ένα στοιχείο μπορούμε να έχουμε παραπάνω από ένα χαρακτηριστικά. Συντακτικά ακολουθούμε την εξής δομή:



Εικόνα 3-6 : Σύνταξη χαρακτηριστικών σε ένα XML έγγραφο

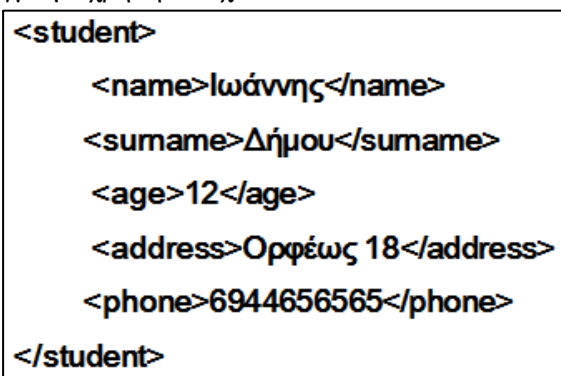
Για κάποιους προγραμματιστές και συγγραφείς XML εγγράφων το χαρακτηριστικό αποτελεί βασική παράμετρος της γλώσσας αλλά στην πραγματικότητα δεν είναι και τόσο βασική και μπορούμε να το αποφύγουμε μέσω χρήσης στοιχείων αντί χαρακτηριστικών. Τα δύο επόμενα παραδείγματα αποτυπώνουν με διαφορετικό τρόπο το ίδιο νόημα:

Παράδειγμα με χρήση χαρακτηριστικών:



Εικόνα 3-7 : Παράδειγμα χρήσης χαρακτηριστικών XML

Παράδειγμα με χρήση στοιχείων:

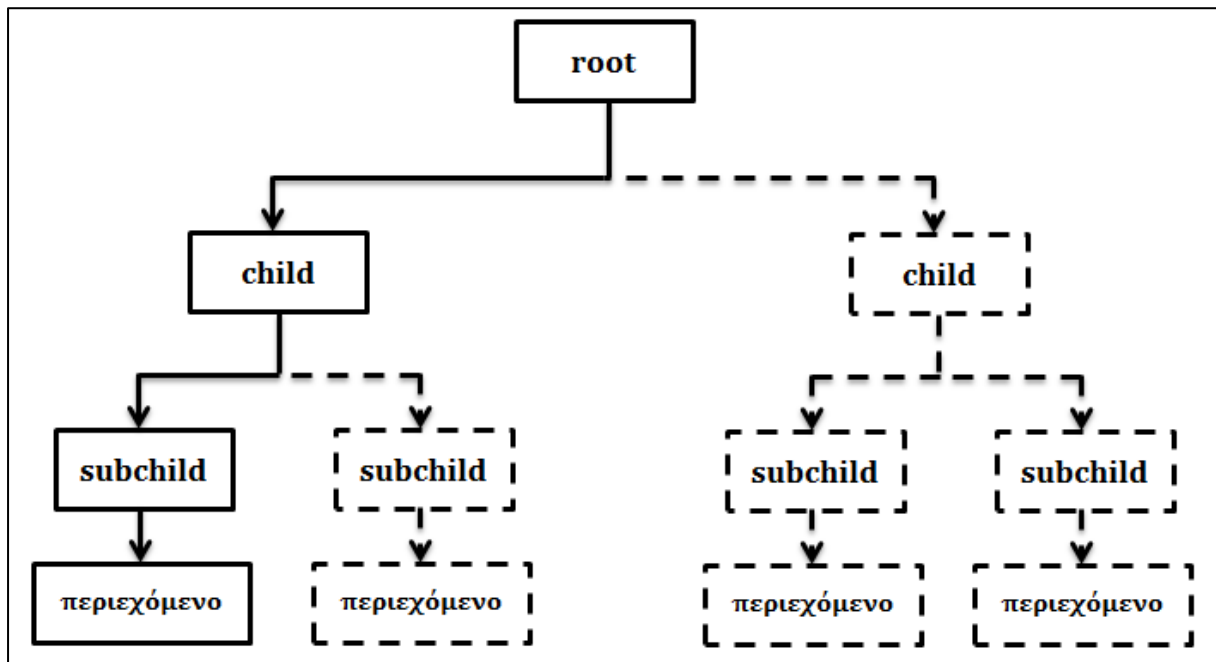


Εικόνα 3-8 : Παράδειγμα χρήσης στοιχείων XML

Η χρήση στοιχείων έναντι χαρακτηριστικών έγκειται στις ανάγκες του εκάστοτε χρήστη και πολλές φορές αυτό εξαρτάτε από την εφαρμογή τόσο σε θέματα λειτουργικότητας όσο και σε θέματα ανάπτυξης. Για παράδειγμα αν είχαμε ένα μεγάλο κείμενο δεν θα ήταν και πολύ λειτουργικό η οπτικά ωραίο από προγραμματιστική άποψη να το τοποθετήσουμε σε ένα χαρακτηριστικό.

3.2.10 Δενδρική δομή και αναπαράσταση ενός XML εγγράφου

Όπως προαναφέραμε η ιεραρχία των στοιχείων δομικά μέσα σε ένα έγγραφο XML θυμίζει δέντρο. Τα στοιχεία αναπαριστούν τα κλαδιά του δέντρου και το περιεχόμενο των στοιχείων τα φύλλα του. Σύμφωνα με την εικόνα 3-2 η δενδρική αναπαράσταση ενός τέτοιου εγγράφου θα ήταν η εξής:



Εικόνα 3-9 : Δενδρική αναπαράσταση βασικής δομής XML εγγράφου

Όπου :

- Οι συνεχόμενες γραμμές αποτυπώνουν την βασική δομή, σύμφωνα με την εικόνα 3-2 της παραγράφου 3.2.4.
- Οι διακεκομμένες γραμμές αποτυπώνουν την περαιτέρω ανάπτυξη ενός τέτοιου εγγράφου, με μία διακλάδωση στην οποία τα στοιχεία περιέχουν αλλά στοιχεία δημιουργώντας ένα μεγαλύτερο δέντρο με περισσότερα κλαδιά

Σε κάθε XML έγγραφο τα στοιχεία που το αποτελούν έχουν την ιεραρχική σχέση πατέρα παιδιού, με το στοιχείο ριζά είναι ο πατέρας όλων των άλλων στοιχείων-παιδιών. Τα στοιχεία αυτά ονομάζονται επίσης κόμβοι(nodes) ενώ τα περιεχόμενα στο τέλος των στοιχείων ονομάζονται επίσης φύλλα(leaves).

Παρατηρώντας το σχήμα της εικόνας 3-9 μπορούμε να δούμε πιο άμεσα την σχέση των στοιχείων καθώς και την ανάπτυξη τους. Ο πατέρας έχει κάποια παιδιά που με την σειρά τους αυτά έχουν κάποια άλλα παιδιά και η διαδικασία συνεχίζεται μέχρι να φτάσουμε σε κόμβο χωρίς παιδιά. Κάθε κόμβος σε ένα δέντρο μπορεί να θεωρηθεί ως πατέρας σε ένα μικρότερο υπό δέντρο. Το στοιχείο "child" στην εικόνα 3-9 είναι ο πατέρας των άλλων δύο στοιχείων "subchild". Τα στοιχεία που μοιράζονται τον ίδιο πατέρα ονομάζονται αδέρφια(siblings).

3.2.11 Προκαθορισμένες Οντότητες

Σύμφωνα με τους κανόνες σύνταξης της XML μερικοί ειδικοί χαρακτήρες δεν μπορούν να χρησιμοποιηθούν μέσα σε ένα έγγραφο γιατί είναι δεσμευμένοι από την XML. Για παράδειγμα οι αγκύλες “< >” που χρησιμοποιούνται για να ορίσουν τις ετικέτες των στοιχείων δεν μπορούν να χρησιμοποιηθούν μέσα σε ένα XML έγγραφο για κάποιο άλλο λόγο. Για να μπορέσει να λυθεί το πρόβλημα που προκύπτει η XML χρησιμοποιεί πέντε προκαθορισμένες οντότητες.

1. Ο χαρακτήρας > αποτυπώνετε με το συμβολισμό >
2. Ο χαρακτήρας < αποτυπώνετε με το συμβολισμό <
3. Ο χαρακτήρας ‘ αποτυπώνετε με το συμβολισμό '
4. Ο χαρακτήρας “ αποτυπώνετε με το συμβολισμό "
5. Ο χαρακτήρας & αποτυπώνετε με το συμβολισμό &

Στο παράδειγμα που ακολουθεί φαίνετε η λανθασμένη και η ορθή χρήση των προκαθορισμένων οντοτήτων:

- Λανθασμένη χρήση

```
<speed_limit>ταχύτητα < 220 </speed_limit>
```

Εικόνα 3-10 : Λανθασμένη χρήση προκαθορισμένων οντοτήτων της XML

- Ορθή χρήση

```
<speed_limit>ταχύτητα &lt; 220 </speed_limit>
```

Εικόνα 3-11 : Ορθή χρήση προκαθορισμένων οντοτήτων της XML

3.2.12 Περιεχόμενα στοιχείου

Το περιεχόμενο ενός στοιχείου σε ένα XML έγγραφο μπορεί να είναι ένα απλό κείμενο, μια εικόνα, ένα βίντεο ή οτιδήποτε άλλο βρίσκεται σε κωδικοποιημένη ή μη ψηφιακή μορφή. Όταν το περιεχόμενο αυτό έχει πολλούς ειδικούς χαρακτήρες αντί να χρησιμοποιήσουμε την χρήση των προκαθορισμένων οντοτήτων που αναφέραμε στην παράγραφο 3.2.11, είναι πιο λειτουργικό και γρήγορο να χρησιμοποιούμε τα τμήματα CDATA. Ο όρος CDATA(συντομογραφία του character data) μας δίνει την ευκολία να χρησιμοποιήσουμε μεγάλα κομμάτια κειμένου που περιέχουν ειδικούς χαρακτήρες χωρίς να χρειάζεται να κάνουμε αναφορά προκαθορισμένης οντότητας σε κάθε ένα ξεχωριστά ειδικό χαρακτήρα.

Όταν ο XML αναλυτής διαβάζει ένα τμήμα CDATA ενημερώνετε ότι το κομμάτι αυτό δεν περιέχει σήμανση και το διαχειρίζεται σαν ένα απλό κείμενο. Η σύνταξη τμημάτων CDATA ακολουθεί την εξής δομή:

- <![CDATA[κείμενο με προκαθορισμένους χαρακτήρες]]>



Εικόνα 3-12 : Σύνταξη τμήματος CDATA δεδομένων

Όπου:

1. Η σύνταξη τμημάτων CDATA ξεκινά πάντα με το χαρακτηριστικό `'<![CDATA[`' το οποίο δηλώνει στον αναλυτή την φύση των δεδομένων που ακολουθούν
2. Η ενότητα δύο αφορά το κείμενο το οποίο περιέχει χαρακτήρες σήμανσης οι οποίοι αγνοούνται από τον αναλυτή λόγω του ειδικού χαρακτηριστικού `'<![CDATA[`'.
3. Η τρίτη ενότητα αναφέρετε στον οριοθέτη τέλους `']]>`' ο οποίος δηλώνει το τέλος των δεδομένων που περιέχουν προκαθορισμένους χαρακτήρες και πρέπει να παραβλεφθούν από τον αναλυτή.

Μία σημαντική διευκρίνιση για τα τμήματα CDATA είναι ότι στο περιεχόμενο δεν μπορεί να μπει ο οριοθέτης τέλους `']]>`'. Για να είναι ορθή η σύνταξη πρέπει να χρησιμοποιήσουμε τους προκαθορισμένους χαρακτήρες `']]>`'.

3.2.13 Χώροι ονοματοδοσίας(Namespaces)

Ένα πολύ σημαντικό πλεονέκτημα της γλώσσας XML είναι ότι επιτρέπει στο χρήστη να ορίσει αυτός τις ετικέτες, επιλέγοντας την ονοματοδοσία των στοιχείων και των χαρακτηριστικών με βάση τις ανάγκες που εξυπηρετεί το εκάστοτε XML έγγραφο. Τι γίνεται όμως στην περίπτωση που θέλουμε να συνδυάσουμε δύο ή περισσότερα έγγραφα XML που μπορεί να περιέχουν στοιχεία η χαρακτηριστικά με τα ίδια ονόματα; Πως θα αποφύγουμε τις συγκρούσεις των διπλότυπων στοιχείων; Εδώ έρχονται να δώσουν λύση οι χώροι ονοματοδοσίας μέσω των οποίων επιτυγχάνετε η συνύπαρξη ίδιων στοιχείων και χαρακτηριστικών από διαφορετικούς χώρους. Με αυτό το τρόπο διευκρινίζετε η ασάφεια που θα μπορούσε να δημιουργηθεί συνδυάζοντας διαφορετικά XML έγγραφα με ίδια ονοματοδοσία στοιχείων η χαρακτηριστικών και διαφορετικής σημασιολογικής σημασίας.

Για την σύνταξη των χώρων ονοματοδοσίας ακολουθούμε τους εξής δύο τρόπους:

1. `<πρόθεμα:όνομα στοιχείου xmlns:πρόθεμα="URI">`

Εικόνα 3-13 : Σύνταξη χώρου ονοματοδοσίας με πρόθεμα

2. `<όνομα στοιχείου xmlns="URI">`

Εικόνα 3-14 : Σύνταξη προκαθορισμένου χώρου ονοματοδοσίας

Με τον πρώτο τρόπο χρησιμοποιούμε ένα πρόθεμα μέσω της ιδιότητας `"xmlns:πρόθεμα"` ως αναγνωριστικό του χώρου ονοματοδοσίας με τον οποίο συνδέετε το στοιχείο. Η σύνδεση του στοιχείου με το χώρο ονοματοδοσίας γίνεται μέσω της τιμής URI του προθέματος. Κατά την σύνταξη του χώρου ονοματοδοσίας με πρόθεμα πρέπει να κάνουμε αναφορά προθέματος σε όλα τα στοιχεία του εγγράφου. Η μορφή σύνταξης ενός εγγράφου XML με χρήση χώρου ονοματοδοσίας με πρόθεμα έχει την εξής μορφή:

```
<?xml version="1.0"?>
  <air:Airplane xmlns:air="http://www.airplains.com/airplanes">
    <air:model>Boeing 147</air:model>
    <air:seats>120</air:seats>
  </air:Airplane>
```

Εικόνα 3-15 : Παράδειγμα χρήσης χώρου ονοματοδοσίας με πρόθεμα

Στο παράδειγμα της εικόνας 3-15 βλέπουμε πως χρησιμοποιείτε το πρόθεμα “air” ώστε να προσδιοριστεί ο χώρος ονοματοδοσίας των στοιχείων σύμφωνα με την τιμή URI. Η τιμή αυτή στο παράδειγμα μας περιλαμβάνει μια διεύθυνση διαδικτύου(URL). Μέσο της σύνταξης χώρου ονοματοδοσίας με πρόθεμα μπορούμε να προσδιορίσουμε και στοιχεία και χαρακτηριστικά.

Στο δεύτερο τρόπο σύνταξης (βλ. εικόνα 26) ο χώρος ονοματοδοσίας είναι προκαθορισμένος. Αυτό σημαίνει ότι δεν χρειάζεται να κάνουμε αναφορά προθέματος στα υπόλοιπα στοιχεία του XML εγγράφου. Μέσο της προκαθορισμένης σύνταξης χώρου ονοματοδοσίας μπορούμε να προσδιορίσουμε μόνο τα στοιχεία και όχι τα χαρακτηριστικά. Σύμφωνα με το παράδειγμα της εικόνας 3-15 η αντίστοιχη μορφή σύνταξης ενός XML εγγράφου με προκαθορισμένο χώρο ονοματοδοσίας φαίνεται στο παράδειγμα της εικόνας που ακολουθεί:

```
<?xml version="1.0"?>
  <Airplane xmlns="http://www.airplains.com/airplanes">
    <model>Boeing 147</model>
    <seats>120</seats>
  </air:Airplane>
```

Εικόνα 3-16 : Παράδειγμα χρήσης προκαθορισμένου χώρου ονοματοδοσίας

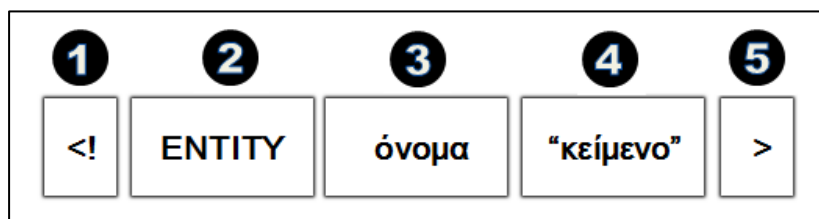
3.2.14 Οντότητες(entities)

Όταν υπάρχει η ανάγκη να αναφερθούμε σε ένα συγκεκριμένο κείμενο πολλές φορές μέσα σε ένα έγγραφο XML χρησιμοποιούμε μια οντότητα για να αποφύγουμε την άσκοπη συγγραφή ίδιου κειμένου. Μπορούμε να φανταστούμε την οντότητα σαν μια μεταβλητή που της περνάμε ένα περιεχόμενο και μέσο της οντότητας μπορούμε να επαναχρησιμοποιήσουμε το περιεχόμενο αυτό πολλές φορές μέσα στο ίδιο έγγραφο.

Υπάρχουν πολλοί τύποι οντοτήτων, οι πιο σημαντικοί είναι οι εξής:

- Παραμετρικές οντότητες (χρησιμοποιούνται μόνο στα DTD)
- Γενικές οντότητες

Σε αυτή την εργασία θα εστιάσουμε μόνο στις γενικές οντότητες που τις χρησιμοποιούμε για να αναφερθούμε σε ένα ειδικό κείμενο το οποίο μπορεί να εμφανίζετε πολλές φορές μέσα σε ένα έγγραφο XML. Στην πραγματικότητα το κείμενο αυτό μπορεί να αναφέρετε σε ένα χαρακτήρα έως και ένα ολόκληρο έγγραφο XML. Για να δηλώσουμε μια οντότητα ακολουθούμε την εξής σύνταξη:



Εικόνα 3-17 : Σύνταξη δήλωσης οντότητας σε ένα έγγραφο XML

Όπου:

1. Σήμανση αρχής ετικέτας
2. Δήλωση οντότητας
3. Δήλωση ονόματος οντότητας
4. Κείμενο που αντιπροσωπεύει η οντότητα ή αλλιώς η τιμή της οντότητας
5. Κλείσιμο ετικέτας

Εφόσον έχουμε κάνει δήλωση μίας γενικής οντότητας σε ένα έγγραφο XML, μπορούμε να αναφερθούμε στην οντότητα αυτή με το χαρακτήρα “&”, το όνομα της οντότητας και το χαρακτήρα “;” στο τέλος.

&ονομα_οντότητας;

Εικόνα 3-18 : Σύνταξη αναφοράς οντότητας σε ένα XML έγγραφο

3.2.15 Σχόλια(Comments)

Μέσο των σχολίων η γλώσσα XML μας επιτρέπει να δημιουργήσουμε σημειώσεις μέσα σε ένα έγγραφο XML η οποίες δεν μεταφράζονται από τον αναλυτή. Η σύνταξη των σχολίων ακολουθεί την εξής δομή:



Εικόνα 3-19 : Σύνταξη σχολίου σε ένα έγγραφο XML

Όπου:

1. Σήμανση αρχής σχολίου
2. Περιεχόμενο σχολίου
3. Σήμανση τέλους σχολίου

Κάθε σχόλιο μπορεί να περιέχει κάθε είδος κειμένου με κενά ειδικούς χαρακτήρες, νέα γραμμή ακόμα και σήμανση. Το μόνο που δεν επιτρέπεται είναι να ξεκινήσουμε με δυο παύλες και να τελειώσουμε με δυο παύλες. Οι διπλές παύλες είναι ήδη κατοχυρωμένες από την σύνταξη της ετικέτας. Επίσης τα σχόλια μπαίνουν μετά το πρόλογο της XML και σε οποιοδήποτε σημείο του εγγράφου.

3.2.16 Well-formed και valid XML έγγραφα

Τα XML έγγραφα σύμφωνα με το πρότυπο της γλώσσας χωρίζονται σε δύο κατηγορίες:

1. Well-formed(ορθά ορισμένα ή μορφοποιημένα) έγγραφα
2. Valid(έγκυρα) έγγραφα

Τα well-formed έγγραφα ακολουθούν τους γενικούς κανόνες σύνταξης και δομής σύμφωνα με το πρότυπο της γλώσσας XML και δεν είναι τίποτα παραπάνω από τους κανόνες που αναφέραμε στα προηγούμενα υποκεφάλαια. Συνοπτικά μερικοί από τους κανόνες αυτούς είναι:

- Κάθε έγγραφο XML ξεκινάει με τον πρόλογο
- Κάθε έγγραφο XML έχει μόνο ένα στοιχείο ρίζα
- Κάθε ετικέτα αρχής “<ετικέτα>” πρέπει να έχει και μια ετικέτα τέλους “</ετικέτα>”
- Η σήμανση της XML ξεκινάει πάντα με το σύμβολο “<” ή “&”
- Οι τιμές των χαρακτηριστικών απαιτούν εισαγωγικά
- Στην ονοματοδοσία των στοιχείων δεν μπορεί να υπάρχει το κενό μονό κάτω παύλα

Από την άλλη πλευρά τα valid έγγραφα προσδιορίζονται από ένα άλλο έγγραφο το οποίο περιλαμβάνει τους κανόνες που πρέπει να ακολουθήσει το εκάστοτε έγγραφο XML για να είναι έγκυρο. Αυτοί οι κανόνες καθορίζουν το όνομα, τον τύπο ή ακόμα και το περιεχόμενο ενός στοιχείου. Τα valid έγγραφα προσφέρουν μεγαλύτερη ευελιξία και περισσότερες δυνατότητες στην επεξεργασία και στην συντακτική εγκυρότητα ενός εγγράφου XML και για αυτό στην πλειοψηφία των διαδικτυακών εφαρμογών είναι αναπόσπαστο κομμάτι του αρχικού εγγράφου. Πάντα όμως για να είναι ένα έγγραφο XML έγκυρο πρέπει να είναι well-formed και όχι το αντίθετο.

Η εγκυρότητα(validation) ενός XML εγγράφου όπως αναφέραμε πραγματοποιείτε μέσω ενός άλλου εγγράφου το οποίο ορίζει τους κανόνες σύνταξης. Τα δημοφιλέστερα έγγραφα μοντελοποίησης και καθορισμού κανόνων για έγγραφα XML είναι τα DTD και η τεχνολογία XML schema(Βλ. παρακάτω). Τα DTD δεν πολυχρησιμοποιούνται σήμερα και θεωρούνται παλαιά τεχνολογία σε αντίθεση με την τεχνολογία XML schema η οποία λόγω της ευκολίας χρήσης της και των νέων χαρακτηριστικών που έφερε έχει επικρατήσει.

3.2.17 XML Parsers (αναλυτές)

Για να μπορέσουμε να προσπελάσουμε και να διαχειριστούμε το περιεχόμενο σε ένα έγγραφο XML χρησιμοποιούμε έναν XML αναλυτή (XML parser). Ο αναλυτής XML είναι ένα λογισμικό με το οποίο μπορούμε να διαβάσουμε, να επεξεργαστούμε και να ελέγξουμε την συντακτική εγκυρότητα ενός XML εγγράφου. Όπως κατανοούμε, θα μπορούσαμε πολύ εύκολα να ερμηνεύσουμε την λειτουργία ενός αναλυτή σαν ένα μεσισμικό λογισμικό ανάμεσα σε μια εφαρμογή και σε ένα έγγραφο XML. Έτσι μια εφαρμογή για να μπορέσει να προσπελάσει ένα έγγραφο XML και να έχει πρόσβαση στην αποθηκευμένη πληροφορία χρησιμοποιεί έναν XML αναλυτή, ο οποίος αποτελεί αναπόσπαστο κομμάτι της.

Σήμερα οι εφαρμογές κάνουν χρήση καθορισμένων API(Application Programming Interface) για να μπορέσουν να διαχειριστούν ένα έγγραφο XML. Οι δύο γενικές κατηγορίες αναλυτών XML είναι οι εξής:

1. Ο DOM(Document Object Model) αναλυτής
2. Ο SAX(Simple Api for XML) αναλυτής

Ο DOM αναλυτής είναι βασισμένος στην δενδρική αποτύπωση ενός XML εγγράφου με έναν αντικειμενοστραφή τρόπο. Κατά την χρήση του ο DOM αναλυτής διαβάζει ένα έγγραφο XML και δημιουργεί στην μνήμη του υπολογιστή την δενδρική δομή του, την οποία μετά μπορεί να προσπελάσει και να επεξεργαστεί. Με άλλα λόγια ένας αναλυτής DOM ορίζει μεθόδους για πρόσβαση και τροποποίηση στην δενδρική δομή αποτύπωσης του XML εγγράφου.

Από την άλλη πλευρά ο SAX αναλυτής διαβάζει σειριακά ένα XML έγγραφο και το αναλύει σε διάφορα μέρη, δημιουργώντας τα αντίστοιχα συμβάντα(event-based) που θα ενημερώσουν την εφαρμογή ανάλογα με τις απαιτήσεις της. Τέτοια συμβάντα μπορεί να είναι για παράδειγμα ένας κόμβος στοιχείου (element node), ένας κόμβος στοιχείου δεδομένων(text node), ένα χαρακτηριστικό κ.τ.λ. Κατά την σειριακή ανάγνωση ο SAX αναλυτής θεωρεί κάθε στοιχείο του XML εγγράφου σαν ένα κόμβο(node).

3.3 Τεχνολογίες Μοντελοποίησης XML εγγράφων

Όπως προαναφέραμε και στο υποκεφάλαιο 3.2.16 ένα έγγραφο XML θεωρείτε ορθά ορισμένο(well-formed) όταν ακολουθεί τους γενικούς κανόνες σύνταξης και δομής όπως ορίζει το πρότυπο της γλώσσας XML. Έκτος όμως από ορθά ορισμένο ένα έγγραφο XML μπορεί να θεωρηθεί και έγκυρο. Τι είναι όμως ένα έγκυρο έγγραφο; Και ποιος είναι ο σκοπός του από την στιγμή που ένα εκάστοτε έγγραφο XML είναι ορθά ορισμένο;

Πριν πάμε να απαντήσουμε στα παραπάνω ερωτήματα ας θυμηθούμε τη δυνατότητα ορισμού νέων γλωσσών σήμανσης που παρέχει η γλώσσα XML, καθορίζοντας νέα στοιχεία και χαρακτηριστικά τα οποία εξυπηρετούν τις εκάστοτε ανάγκες ενός χρήστη. Για τον πλήρη ορισμό

λοιπόν μια νέας γλώσσας σήμανσης, γεννιούνται νέα ερωτήματα στα οποία ένα ορθά ορισμένο έγγραφο XML δεν φτάνει να καλύψει. Ερωτήματα όπως το είδος περιεχομένου που θα έχει ένα στοιχείο, πόσες φορές το στοιχείο αυτό μπορεί να εμφανιστεί μέσα σε ένα συγκεκριμένο έγγραφο, τι χαρακτηρίστηκα μπορεί να έχει ένα στοιχείο. Τα ερωτήματα αυτά δεν αφορούν μονό τους γενικούς κανόνες σύνταξης και δομής της XML, αφορούν και τη σημασιολογική αξία που περιγράφουν τα δεδομένα σε ένα συγκεκριμένο έγγραφο XML. Σε αυτό το σημείο έρχεται να απαντήσει η τεχνολογία μοντελοποίησης εγγράφων για την περιγραφή νέων γλωσσών σήμανσης, ορίζοντας τη σημασιολογική υπόσταση και τους συντακτικούς κανόνες που πρέπει να ακολουθεί ένα έγγραφο XML.

Η τεχνολογία μοντελοποίησης αφορά την δημιουργία ενός εγγράφου το οποίο καθορίζει τους κανόνες για την περιγραφή της δομής και του περιεχομένου ενός συγκεκριμένου XML εγγράφου. Εδώ μπαίνει και ο όρος “έγκυρο” έγγραφο ή “valid” όπως είναι στην αγγλική ορολογία. Ένα έγκυρο έγγραφο XML σημαίνει ότι συμμορφώνετε με αυτούς τους κανόνες που ορίζει το XML Schema αλλά και με τους γενικούς κανόνες σύνταξης και δομής που ορίζει το πρότυπο της γλώσσας XML. Η δημιουργία ενός τέτοιου μοντέλου εγγράφου είναι προαιρετική καθώς η XML γλώσσα έχει σχεδιαστεί έτσι ώστε να μην χρειάζεται ένα τέτοιο έγγραφο. Παρ’ όλα αυτά η χρήση του έχει σημαντικά πλεονεκτήματα όπως:

- Ελαχιστοποιείτε η πιθανότητα λάθους σημασιολογικής ασάφειας σε ένα έγγραφο XML
- Μπορούν να καθοριστούν συγκεκριμένα στοιχεία
- Η περιγραφή των στοιχείων, των χαρακτηριστικών και του περιεχομένου γίνεται ποιο πλούσια και συγκεκριμένη
- Επιτρέπει μεγαλύτερη ευελιξία και επαναχρησιμοποίηση περιορισμών και κανόνων στη σύνταξη και τη δομή ενός εγγράφου XML

Οι δύο πιο δημοφιλείς τεχνολογίες μοντελοποίησης για την περιγραφή και τον καθορισμό κανόνων των XML εγγράφων είναι:

1. Ορισμός τύπου εγγράφων DTD(Document Type Definition)
2. XML Schema

Σύμφωνα με τις τεχνολογικές απαιτήσεις και την εξέλιξη του διαδικτύου σήμερα, οι διάφορες εφαρμογές και υπηρεσίες που χρησιμοποιούν έγγραφα XML κάνουν κατά κόρον χρήση της τεχνολογίας μοντελοποίησης XML Schemas έναντι των DTD που αποτελούν παλαιότερη τεχνολογία και πιο περιορισμένη. Συνοπτικά τα πιο σημαντικά πλεονεκτήματα που παρουσιάζει η τεχνολογία XML Schema είναι:

- Είναι πιο ευέλικτη και επεκτάσιμη για νέες τεχνολογίες
- Ισχυρότερη τεχνολογία
- Πλουσιότερη γλώσσα παρουσιάζοντας περισσότερες δυνατότητες για την περιγραφή της δομής και του περιεχομένου σε ένα έγγραφο XML
- Οι κανόνες σύνταξης και δομής των XML Schema βασίζονται στη γλώσσα XML
- Δυνατότητα επαναχρησιμοποίησης και τροποποίησης των σχημάτων
- Υποστηρίζει παραπάνω από 44 τύπους δεδομένων και επιτρέπει την δημιουργία νέων τύπων

3.4 XML Schema

Η τεχνολογία μοντελοποίησης XML Schema είναι ένα έγγραφο γραμμένο σε γλώσσα XML το οποίο ορίζει την “γραμματική” που πρέπει να ακολουθεί ένα άλλο έγγραφο XML. Η διαφορά των δύο εγγράφων έγκειται στο ρόλο που έχει το κάθε έγγραφο. Για παράδειγμα, για την ιστοσελίδα online δημοπρασίες έχουμε δημιουργήσει ένα έγγραφο XML για τις δημοπρασίες. Το έγγραφο αυτό έχει 2 σημαντικά στοιχεία. Το πρώτο στοιχείο είναι η “ώρα δημοπρασίας” και το δεύτερο στοιχείο

είναι η “αρχική τιμή” προϊόντος. Τα στοιχεία αυτά πρέπει να περιέχουν ένα συγκεκριμένο τύπο δεδομένων για να μπορέσουμε να παρουσιάσουμε σε ένα χρηστή ορθές πληροφορίες μέσω της ιστοσελίδας μας. Φανταστείτε στο περιεχόμενο του στοιχείου “αρχική τιμή” να υπήρχε αποθηκευμένη πληροφορία “δέκα χιλιάδες”. Δεν θέλουμε στο χρήστη να εμφανιστεί η αρχική τιμή προϊόντος με γραμματική υπόσταση αλλά με αριθμητική “10.000”. Αυτό το κανόνα και περιορισμό στο τύπο δεδομένων που πρέπει να ακολουθεί κάθε στοιχείο τον ορίζουμε μέσω ενός XML Schema. Το XML Schema είναι ένα ξεχωριστό έγγραφο με κατάληξη αρχείου “.xsd” το οποίο ορίζει τους κανόνες για το εκάστοτε έγγραφο XML με κατάληξη αρχείου “.xml”. Έτσι η εγκυρότητα(validation) ενός εγγράφου XML καθορίζεται από το αν ακολουθεί τους κανόνες που ορίζει το αντίστοιχο XML Schema.

Ουσιαστικά η XML Schema είναι μια γλώσσα ορισμού(definition language) όπως αλλιώς χαρακτηρίζεται και συνοπτικά ορίζει τους εξής κανόνες για ένα έγγραφο XML:

- Ορίζει τους τύπους δεδομένων(data types) για τα στοιχεία και τα χαρακτηριστικά
- Ορίζει τη δομή των στοιχείων-παιδιών
- Ορίζει τον αριθμό των στοιχείων-παιδιών
- Ορίζει τα στοιχεία που μπορούν να εμφανιστούν σε ένα έγγραφο
- Ορίζει τις ιδιότητες που μπορούν να εμφανιστούν σε ένα έγγραφο
- Ορίζει αν ένα στοιχείο είναι κενό ή αν περιέχει κείμενο ή άλλη πληροφορία
- Ορίζει σταθερές τιμές για στοιχεία και χαρακτηριστικά

3.4.1 Σύνταξη και δομή ενός εγγράφου XML Schema

Ένα από τα σημαντικότερα πλεονεκτήματα που εξασφάλισαν την άμεση επιτυχία των XML Schema είναι ότι γράφονται σε γλώσσά XML χωρίς ο χρήστης να είναι αναγκαίο να μάθει μια άλλη γλώσσά. Για αυτό το λόγο ένα έγγραφο XML Schema για να είναι ορθά σχηματισμένο πρέπει να ακολουθεί τους εξής κανόνες σύνταξης και δομής της XML:

- Πρέπει να ξεκινάει με το πρόλογο(prolog) ή αλλιώς δήλωση της XML
- Πρέπει να αποτελείται από ένα μοναδικό στοιχείο ρίζα(root element)
- Κάθε ετικέτα αρχής πρέπει να έχει και μια ετικέτα τέλους με το ίδιο όνομα
- Η συγγραφή κεφαλαίων η μικρών χαρακτήρων έχει σημασία στην ονοματοδοσία των στοιχείων και χαρακτηριστικών
- Όλα τα στοιχεία πρέπει να έχουν κατάλληλη δομή
- Όλα τα στοιχεία πρέπει να κλείνουν
- Η χρήση οντοτήτων(entities) στην XML πρέπει να χρησιμοποιείτε αποκλειστικά για ειδικούς χαρακτήρες

3.4.2 Βασικά δομικά στοιχεία των XML Schema εγγράφων

Στη παράγραφο αυτή θα προσπαθήσουμε να εξηγήσουμε μέσω ενός απλού παραδείγματος τα βασικά δομικά στοιχεία τα οποία αποτελούν και συνθέτουν ένα XML Schema έγγραφο. Στο παράδειγμα της εικόνας 3-21 που ακολουθεί, τα βασικά στοιχεία σημειώνονται αριθμητικά μέσα σε ένα κύκλο για την περεταίρω εξήγηση τους.

Υπόδειγμα XML εγγράφου:

```
<?xml version="1.0" encoding="utf-8"?>
<user>
  <First_Name>John</First_Name >
  <Last_Name>Letus</Last_Name >
  <Age>34</Age>
</user>
```

Εικόνα 3-20 : Παράδειγμα ενός XML εγγράφου

Το αντίστοιχο XML Schema με βάση το XML έγγραφο της εικόνας 3-20:

```
<?xml version="1.0" encoding="utf-8"?> ❶
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> ❷
  <xs:element name="user"> ❸
    <xs:complexType> ❹
      <xs:sequence> ❺
        <xs:element name="First_Name" type="xs:string"/>
        <xs:element name="Last_Name" type="xs:string"/>
        <xs:element name="Age" type="xs:integer"/> ❻
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Εικόνα 3-21 : Παράδειγμα ενός XML Schema εγγράφου

1. Πρώτη σημείωση εικόνας 3-21:

Η δημιουργία ενός XML Schema εγγράφου ξεκινάει πάντα με τον πρόλογο της XML και ακολουθεί τους συντακτικούς κανόνες όπως ορίζει το πρότυπο της γλώσσας.

2. Δεύτερη σημείωση εικόνας 3-21:

Στη δεύτερη σημείωση γίνεται η δήλωση του XML σχήματος. Στη δήλωση αυτή υπάρχουν δύο όροι μέσα στην ετικέτα. Ο όρος xs:schema ο οποίος αποτελεί το στοιχείο ρίζα σε κάθε έγγραφο XML Schema και ο όρος xmlns:xs ο οποίος δηλώνει το χώρο ονοματοδοσίας μέσω της τιμής URI <http://www.w3.org/2001/XMLSchema>. Ο χώρος ονοματοδοσίας παρέχει τα βασικά στοιχεία και τους τύπους δεδομένων που χρησιμοποιούνται στο εκάστοτε XML Schema.

3. Τρίτη σημείωση εικόνας 3-21:

Στη τρίτη σημείωση παρουσιάζετε ο τρόπος με τον οποίο δηλώνουμε ένα στοιχείο σε ένα έγγραφο XML Schema. Η δήλωση ενός στοιχείου γίνεται με την ετικέτα <xs:element name="όνομα_στοιχείου"> στην οποία η ιδιότητα name χρησιμοποιείτε για να δηλώσουμε το όνομα του στοιχείου το οποίο βρίσκετε στο αντίστοιχο έγγραφο XML που περιγράφει το εκάστοτε σχήμα. Στο παράδειγμα μας το στοιχείο user που δηλώνετε αντιπροσωπεύει το στοιχείο ρίζα στο αντίστοιχο έγγραφο XML της εικόνας 3-20.

4. Τέταρτη σημείωση εικόνας 3-21:

Στη τέταρτη σημείωση παρουσιάζετε η δήλωση ενός σύνθετου τύπου (complexType) στοιχείου. Τα στοιχεία κατά την τεχνολογία XML Schema διαφοροποιούνται σε δυο βασικούς τύπους. Τα στοιχεία απλού τύπου και τα στοιχεία σύνθετου τύπου. Τα απλά στοιχεία χρησιμοποιούνται για να ορίσουμε συνήθεις τύπους δεδομένων και δηλώνονται με την ετικέτα <xs:simpleType>. Επίσης τα απλά στοιχεία δεν μπορούν να περιέχουν αλλά στοιχεία παρά μόνο δεδομένα. Από την άλλη πλευρά τα στοιχεία σύνθετου τύπου χρησιμοποιούνται για τον συνδυασμό διαφορετικών τύπων δεδομένων και δηλώνονται με την ετικέτα <xs:complexType>. Τα στοιχεία σύνθετου τύπου είναι στοιχεία που εμφωλεύουν άλλα στοιχεία τα οποία μπορούν να περιέχουν και ιδιότητες.

5. Πέμπτη σημείωση εικόνας 3-21:

Στη πέμπτη σημείωση δηλώνετε το στοιχείο <xs:sequence> το οποίο χρησιμοποιείτε για να καθορίσει την σειρά εμφάνισης των στοιχείων που περιέχει στο αντίστοιχο XML έγγραφο. Η σειρά εμφανίσεις των στοιχείων πρέπει να ακολουθεί την σειρά που ορίζει το στοιχείο <xs:sequence>. Σύμφωνα με το παράδειγμά μας το στοιχείο <First_Name> πρέπει να εμφανίζεται πρώτο στο έγγραφο XML, μετά να εμφανίζεται το στοιχείο <Last_Name> και τελευταίο το στοιχείο <Age>.

6. Έκτη σημείωση εικόνας 3-21:

Στη έκτη σημείωση γίνεται η δηλώσει τριών στοιχείων. Τα στοιχεία αυτά επειδή δεν περιέχουν άλλα στοιχεία ή ιδιότητες ονομάζονται απλά στοιχεία. Μέσα στη δήλωση κάθε στοιχείου χρησιμοποιείτε η ιδιότητα type κατά την οποία η τιμή της ορίζει τον τύπο δεδομένων για εκάστοτε στοιχείο.

Παρατηρώντας την συνολική δομή και σύνταξη του παραδείγματος της εικόνας 3-21, καταλαβαίνουμε ότι ένα έγγραφο XML Schema έχει πολλές ομοιότητες με ένα έγγραφο XML. Αυτό μας βοηθά να κατανοήσουμε πιο γρήγορα την τεχνολογία μοντελοποίησης XML Schema χωρίς να χρειάζεται να μάθουμε μια καινούργια γλώσσα για να περιγράψουμε τους κανόνες και τους περιορισμούς που θέλουμε να υπάρχουν σε ένα έγγραφο XML.

3.4.3 Συχνότητα εμφάνισης στοιχείων(minOccurs & maxOccurs)

Για να καθορίσουμε την συχνότητα εμφάνισης ενός στοιχείου μέσα σε ένα έγγραφο XML χρησιμοποιούμε τα χαρακτηριστικά minOccurs και maxOccurs. Το χαρακτηριστικό minOccurs χρησιμοποιείτε για να ορίσουμε τον ελάχιστο αριθμό εμφανίσεων ενός στοιχείου μέσα σε ένα έγγραφο XML, ενώ το χαρακτηριστικό maxOccurs χρησιμοποιείτε για να ορίσουμε το μέγιστο αριθμό εμφανίσεων. Ο συνδυασμός των δύο αυτών χαρακτηριστικών μαζί, μας δίνει την δυνατότητα να ορίσουμε το εύρος εμφανίσεις ενός στοιχείου μέσα σε ένα έγγραφο XML.

Οι τιμές που μπορούν να πάρουν το minOccurs και το maxOccurs είναι οποιοσδήποτε θετικός ακέραιος αριθμός(0, 1, 2, 3...). Στην περίπτωση που θέλουμε ένα στοιχείο να εμφανίζεται άπειρες φορές μέσα σε ένα έγγραφο XML δηλώνουμε την τιμή unbounded στο χαρακτηριστικό maxOccurs.

Στον πίνακα που ακολουθεί, γίνεται αντιστοίχιση της εμφάνισης ενός στοιχείου με βάση τις τιμές που μπορούν να πάρουν τα χαρακτηριστικά minOccurs και maxOccurs.

Τιμή minOccurs	Τιμή maxOccurs	Εύρος Εμφάνισης
1	1	Ένα και μόνο ένα
1	2	Από ένα έως δύο
0	1	Κανένα ή ένα
0	unbounded	Κανένα ή περισσότερα
1	unbounded	Ένα ή περισσότερα

Πίνακας 3-1 : Αντιστοίχιση τιμών minOccurs και maxOccurs

Στο παράδειγμα που ακολουθεί φαίνεται η σύνταξη των χαρακτηριστικών minOccurs και maxOccurs μέσα σε ένα στοιχείο:

```
<xs:element name="Age" type="xs:string" minOccurs="0" maxOccurs="10"/>
```

Εικόνα 3-22 : Παράδειγμα σύνταξης minOccurs και maxOccurs μέσα σε ένα στοιχείο

Σύμφωνα με το παράδειγμα της εικόνας 3-22 το στοιχείο Age μπορεί να εμφανιστεί μέσα στο έγγραφο XML ή 10 φορές ή καμία. Έτσι μέσω των χαρακτηριστικών minOccurs και maxOccurs περιορίσαμε την αριθμητική υπόσταση του στοιχείου Age. Στην περίπτωση που το minOccurs και maxOccurs δεν έχουν οριστεί μέσα σε ένα στοιχείο, οι προκαθορισμένες τιμές που παίρνουν και τα δύο χαρακτηριστικά είναι ίση με 1. Εάν το χαρακτηριστικό minOccurs έχει οριστεί και το maxOccurs δεν έχει οριστεί τότε το maxOccurs παίρνει την τιμή που έχει το minOccurs. Ένα τώρα στην αντίθετη περίπτωση το maxOccurs έχει οριστεί και το minOccurs δεν έχει οριστεί τότε το minOccurs παίρνει την τιμή 1.

3.4.4 Δήλωση χαρακτηριστικών(attribute)

Για να δηλώσουμε μια ιδιότητα μέσα σε ένα έγγραφο XML Schema χρησιμοποιούμε το στοιχείο <xs:attribute>. Για την δήλωση ενός χαρακτηριστικού χρησιμοποιούμε την παρακάτω σύνταξη:

```
<xs:attribute name="Last_Name"/>
```

Εικόνα 3-23 : Παράδειγμα σύνταξης
χαρακτηριστικού XML Schema

Η τιμή της ιδιότητας name χρησιμοποιείται για να δηλωθεί το όνομα του χαρακτηριστικού στο αντίστοιχο XML έγγραφο που περιγράφει το εκάστοτε XML Schema.

3.4.5 Σχόλια σε ένα έγγραφο XML Schema

Όπως αναφέραμε και σε προηγούμενες ενότητες του τρίτου κεφαλαίου ένα έγγραφο XML Schema είναι στην ουσία ένα έγγραφο XML. Έτσι η σύνταξη σχολίων σε ένα έγγραφο XML Schema δεν έχει καμία διαφορά με την σύνταξη σχολίων σε ένα έγγραφο XML:

```
<!-- ΣΧΟΛΙΑ -->
```

Εικόνα 3-24 : Σύνταξη σχολίων
σε ένα XML Schema

Ένας XML επεξεργαστής σύμφωνα με τους κανόνες υλοποίησης της XML 1.0 δεν ελέγχει τα σχόλια που εμφανίζονται σε ένα έγγραφο. Έτσι στην περίπτωση που θα θέλαμε να φτιάξουμε ένα schema documentation για ένα χρήστη ή ακόμα και για μια εφαρμογή που θα μπορούσε να πάρει κάποιες χρήσιμες πληροφορίες για το εκάστοτε XML Schema τότε τα σχόλια δεν θα μπορούσαν να μας καλύψουν. Για να μπορέσουμε να διασαφηνίσουμε καλύτερα και με ποιο κομψό τρόπο ένα XML Schema σε ένα χρήστη ή σε μια εφαρμογή, η τεχνολογία XML Schema μας παρέχει τρία βασικά στοιχεία τα οποία διαφοροποιούνται από τα σχόλια:

1. <annotation>

Το στοιχείο annotation είναι το στοιχείο πατέρας το οποίο ορίζει τα σχόλια του σχήματος. Τα σχόλια τεκμηριώνονται σε μία γραμμή.

2. <documentation>

Το στοιχείο documentation χρησιμοποιείτε για να βάλουμε σχόλια κειμένου σε ένα σχήμα. Χρησιμοποιείτε ευρέως για νομικές πληροφορίες και copyrights.

3. <appInfo>

Το στοιχείο appInfo παρέχει πληροφορίες οι όποιες είναι χρήσιμες για τις εφαρμογές.

Τα στοιχεία <documentation> και <appInfo> δεν μπορούν να χρησιμοποιηθούν ανεξάρτητα αλλά πρέπει να περιέχονται μέσα στο στοιχείο <annotation>.

3.5 Η τεχνολογία μετασχηματισμού και μορφοποίησης XSL

Για τη δημιουργία εγγράφων XML μπορεί να χρησιμοποιηθεί οποιοδήποτε εργαλείο συγγραφής εγγράφων XML ή επεξεργασίας κειμένου(notepad,word,open office κ.τ.λ.). Αντίστοιχα το ίδιο ισχύει και για την εμφάνιση των εγγράφων XML τα όποια σήμερα υποστηρίζονται σχεδόν από οποιονδήποτε φυλλομετρητή. Αν επιλέξουμε να εμφανίσουμε ένα ορθά ορισμένο XML έγγραφο με έναν φυλλομετρητή(browser) που υποστηρίζει αρχεία XML, το αποτέλεσμα θα είναι μια μη μορφοποιημένη μορφή του περιεχόμενου, μαζί με τις ετικέτες του αρχείου XML:

```
<!-- Xml file for customers -->
▼<customers>
  ▼<customer>
    <name>angel</name>
    <last_name>kotsos</last_name>
    <address>Oxford 12</address>
    <phone>6946354671</phone>
  </customer>
  ▼<customer>
    <name>sofia</name>
    <last_name>Bolet</last_name>
    <address>newstreet 12</address>
    <phone>6946353765</phone>
  </customer>
</customers>
```

Εικόνα 3-25 : Μη μορφοποιημένη εμφάνιση εγγράφου XML

Για να μπορέσουμε να εμφανίσουμε το περιεχόμενο των XML εγγράφων σε μια μορφοποιημένη μορφή, παραλείποντας τις ετικέτες που περιγράφουν τη σημασιολογική υπόσταση του εγγράφου, χρησιμοποιούμε την τεχνολογία XSL(eXtensible Stylesheet Language). Η XSL δεν είναι απλά μια γλώσσα σχηματισμού φύλλων στυλ αλλά αναφέρετε σε μία οικογένεια γλωσσών για την μορφοποίηση και των μετασχηματισμό των XML εγγράφων. Οι λειτουργίες που μπορεί να παρέχει η XSL είναι οι εξής:

1. Μορφοποίηση XML εγγράφων:

Παρέχει μία γλώσσα μορφοποίησης για την επεξεργασία της οπτικής εμφάνισης εγγράφων XML.

2. Μετασχηματισμός XML εγγράφων:

Παρέχει μία γλώσσα μετασχηματισμού XML εγγράφων σε νέα έγγραφα.

3. Ορισμός τμημάτων σε XML έγγραφα

Παρέχει μία γλώσσα πλοήγησης για την επεξεργασία και τη μετακίνηση στους διάφορους κόμβους ενός XML εγγράφου.

Η οικογένεια γλωσσών στις οποίες αναφέρετε η τεχνολογία XSL, αποτελείτε από 3 γλώσσες σύμφωνα με τις παραπάνω λειτουργίες:

1. XSL-FO(eXtensible Stylesheet Language – Formatting Objects)

2. XSLT(eXtensible Stylesheet Language Transformations)

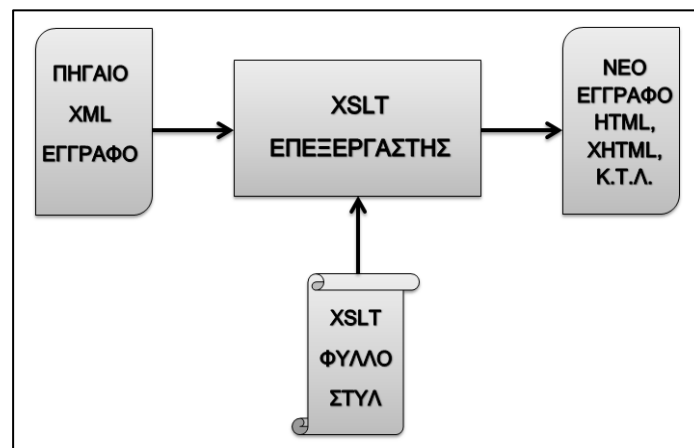
3. XPath(XML Path Language)

Η τεχνολογία XSL είναι ένα πρότυπο που έχει συσταθεί από τον οργανισμό W3C(World Wide Web Consortium). Αρχικά ξεκίνησε σαν μία προσπάθεια για να φέρει τη λειτουργικότητα της τεχνολογίας DSSSL(Document Style Semantics and Specification Language) στο πρότυπο XML και ειδικότερα στο τομέα της εκτύπωσης και στο συνδυασμό χαρακτήρων κειμένου(typesetting). Το Δεκέμβριο του 1997 υπό την επίβλεψη μιας ομάδας του W3C, ξεκίνησε η λειτουργία πάνω στην τεχνολογία XSL και τον Αύγουστο του 1998 έγινε η πρώτη δημοσίευση ενός προσχεδίου. Οι γλώσσες XSLT και XPath έγιναν σύσταση του W3C το Νοέμβριο του 1999 και η πλήρης σύσταση της τεχνολογίας XSL που περιλαμβάνει και τη γλώσσα XSL-FO τον Οκτώβριο του 2001.

3.5.1 Η γλώσσα μετασχηματισμού XSLT

Η γλώσσα μετασχηματισμού XSLT προέρχεται από τα αρχικά των αγγλικών λέξεων eXtensible Stylesheet Language Transformations και αποτελεί την σημαντικότερη τεχνολογία της XSL. Ο ρόλος της έγκειται στο μετασχηματισμό των εγγράφων XML σε άλλα έγγραφα. Κατά την διαδικασία του μετασχηματισμού, αναπροσδιορίζονται τμήματα του XML εγγράφου ώστε να μπορέσει να δημιουργηθεί ένα καινούργιο έγγραφο. Το αρχικό XML έγγραφο δεν υπόκειται σε καμία αλλαγή κατά το μετασχηματισμό, παρά μόνο χρησιμοποιείται ως βάση για τη δημιουργία ενός καινούργιου εγγράφου. Αν για παράδειγμα ένας φυλλομετρητής δεν υποστηρίζει την τεχνολογία XML, μπορούμε πολύ εύκολα να μετασχηματίσουμε ένα έγγραφο XML σε HTML ώστε να μπορέσουμε να εμφανίσουμε τα δεδομένα του XML εγγράφου στον φυλλομετρητή.

Η γλώσσα μετασχηματισμού XSLT συντάσσεται σε ένα ξεχωριστό έγγραφο το οποίο ονομάζεται φύλλο στυλ(stylesheet) και φέρει την κατάληξη αρχείου “.xsl”. Έτσι κατά το μετασχηματισμό ο επεξεργαστής της XSLT δέχεται δύο έγγραφα εισόδου. Το πηγαίο έγγραφο XML που αποτελεί τη βάση με όλα τα χρήσιμα δεδομένα και το XSLT φύλλο στυλ που παρέχει όλες τις οδηγίες που θα καθοδηγήσουν τον XSLT επεξεργαστή να παράγει ένα καινούργιο έγγραφο. Στο σχέδιο της εικόνας 3-26 που ακολουθεί, φαίνεται η διαδικασία μετασχηματισμού ενός XML εγγράφου μέσω XSLT φύλλων στυλ.



Εικόνα 3-26 : Διαδικασία μετασχηματισμού XML εγγράφου

Η διαδικασία μετασχηματισμού των XML εγγράφων υλοποιείται με έναν από τους εξής τρεις τρόπους:

- **Μετασχηματισμός από την πλευρά του πελάτη:**
Η διαδικασία μετασχηματισμού του XML εγγράφου γίνεται από πρόγραμμα που εκτελείτε από την πλευρά του πελάτη. Για παράδειγμα μια εφαρμογή που μετασχηματίζει XML έγγραφα στον φυλλομετρητή ενός χρήστη.
- **Μετασχηματισμός από την πλευρά του εξυπηρετητή:**
Ένα πρόγραμμα που εκτελεί την διαδικασία του μετασχηματισμού του XML εγγράφου από την πλευρά του εξυπηρετητή. Χρησιμοποιώντας το φύλλο στυλ και το πηγαίο έγγραφο XML μετασχηματίζει το νέο έγγραφο και το στέλνει στο πελάτη.

- **Μετασχηματισμός μέσω μη διαδικτυακών εφαρμογών:**
Ο μετασχηματισμός των XML εγγράφων μέσω μη διαδικτυακών εφαρμογών που υποστηρίζουν τη γλώσσα μετασχηματισμού XSLT.

Για να πραγματοποιηθεί οποιαδήποτε ενέργεια επεξεργασίας, κατά την διαδικασία του μετασχηματισμού μέσω της γλώσσας XSLT, θα πρέπει να υπάρχει ένας μηχανισμός εύκολης μετακίνησης ανάμεσα στους κόμβους ενός XML εγγράφου. Έτσι για το καθορισμό των μερών ενός πηγαίου εγγράφου, την δυνατότητα αναγνώρισης του τρέχοντα κόμβου και την επιλογή συγκεκριμένων κόμβων για περαιτέρω επεξεργασία, η XSLT χρησιμοποιεί την τεχνολογία XPath. Μέσω της τεχνολογίας XPath επιτυγχάνετε η δυνατότητα επιλογής ενός συνόλου από κόμβους, καθώς και η δυνατότητα σήμανσης σημείων ενός XML εγγράφου που ταιριάζουν με καθορισμένα πρότυπα (template) στο αντίστοιχο έγγραφο μετασχηματισμού XSLT.

Συνοπτικά μερικές από τις λειτουργίες της γλώσσας μετασχηματισμού XSLT είναι:

- Μετασχηματίζει τα XML έγγραφα σε αλλά έγγραφα επιλύοντας προβλήματα συμβατότητας
- Προσφέρει την δυνατότητα εμφάνισης της πληροφορίας ενός XML εγγράφου με κατάλληλο και λεπτομερή τρόπο για τον αναγνώστη
- Μπορεί να αλλάξει την σειρά εμφάνισης των στοιχείων στο XML έγγραφο εξόδου
- Μπορεί να βάλει στη σειρά τα στοιχεία κατά το μετασχηματισμό του πηγαίου εγγράφου XML
- Προσφέρει την δυνατότητα δημιουργίας νέου περιεχομένου από το ήδη υπάρχον
- Μπορεί να χρησιμοποιεί λογικές διακλαδώσεις κατά τον μετασχηματισμό μερών του πηγαίου εγγράφου

3.5.2 Σύνταξη XSLT φύλλων στυλ

Η διαδικασία μετασχηματισμού των XML εγγράφων, δέχεται σαν είσοδο ένα έγγραφο XML σε δένδροειδή μορφή και ως έξοδο παράγει ένα άλλο έγγραφο σε δένδροειδή μορφή. Η δημιουργία του τελικού εγγράφου στηρίζεται στους κανόνες που καθορίζει το αντίστοιχο έγγραφο μετασχηματισμού XSLT το οποίο πρέπει να είναι ορθά ορισμένο.

Στο παράδειγμα που ακολουθεί παρουσιάζετε ο μετασχηματισμός ενός XML εγγράφου σε ένα ορθά συντακτικά έγγραφο HTML:

- Παράδειγμα πηγαίου εγγράφου XML (αρχείο students.xml):

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="students.xsl"?>
<class>
  <student rollno="393">
    <firstname>Dinkar</firstname>
    <lastname>Kad</lastname>
    <marks>85</marks>
  </student>
  <student rollno="493">
    <firstname>Vaneet</firstname>
    <lastname>Gupta</lastname>
    <marks>95</marks>
  </student>
</class>
```

Εικόνα 3-27 : XML έγγραφο για τους μαθητές μιας τάξης

- Παράδειγμα XSLT εγγράφου (αρχείο students.xml):

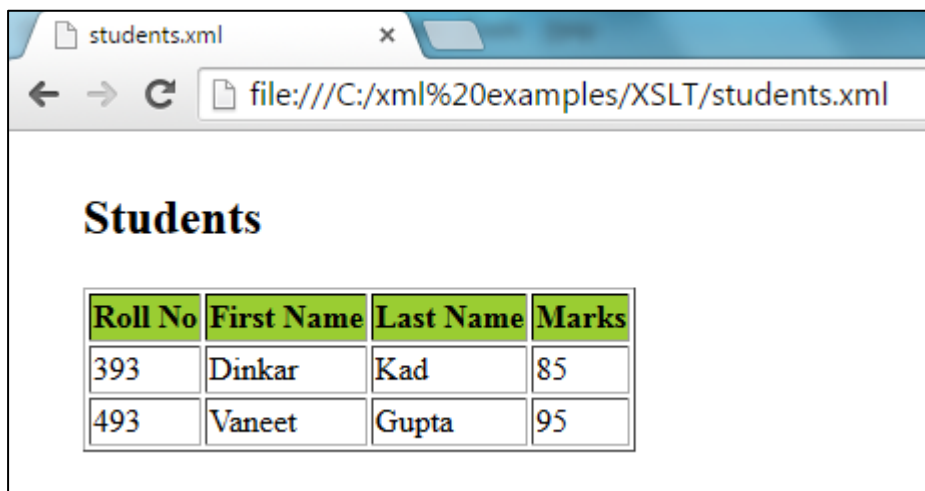
```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <xsl:stylesheet version="1.0"
3.   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4.   <xsl:template match="/">
5.     <html>
6.     <body>
7.     <h2>Students</h2>
8.     <table border="1">
9.       <tr bgcolor="#9acd32">
10.        <th>Roll No</th>
11.        <th>First Name</th>
12.        <th>Last Name</th>
13.        <th>Marks</th>
14.      </tr>
15.      <xsl:for-each select="class/student">
16.        <tr>
17.          <td><xsl:value-of select="@rollno"/></td>
18.          <td><xsl:value-of select="firstname"/></td>
19.          <td><xsl:value-of select="lastname"/></td>
20.          <td><xsl:value-of select="marks"/></td>
21.        </tr>
22.      </xsl:for-each>
23.    </table>
24.    </body>
25.  </html>
26. </xsl:template>
27. </xsl:stylesheet>

```

Εικόνα 3-28 : Παράδειγμα σύνταξης XSLT εγγράφου

- Αποτέλεσμα μετασχηματισμού σε HTML μορφή:



The screenshot shows a web browser window with the title 'students.xml' and the address bar containing 'file:///C:/xml%20examples/XSLT/students.xml'. The main content area displays the heading 'Students' followed by a table with the following data:

Roll No	First Name	Last Name	Marks
393	Dinkar	Kad	85
493	Vaneet	Gupta	95

Εικόνα 3-29 : Μετασχηματισμός XML εγγράφου σε HTML μορφή

Σύμφωνα με το παράδειγμα της εικόνας 3-28, ένα ορθά ορισμένο XSLT έγγραφο ξεκινά πάντα με το πρόλογο της XML και αποτελείται από τα εξής βασικά στοιχεία:

- **Το στοιχείο-ρίζα <xsl:stylesheet> or <xsl:transform> :**
Μέσω του στοιχείου-ρίζα stylesheet γίνεται η δήλωση του XSL τύπου εγγράφου. Η δήλωση αποτελείται από το χαρακτηριστικό version που αναφέρετε στην έκδοση της XSL γλώσσας και το χαρακτηριστικό xmlns:xsl το οποίο ορίζει το χώρο ονοματοδοσίας μέσω της τιμής URI <http://www.w3.org/1999/XSL/Transform>. (2^η κ' 3^η γραμμή εικόνας 3-28)
- **Το στοιχείο <xsl:template> :**
Μέσω του στοιχείου template ορίζουμε τα πρότυπα επεξεργασίας που θα εφαρμοστούν στους κόμβους τους οποίους ορίζει το χαρακτηριστικό match. Ανάλογα με την τιμή που παίρνει το χαρακτηριστικό match, το πρότυπο επεξεργασίας συσχετίζεται με ένα στοιχείο ή με ένα σύνολο στοιχείων του εγγράφου XML. Η τιμή του χαρακτηριστικού match καθορίζεται μέσω εκφράσεων της γλώσσας XPath. Στο παράδειγμα μας το πρότυπο επεξεργασίας προσδιορίζει ολόκληρο το έγγραφο XML σύμφωνα με την τιμή του χαρακτηριστικού match="/". (4^η γραμμή εικόνας 3-28)
- **Το στοιχείο <xsl:for-each> :**
Το στοιχείο for-each χρησιμοποιείται για την δημιουργία ενός βρόγχου επανάληψης που εκτελείτε για κάθε στοιχείο που ορίζει η τιμή-έκφραση XPath του χαρακτηριστικού select. Στο παράδειγμα μας ο βρόγχος επανάληψης θα εκτελεστεί για κάθε στοιχείο-μαθητή που εμφανίζεται στο έγγραφο XML(select="class/student"). (15^η γραμμή εικόνας 3-28)
- **Το στοιχείο <xsl:value-of> :**
Το στοιχείο value-off χρησιμοποιείται για την εξαγωγή της τιμής ενός συγκεκριμένου κόμβου από το πηγαίο έγγραφο XML ώστε να εισαχθεί μετέπειτα στο τελικό μετασχηματισμένο έγγραφο. Η επιλογή του κατάλληλου κόμβου γίνεται με τη χρήση του χαρακτηριστικού select, το οποίο παίρνει τιμές που καθορίζονται από εκφράσεις της γλώσσας XPath. Σύμφωνα με το παράδειγμα μας, στην 17^η γραμμή εξάγουμε την τιμή του χαρακτηριστικού rollno από το στοιχείο student(select="@rollno") και στην 18^η, 19^η και 20^η γραμμή εξάγουμε τις τιμές των στοιχείων firstname, lastname και marks. Προσοχή, το στοιχείο value-off εξάγει την τιμή ενός συγκεκριμένου στοιχείου και δεν αναφέρεται στις τιμές ενός συνόλου από στοιχεία. Ο συνδυασμός των στοιχείων for-each και value-off μας δίνει την ικανότητα να εξάγουμε όλες τις τιμές από κάθε στοιχείο-μαθητή του XML εγγράφου.

Στο παράδειγμα(εικόνα 3-27) του πηγαίου εγγράφου XML, ο πρόλογος της XML περιλαμβάνει άλλη μια οδηγία επεξεργασίας <?xml-stylesheet type="text/xsl" href="students.xsl"?>. Μέσω της οδηγίας αυτής, ένας XSLT επεξεργαστής μετασχηματίζει το έγγραφο XML σύμφωνα με το XSLT φύλλο στυλ που υποδεικνύει το χαρακτηριστικό href. Ποιο συγκεκριμένα μια οδηγία επεξεργασίας stylesheet περιλαμβάνει δύο βασικά χαρακτηριστικά:

1. Το χαρακτηριστικό type:

Μέσω του χαρακτηριστικού type ορίζεται ο τύπος περιεχομένου εγγράφου όπως καθορίζεται από το πρότυπο MIME type^[64]

2. Το χαρακτηριστικό href:

Μέσω του χαρακτηριστικού href καθορίζεται το URI του XSLT φύλλου στυλ, το οποίο θα μετασχηματίσει το XML έγγραφο

Ένα XSLT έγγραφο είναι ένα ορθά ορισμένο έγγραφο XML. Αν παρατηρήσουμε τη σύνταξη και τη δομή του XSLT φύλλου στυλ στο παράδειγμα της εικόνας 3-28, βλέπουμε ότι το έγγραφο ακολουθεί τους ίδιους κανόνες που ακολουθεί και ένα ορθά ορισμένο έγγραφο XML. Έτσι η

συγγραφή XSLT φύλλων στυλ παρότι θεωρείτε περίπλοκη, δεν απαιτεί την εκμάθηση μιας νέας γλώσσας αλλά τη γνώση νέων στοιχείων που προσδιορίζουν τη γλώσσα.

3.5.3 Άλλα στοιχεία που συνθέτουν τα XSLT φύλλα στυλ

Η συγγραφή XSLT φύλλων στυλ, δεν περιλαμβάνει μόνο τα στοιχεία που αναφέραμε στην ενότητα 3.5.2. Μερικά από τα στοιχεία που χρησιμοποιούνται για την υλοποίηση συγκεκριμένων ενεργειών για το μετασχηματισμό ενός XML εγγράφου, είναι τα εξής:

- **Το στοιχείο <xsl:apply-templates>:**
Είναι ένας εναλλακτικός τρόπος για τον ορισμό προτύπων(templates) στα στοιχεία-παιδιά ενός συγκεκριμένου κόμβου. Η επιλογή του κατάλληλου κόμβου γίνεται με τη χρήση του χαρακτηριστικού select. Παράδειγμα σύνταξης:
`<xsl:apply-templates select="age"/>`
- **Το στοιχείο <xsl:sort>:**
Το στοιχείο sort χρησιμοποιείτε για την ταξινόμηση ενός συνόλου κόμβων και μπορεί να έχει πέντε προαιρετικά χαρακτηριστικά. Τα δυο πιο σημαντικά χαρακτηριστικά, είναι το select που χρησιμοποιείτε για να καθορίσει με βάση ποιο κόμβο ή σύνολο κόμβων θα γίνει η ταξινόμηση και το στοιχείο order το οποίο χρησιμοποιείται για να καθορίσει τη σειρά ταξινόμησης (ascending ή descending). Παράδειγμα σύνταξης:
`<xsl:sort select="category" order="ascending"/>`
- **Το στοιχείο <xsl:output>:**
Το στοιχείο output χρησιμοποιείτε για να ορίσουμε τη μορφή(format) του τελικού εξαγόμενου εγγράφου. Συντακτικά το χαρακτηριστικό output πρέπει να εμφανίζεται στο υψηλότερο επίπεδο ιεραρχίας σε ένα XSLT έγγραφο και θα πρέπει να είναι στοιχείο-παιδί του στοιχείου-ρίζα <xsl:stylesheet>. Η δήλωση του στοιχείου μπορεί να περιέχει δέκα προαιρετικά χαρακτηριστικά εκ των οποίων το χαρακτηριστικό method(values="xml|html|text|name") χρησιμοποιείτε κατά κόρον για τον ορισμό της εξαγόμενης μορφής(προκαθορισμένη τιμή είναι το XML). Παράδειγμα σύνταξης:
`<xsl:output method="html"/>`
- **Το στοιχείο <xsl:variable>:**
Το στοιχείο variable χρησιμοποιείτε για να δηλώσουμε μια τοπική ή καθολική μεταβλητή σε ένα XSLT φύλλο στυλ. Εάν το στοιχείο variable δηλώνετε στο υψηλότερο επίπεδο ιεραρχίας των στοιχείων, θεωρείτε ως καθολική μεταβλητή ενώ αν δηλώνετε μέσα στο στοιχείο template, θεωρείτε ως τοπική μεταβλητή. Σημαντική σημείωση, είναι ότι από την στιγμή που έχουμε αποδώσει τιμή για μία μεταβλητή τότε αυτή δεν μπορεί να αλλάξει. Ο μόνος τρόπος με τον οποίο μπορεί να αλλάξει η τιμή μίας μεταβλητής, είναι να έχει δηλωθεί μέσα σε ένα βρόγχο επανάληψης for-each και να μην εμφανίζεται απευθείας ως κόμβος-παιδί του στοιχείου ρίζα stylesheet. Η σύνταξη του στοιχείου variable πρέπει να περιέχει το χαρακτηριστικό name και μπορεί να περιέχει το προαιρετικό χαρακτηριστικό select. Το χαρακτηριστικό name χρησιμοποιείται για να καθορίσουμε το όνομα μιας μεταβλητής ενώ το προαιρετικό χαρακτηριστικό select χρησιμοποιείται για να καθορίσουμε την τιμή μίας μεταβλητής. Εάν το στοιχείο select έχει ως τιμή ένα string τότε η τιμή πρέπει να περιέχεται μέσα σε εισαγωγικά. Παράδειγμα σύνταξης:
`<xsl:variable name="gender" select="male"/>`
- **Το στοιχείο <xsl:attribute>:**
Το στοιχείο attribute χρησιμοποιείτε για να εισάγουμε ένα χαρακτηριστικό σε ένα στοιχείο. Επίσης μπορεί να χρησιμοποιηθεί για να αντικαταστήσει ήδη υπάρχοντα χαρακτηριστικά με ισοδύναμα ονόματα. Η σύνταξη του στοιχείου attribute πρέπει να περιέχει το

χαρακτηριστικό name και μπορεί να περιέχει το προαιρετικό χαρακτηριστικό namespace. Το χαρακτηριστικό name χρησιμοποιείτε για να καθοριστούμε το όνομα του χαρακτηριστικού ενώ το προαιρετικό χαρακτηριστικό namespace χρησιμοποιείτε για να ορίσουμε το χώρο ονομάτων ενός χαρακτηριστικού με βάση την τιμή URI. Παράδειγμα σύνταξης:

```
<xsl:attribute name="color"/>
```

- **Το στοιχείο <xsl:if>:**

Το στοιχείο if χρησιμοποιείτε για να εισάγουμε ένα είδος επεξεργασίας υπό συνθήκη στο περιεχόμενο του εγγράφου XML. Η σύνταξη του στοιχείου if πρέπει να περιέχει το χαρακτηριστικό test, του οποίου η τιμή περιέχει την έκφραση συνθήκης που πρέπει να αξιολογηθεί. Παράδειγμα σύνταξης:

```
<xsl:if test="price > 10">
```

εμφανίσει πληροφορίας μόνο όταν η συνθήκη ισχύει

```
</xsl:if>
```

- **Το στοιχείο <xsl:choose>:**

Το στοιχείο choose χρησιμοποιείτε σε συνδυασμό με τα στοιχεία when και otherwise για να έκφραση πολλαπλές καταστάσεις υπό όρους. Η σύνταξη του στοιχείου choose πρέπει να περιέχει το χαρακτηριστικό test, του οποίου η τιμή περιέχει την έκφραση συνθήκης που πρέπει να αξιολογηθεί. Παράδειγμα σύνταξης:

```
<xsl:choose>
```

```
  <xsl:when test=" price > 10">
```

... κάποιο είδος πληροφορίας ...

```
  </xsl:when>
```

```
  <xsl:otherwise>
```

... κάποιο είδος πληροφορίας

```
  </xsl:otherwise>
```

```
</xsl:choose>
```

3.6 Η γλώσσα ερωτημάτων XPath

Η XPath είναι μια γλώσσα εκτέλεσης ερωτημάτων με σκοπό την επιλογή κόμβων(nodes) και χαρακτηριστικών από ένα έγγραφο XML. Η λειτουργία της στηρίζεται στην δενδρική αναπαράσταση ενός XML εγγράφου, παρέχοντας την ικανότητα εύκολης μετακίνησης και επιλογής κόμβων, σύμφωνα με τα κριτήρια που ορίζονται. Επιπλέον παρέχει την δυνατότητα υπολογισμού τιμών (π.χ. αριθμοί, συμβολοσειρές ή λογικές πράξεις) από το περιεχόμενο ενός XML εγγράφου.

Η γλώσσα XPath ορίστηκε από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C), με αρχική ιδέα να παρέχει ένα λειτουργικό μοντέλο, με ενιαία σύνταξη ανάμεσα στις γλώσσες XPath και XSLT. Μέχρι σήμερα υπάρχουν τρεις εκδόσεις της XPath που βρίσκονται σε χρήση, με την έκδοση της XPath 1.0 να κυριαρχεί στο σύνολο εφαρμογής της. Οι τρεις εκδόσεις της XPath είναι:

- **XPath 1.0**

Η σύσταση της πρώτης έκδοσης XPath 1.0 έγινε στις 16 Νοεμβρίου του 1999. Η έκδοση XPath 1.0 εφαρμόζετε είτε αυτόνομα με τον συνδυασμό κλήσεων από διεπαφές γλωσσών προγραμματισμού(Java, PHP, Python, C#), είτε ενσωματώνετε μέσα σε γλώσσες όπως την XSLT, XML Schema, Proc, XForms.

- **XPath 2.0**

Η σύσταση της έκδοσης XPath 2.0 έγινε στις 23 Ιανουαρίου 2007. Η XPath 2.0 παρέχει μία σειρά από εφαρμογές που προϋπήρχαν στην έκδοση της XPath 1.0, άλλα δεν είχαν τόση ευρεία χρήση. Η προδιαγραφή που ορίζει η έκδοση της XPath 2.0 είναι πολύ μεγαλύτερη από την έκδοση 1.0, με θεμελιώδεις αλλαγές στην κεντρική ιδέα της γλώσσας όπως είναι ο τύπος συστήματος.

- **XPath 3.0**

Η σύσταση της έκδοσης XPath 3.0 έγινε στις 8 Απριλίου 2014. Η Τρίτη έκδοση έφερε μεγάλη καινοτομία στη γλώσσα, παρουσιάζοντας νέα χαρακτηριστικά όπως είναι η υποστήριξη συναρτήσεων ως τιμές πρώτης κατηγορίας.^[63]

Η σύνταξη εκφράσεων XPath υλοποιείτε με ένα κόμβο ή με ένα σύνολο κόμβων που αναπαριστά/ουν το μονοπάτι το οποίο οδηγεί σε ένα συγκριμένο κόμβο. Η σύνθεση ενός μονοπατιού αποτελείται από τους εξής κόμβους:

- Κόμβος ρίζα (root node)
- Κόμβος στοιχείο (element node)
- Κόμβος κειμένου (text element)
- Κόμβος χαρακτηριστικών (Attribute node)
- Κόμβος οδηγιών επεξεργασίας (processing instruction nodes)
- Κόμβος χώρου ονομάτων (namespace nodes)
- Κόμβος σχολίων (comment nodes)

Στο παράδειγμα που ακολουθεί παρουσιάζετε η χρήση μερικών XPath εκφράσεων για την επιλογή συγκεκριμένων κόμβων μέσα από το έγγραφο XML.

```
<books>
  <book><title author="Victoria Aveyard">Red Queen</title></book>
  <book><title author="E.L. James">Grey</title></book>
  <book><title author="Kristin Hannah">The Nightingale</title></book>
</books>
```

Εικόνα 3-30 : XML έγγραφο για προσωπική λίστα βιβλίων

Σύμφωνα με το παράδειγμα της εικόνας 3-30:

- Αν θέλουμε να μας επιστραφούν όλοι οι τίτλοι (title) των βιβλίων, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book/title ή **/books/book/title[text()]**
- Αν θέλουμε να μας επιστραφεί ο πρώτος τίτλος (title) βιβλίου, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book[1]/title
- Αν θέλουμε να μας επιστραφούν όλα τα χαρακτηριστικά author από όλους τους τίτλους βιβλίων, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book/title/@author
- Αν θέλουμε να μας επιστραφούν όλα τα βιβλία με τους τίτλους τους και τον συγγραφέα, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book
- Αν θέλουμε να μας επιστραφεί ο τίτλος του βιβλίου The Nightingale, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book/title[text()='The Nightingale']
- Αν θέλουμε να μας επιστραφεί ο συγγραφέας που έχει γράψει το βιβλίο The Nightingale, η έκφραση XPath θα είχε την εξής σύνταξη:
/books/book[title='The Nightingale']/title/@author

3.7 Η γλώσσα προγραμματισμού PHP

Η γλώσσα προγραμματισμού PHP αποτελεί ένα ευρέως διαδεδομένο εργαλείο για τη συγγραφή σεναρίων από την μεριά του εξυπηρετητή (server-side scripting), με σκοπό την δημιουργία σελίδων διαδικτύου με δυναμικό περιεχόμενο. Στην αγγλική ορολογία χαρακτηρίζετε με τον όρο “general-purpose programming language” δηλαδή γλώσσα προγραμματισμού γενικού σκοπού, διευρύνοντας τους ρόλους χρήσης και εφαρμογής της.



Εικόνα 3-31 : PHP Λογότυπο

Η συγγραφή του κώδικα PHP πλαισιώνετε συνήθως μέσα σε μία σελίδα HTML, όπου πριν την εμφάνιση της σελίδας στο φυλλομετρητή ενός χρήστη, ο PHP κώδικας επεξεργάζεται και μεταφράζετε μέσω ενός συμβατού διακομιστή για την παραγωγή του τελικού περιεχομένου. Ο κώδικας PHP δεν μπορεί να εμφανιστεί μόνος του σε μια ιστοσελίδα, πρώτα ενθυλακώνετε μέσα στην HTML σελίδα, μετά μεταφράζετε και τέλος εκτελείτε και εμφανίζετε σε συνδυασμό με το HTML περιεχόμενο στο φυλλομετρητή του χρήστη. Αυτή η εξάρτηση της PHP από την HTML για την εμφάνιση των αποτελεσμάτων εκτέλεσης του κώδικα PHP σε μια ιστοσελίδα, αποτελεί σημαντικό πλεονέκτημα προσφέροντας έναν ασφαλή τρόπο πρόσβασης των PHP σελίδων σε βάσεις δεδομένων και άλλες ασφαλείς πληροφορίες. Τα αρχεία PHP έχουν την κατάληξη .php, .php1, .php2, .php3, .php4, .php5, .phps ανάλογα με την έκδοση της PHP γλώσσας και του εκάστοτε ρόλου χρήσης και εφαρμογής της. Για να μπορέσει να λειτουργήσει και να εμφανιστεί ο κώδικας PHP που ενσωματώνετε σε ένα αρχείο HTML, πρέπει από την πλευρά του διακομιστή να έχουν γίνει οι κατάλληλες ρυθμίσεις στα MIME Types.^[64] Επιπλέον θα πρέπει ο διακομιστής να είναι ρυθμισμένος ώστε να μπορεί να μεταφράζει και να εκτελεί σεναρία PHP.

Σήμερα η PHP είναι συμβατή με τα περισσότερα λειτουργικά συστήματα, όπως Microsoft Windows, Linux, Mac, Unix και άλλα. Επίσης παρέχει υποστήριξη για τους περισσότερους διακομιστές διαδικτύου, Apache, Microsoft IIS, Netscape, Simple Web Server κ.α. και υποστηρίζει μία μεγάλη λίστα από βάσεις δεδομένων όπως MySQL, ODBC, Oracle, PostgreSQL, dBase, SQLite, IBM DB2 και άλλες πολλές.

3.7.1 Ιστορία της PHP

Η σχεδίαση και ανάπτυξη της γλώσσας προγραμματισμού PHP, ξεκίνησε το 1994 από τον Rasmus Lerdorf, όπου η πρώτη ενσάρκωση της PHP γλώσσας ήταν ένα απλό σενάριο CGI σεναρίων^[66] γραμμένα σε δυαδική μορφή, στη γλώσσα προγραμματισμού C. Αρχικά χρησιμοποιήθηκε για να παρακολουθεί τους επισκέπτες που έβλεπαν το βιογραφικό του σημείωμα στο διαδίκτυο. Η σουίτα των σεναρίων που χρησιμοποιούσε ονομάστηκε Personal Home Page Tools ή αλλιώς PHP Tools. Με την πάροδο του χρόνου παρουσιάστηκαν όλο και περισσότερες ανάγκες και για αυτό ο Rasmus Lerdorf ξαναέγραψε τα PHP Tools, παράγοντας μια μεγαλύτερη και πλουσιότερη υλοποίηση. Το νέο μοντέλο ήταν ικανό να αλληλεπιδρά με βάσεις δεδομένων και επιπλέον παρείχε ένα πλαίσιο (framework) στο οποίο οι χρήστες θα μπορούσαν να αναπτύξουν και να υλοποιήσουν απλές δυναμικές εφαρμογές διαδικτύου. Το Ιούνιο του 1995 ο Rasmus Lerdorf κυκλοφόρησε το πηγαίο κώδικα για τα PHP Tools στο κοινό, παρέχοντας την δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν τον κώδικα με τον δικό τους τρόπο. Το Σεπτέμβριο του ίδιου χρόνου έγιναν κάποιες σημαντικές αλλαγές και το όνομα PHP Tools μετονομάστηκε σε FI. Η νέα εφαρμογή είχε κάποιες βασικές λειτουργίες της PHP όπως την γνωρίζουμε σήμερα και έμοιαζε στη γλώσσα Perl. Ένα μήνα μετά, τον Οκτώβριο του 1995 ο Rasmus ξαναέγραψε τον κώδικα και κυκλοφόρησε το νέο μοντέλο με το όνομα Personal Home Page Construction Kit, φέρνοντας πίσω το όνομα PHP.

Τον Απρίλιο του 1996, μετά από πολλές αλλαγές του κώδικα, ο Rasmus παρουσίασε την PHP/FI, όπου ήταν ο συνδυασμός των προηγούμενων κυκλοφοριών. Έτσι το νέο μοντέλο PHP/FI διαμορφώθηκε από μια σουίτα εργαλείων σε μια γλώσσα προγραμματισμού και πολύ γρήγορα, τον Ιούνιο του ίδιου χρόνου παρουσιάστηκε η έκδοση 2.0. Την ίδια χρονιά ο Zeev Suraski και ο Andi

Gutmans ξαναέγραψαν τον αναλυτή(parser), θέτοντας έτσι τις βάσεις για την έκδοση PHP 3. Το αναδρομικό ακρόνυμο της γλώσσας PHP μετονομάστηκε Hypertext Preprocessor. Η επίσημη έκδοση της PHP 3 ήρθε τον Ιούνιο του 1998. Μετά την κυκλοφορία της PHP 3, ο Suraski και ο Gutmans ξαναέγραψαν τον πυρήνα της PHP και μέσα στον επόμενο χρόνο, το 1999 παρουσίασαν την μηχανή Zend^[65] για την ερμηνεία της PHP. Στην συνέχεια ο Suraski και ο Gutmans δημιούργησαν την εταιρία Zend και στις 22 Μαΐου του 2000 κυκλοφόρησαν την έκδοση της PHP 4 υπό την λειτουργία της μηχανής Zend 1.0.

Τον Ιούλιο του 2004 κυκλοφόρησε η έκδοση PHP 5 υπό την λειτουργία της μηχανής Zend 2.0. Η νέα έκδοση έφερε νέα χαρακτηριστικά στην γλώσσα, με σημαντικές βελτιώσεις στην υποστήριξη του αντικειμενοστραφή προγραμματισμού. Μέσα στα επόμενα χρόνια οι ερμηνευτές (interpreters) της PHP, έγιναν διαθέσιμοι στα περισσότερα λειτουργικά συστήματα(32-bit και 64-bit) και ξεκίνησε η προσπάθεια για την κυκλοφορία της έκδοσης PHP 6 υπό την επίβλεψη του Andrei Zmievski. Η έκδοση της PHP 6 προβλεπόταν να περιλαμβάνει εγγενή υποστήριξη για Unicode αλλά λόγω των μεγάλων αλλαγών που θα επέφερε, τόσο στη γλώσσα όσο και στον κώδικα των χρηστών, η έκδοση αυτή της PHP εγκαταλείφτηκε. Στην 12 Νοεμβρίου του 2015, μετά από σχεδόν 12 χρονιά κυκλοφόρησε η καινούργια έκδοση PHP 7 υπό την λειτουργία της μηχανής Zend 3.0. Η νέα έκδοση που παραμένει μέχρι και σήμερα έφερε σημαντικές αλλαγές βελτιώνοντας αισθητά την απόδοση της γλώσσας.

3.7.2 Πλεονεκτήματα της PHP

Η PHP είναι μια ισχυρή γλώσσα συγγραφής σεναρίων και αποτελεί σήμερα μια από τις πιο διαδεδομένες γλώσσες για την σχεδίαση, ανάπτυξη και υλοποίηση δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών που αλληλεπιδρούν με βάσεις δεδομένων. Η μεγάλη επιτυχία της γλώσσας PHP οφείλετε στα εξής πλεονεκτήματα:

- Ο πηγαίος κώδικας της και ο συντακτικός αναλυτής της διανέμονται ελεύθερα στο διαδίκτυο.(open source)
- Υποστηρίζει σχεδόν όλα τα λειτουργικά συστήματα
- Υποστηρίζει ένα μεγάλο αριθμό βάσεων δεδομένων
- Η συγγραφή και εκμάθηση του κώδικα PHP είναι παρά πολύ εύκολη και κατανοητή
- Διαθέτει ενσωματωμένες εντολές και συναρτήσεις για την υποστήριξη πολλών βάσεων δεδομένων
- Υποστηρίζει τον αντικειμενοστραφή προγραμματισμό
- Υποστηρίζει το διαδικαστικό προγραμματισμό
- Υποστηρίζει αρχεία XML με ενσωματωμένες συναρτήσεις και εντολές
- Παράγει δυναμικό περιεχόμενο
- Υποστηρίζει ένα μεγάλο αριθμό πρωτοκόλλων όπως HTTP, POP3, LDAP, SNMP κ.α.
- Παρέχει ενσωματωμένες βιβλιοθήκες για πολλές διαδικασίες διαδικτύου
- Δυνατότητα αποστολής email και δημιουργίας εικόνων gif

3.7.3 Η επέκταση simpleXML

Το SimpleXML αποτελεί μία επέκταση της γλώσσας PHP που επιτρέπει με εύκολο τρόπο να χειριστούμε και να πάρουμε τα δεδομένα ενός XML εγγράφου. Εισήχθη για πρώτη φορά στην έκδοση της PHP 5 σαν μια αντικειμενοστραφή προσέγγιση, για να παρέχει ένα σετ από εργαλεία, για την μετατροπή ενός XML εγγράφου σε ένα αντικείμενο το οποίο μπορεί να επεξεργαστεί πολύ εύκολα με κανονικούς δείκτες και πίνακες επανάληψης (array iterator^[67-68]) τους οποίους μπορούμε να διασχίσουμε. Επί της ουσίας η επέκταση SimpleXML αποτελεί μια εύκολη μέθοδο για να πάρουμε το όνομα ενός στοιχείου, το περιεχόμενό του και τα χαρακτηριστικά του, εφόσον γνωρίζουμε την δομή του XML εγγράφου.

Ο αναλυτής της επέκτασης SimpleXML βασίζεται στην δένδρική δομή και σε σύγκριση με τους αναλυτές DOM και Expat είναι πολύ πιο απλός στην χρήση του και χρειάζεται λιγότερες γραμμές κώδικα για να διαβάσει τα δεδομένα κειμένου ενός στοιχείου. Οι συναρτήσεις που αποτελούν την επέκταση SimpleXML είναι οι εξής:

- **__construct()**
Δημιουργεί ένα καινούργιο αντικείμενο SimpleXMLElement
- **addAttribute()**
Προσθέτει ένα χαρακτηριστικό στο SimpleXML στοιχείο
- **addChild()**
Προσθέτει ένα στοιχείο-παιδί στο SimpleXML στοιχείο
- **attribute()**
Επιστρέφει τις τιμές των χαρακτηριστικών ενός στοιχείου
- **asXML()**
Επιστρέφει ένα ορθά ορισμένο XML έγγραφο σε μια συμβολοσειρά από ένα SimpleXML αντικείμενο
- **children()**
Επιστρέφει τα παιδιά ενός συγκεκριμένου κόμβου
- **count()**
Μετρά τα παιδιά ενός συγκεκριμένου κόμβου
- **getDocNamespaces()**
Επιστρέφει τους χώρους ονοματοδοσίας που έχουν δηλωθεί στο έγγραφο
- **getNamespaces()**
Επιστρέφει τους χώρους ονοματοδοσίας που χρησιμοποιούνται στο έγγραφο
- **getName()**
Επιστρέφει το όνομα της XML ετικέτας που παραπέμπει το SimpleXML στοιχείο
- **registerXPathNamespace()**
Δημιουργεί το περιεχόμενο ενός χώρου ονοματοδοσίας για το επόμενο ερώτημα XPath
- **saveXML()**
Αποθηκεύει τις εκάστοτε αλλαγές του SimpleXML αντικειμένου σαν XML δομή σε ένα αρχείο
- **simplexml_import_dom()**
Επιστρέφει ένα αντικείμενο SimpleXMLElement από ένα κόμβο DOM
- **simplexml_load_file()**
Μετατρέπει ένα έγγραφο XML σε ένα αντικείμενο SimpleXMLElement
- **simplexml_load_string()**
Μετατρέπει μια συμβολοσειρά XML σε ένα αντικείμενο SimpleXMLElement
- **xpath()**
Εκτελεί μια ερώτηση XPath στα δεδομένα ενός XML εγγράφου

SimpleXML Iteration Functions:

- **current()**
Επιστρέφει το τρέχον στοιχείο
- **getChildren()**
Επιστρέφει το στοιχείο-παιδί από το τρέχον στοιχείο
- **hasChildren()**
Ελέγχει αν το τρέχον στοιχείο έχει παιδιά
- **key()**
Επιστρέφει το τρέχον κλειδί
- **next()**
Ενέργεια μετακίνησης στο επόμενο στοιχείο
- **rewind()**
Ενέργεια επιστροφής στο πρώτο στοιχείο
- **valid()**
Ελέγχει αν το τρέχον στοιχείο είναι έγκυρο

3.7.4 Η επέκταση DOM στην γλώσσα PHP

Η επέκταση DOM της PHP γλώσσας, μας επιτρέπει να επεξεργαστούμε ένα XML έγγραφο μέσω του DOM API. Σε σύγκριση με την επέκταση SimpleXML που μας επιτρέπει πολύ εύκολα και γρήγορα να χειριστούμε XML έγγραφα, όταν θέλουμε να δουλέψουμε πάνω σε αρχεία XML με μεγάλη χωρητικότητα χρησιμοποιούμε την επέκταση DOM.

Η επέκταση DOM αποτελεί μια υλοποίηση του προτύπου W3C DOM και βασίζεται περισσότερο στο μοντέλο αντικειμένου από ότι η επέκταση SimpleXML. Όπως και ο αναλυτής της επέκτασης SimpleXML έτσι και ο αναλυτής DOM βασίζεται στην δενδρική δομή.

3.7.5 Η έκδοση PHP 5.3 που χρησιμοποιήθηκε

Η έκδοση PHP που χρησιμοποιήθηκε για την υλοποίηση του πρακτικού μέρους της πτυχιακής εργασίας είναι η έκδοση 5.3.13. Η έκδοση PHP 5.3 κυκλοφόρησε στις 30 Ιουνίου του 2009 και σε συνδυασμό με την ανανεωμένη έκδοση 5.3.13 που κυκλοφόρησε στις 8 Μαΐου του 2012 παρέχουν σημαντικές βελτιώσεις στη γλώσσα. Συνοπτικά αναφέρονται:

- Υποστήριξη χώρων ονοματοδοσίας(namespaces)
- Βελτιωμένη υποστήριξη των Windows
- Βελτιωμένη υποστήριξη για sqlite3 και mysqlnd
- Υποστήριξη αρχείων PHP(phar)
- Καλύτερη υποστήριξη για MIME^[64]
- Βελτιωμένη υποστήριξη και δυνατότητες για αντικειμενοστραφή προγραμματισμό
- Υποστήριξη closures^[81]
- Υποστήριξη late static bindings^[69]
- Βελτίωση σε θέματα ασφάλειας
- Βελτιωμένη υποστήριξη στις ρυθμίσεις CGI (CVE-2012-2311)
- Υποστήριξη για garbage collection (GC)^[70]
- Υποστήριξη για jump label (limited goto)^[71]

3.8 Η γλώσσα JavaScript

Η JavaScript(συντομογραφία JS) είναι μια υψηλού επιπέδου, δυναμική, με ασθενής τύπους^[72], διερμηνευμένη γλώσσα προγραμματισμού. Αποτελεί μια γλώσσα σεναρίων από την πλευρά του πελάτη(client-side), βασιζόμενη στα πρωτότυπα(prototype-based), με συναρτήσεις πρώτης τάξης και διαφορετικά προγραμματιστικά παραδείγματα(multi-paradigm). Επίσης υποστηρίζει τον αντικειμενοστραφή, τον συναρτησιακό και τον προστακτικό προγραμματισμό. Αν και η σύνταξή της είναι επηρεασμένη από την γλώσσά C, οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Scheme και Self. Πολλά ονόματα και συμβάσεις ονοματοδοσίας έχουν παρθεί από την γλώσσά JAVA αλλά στην πραγματικότητα αποτελούν δυο πολύ διαφορετικές τεχνολογίες με διαφορετική σημασιολογία. Παράλληλα με την γλώσσα HTML και CSS, αποτελεί μια από τις τρεις βασικές τεχνολογίες για την παραγωγή δυναμικού περιεχομένου του Παγκοσμίου Ιστού.



Εικόνα 3-32 : JavaScript Λογότυπο

Η JavaScript αρχικά αποτελούσε βασικό μέρος της υλοποίησης των φυλλομετρητών Παγκόσμιου Ιστού, ώστε να μπορούν τα σενάρια που τρέχουν από την πλευρά του πελάτη(client-side scripts), να ανταλλάσουν δεδομένα ασύγχρονα, να επικοινωνούν με το πελάτη, και να υποστηρίζουν την δυναμική αλλαγή περιεχομένου στις ιστοσελίδες. Επίσης χρησιμοποιείτε σε μεγάλο βαθμό για να προσφέρει διαδραστικότητα(interactivity) και δυνατότητες αλληλεπίδρασης ανάμεσα στις ιστοσελίδες και τους χρήστες που τις επισκέπτονται.

Σήμερα η γλώσσα Javascript είναι μια από τις πιο δημοφιλής γλώσσες διαδικτύου και εκτός από την χρήση της σε εφαρμογές ιστοσελίδων, χρησιμοποιείτε σε έγγραφα PDF, σε μικρές εφαρμογές για την επιφάνεια εργασίας(desktop widgets) και σε εξειδικευμένους φυλλομετρητές(site-specific browsers)^[76]. Στα πλαίσια ανάπτυξης της γλώσσας η Javascript επεκτάθηκε στην ανάπτυξη εφαρμογών του Παγκόσμιου Ιστού από την πλευρά του διακομιστή (server-side), κάτι το οποίο την έκανε ακόμα πιο δημοφιλή σαν γλώσσα. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript^[74-75].

3.8.1 Ιστορία της JavaScript

Η γλώσσα Javascript αρχικά αναπτύχθηκε από τον Brendan Eich, ενώ εργαζόταν για την εταιρεία Netscape Communications Corporation. Ξεκίνησε με την επωνυμία Mocha αλλά στην συνέχεια μετονομάστηκε σε LiveScript, όπου τον Σεπτέμβριο του 1995 κυκλοφόρησε για πρώτη φορά στην αγορά με δοκιμαστικές(beta) εκδόσεις μαζί με το πρόγραμμα περιήγησης Netscape Navigator 2.0. Στις 4 Δεκεμβρίου του 1995 η LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση που έγινε με την εταιρεία Sun Microsystems. Καθοριστικό ρόλο στην ονομασία JavaScript έπαιξε η γλώσσα προγραμματισμού JAVA η οποία επηρέασε σημαντικά την όλη ανάπτυξη της γλώσσας. Το Νοέμβριο του 1996 η Netscape Communications Corporation ανακοίνωσε ότι είχε υποβάλει την γλώσσα JavaScript για εξέταση ως βιομηχανικό πρότυπο, στην οργάνωση τυποποίησης των γλωσσών προγραμματισμού Ecma International. Η επίτευξη του στόχου αυτού, είχε ως αποτέλεσμα την τυποποιημένη μορφή της γλώσσας που ονομάζεται ECMAScript.^[74-75]

Αρχικά πολλοί επαγγελματίες προγραμματιστές είχαν υποτιμήσει την γλώσσα JavaScript λόγο του ότι μόνο ερασιτέχνες συγγραφείς ιστοσελίδων την χρησιμοποιούσαν. Όμως η ανάπτυξη της μέσα στα επόμενα χρονιά έκανε αισθητή τις προγραμματιστικές ικανότητες, αποκτώντας μεγάλη επιτυχία ως γλώσσα που εκτελείτε από την μεριά του πελάτη και παρέχει διαδραστικές δυνατότητες στις ιστοσελίδες. Έτσι, σιγά σιγά όλο και περισσότερα προγράμματα περιήγησης Παγκόσμιου Ιστού την περιελάμβαναν στις νέες του εκδόσεις. Με την εμφάνιση κυρίως της τεχνολογίας Ajax, η JavaScript κέρδισε τον επαγγελματικό σεβασμό σαν γλώσσα προγραμματισμού θέτοντας νέες δυνατότητες για την ίδια την γλώσσα και επεκτείνοντας την έξω από τα προγράμματα περιήγησης του Παγκόσμιου Ιστού. Τον Ιανουάριο του 2009, ιδρύθηκε το έργο με στόχο τον καθορισμό ενός κοινού

προτύπου βιβλιοθήκης, που στόχευε κυρίως στην ανάπτυξη της JavaScript σε νέες τεχνολογίες(π.χ. server-side).

3.9 Η τεχνολογία Ajax

Η τεχνολογία Ajax(asynchronous JavaScript and XML) αποτελεί μια ομάδα αλληλένδετων τεχνικών ανάπτυξης διαδικτύου, που χρησιμοποιείτε από την πλευρά του πελάτη(client-side) για την δημιουργία ασυγχρόνιστων εφαρμογών.^[77] Μέσω της τεχνικής Ajax οι ιστοσελίδες και οι διαδικτυακές εφαρμογές μπορούν να στείλουν και να ανακτήσουν δεδομένα από ένα διακομιστή ασύγχρονα στο παρασκήνιο, χωρίς να χρειάζεται να παρέμβουν στην εμφάνιση και την συμπεριφορά της όλης σελίδας. Η ανάκτηση των δεδομένων γίνεται μέσω της χρήσης του αντικείμενου XMLHttpRequest.^[80]



Εικόνα 3-33 : Ajax Λογότυπο

Όπως προαναφέραμε στην προηγούμενη παράγραφο, ο Ajax αποτελεί ένα συνδυασμό τεχνολογιών με σκοπό να παρέχει μια ασυγχρόνιστη επικοινωνία μεταξύ των διαδικτυακών εφαρμογών και του διακομιστή. Έτσι κερδίζουμε και χρόνο και χώρο. Χρόνο κερδίζουμε γιατί ο διακομιστής δεν στέλνει ολόκληρη την σελίδα στον πελάτη αλλά στέλνει μονό τα κομμάτια της σελίδας που έχει αλλάξει το περιεχόμενό τους και χώρο κερδίζουμε γιατί δεν φορτώνουμε το δίκτυο με την άσκοπη αποστολή πληροφορίας που είναι αμετάβλητη και δεν χρειάζεται να ξανά σταλεί στο χρήστη. Για την διαμόρφωση και την έμφαση μίας ιστοσελίδας μπορούμε να χρησιμοποιήσουμε συνδυαστικά την HTML και το CSS. Αντίστοιχα μέσω της JavaScript επιτυγχάνετε η πρόσβαση στο DOM^[82], η δυναμική εμφάνιση περιεχομένου και η διαδραστικότητα μεταξύ ενός χρήστη και της πληροφορίας που του εμφανίζεται. Επιπλέον η Javascript μαζί με το αντικείμενο XMLHttpRequest παρέχουν μία μέθοδο για την ανταλλαγή δεδομένων μεταξύ ενός φυλλομετρητή και του διακομιστή χωρίς να χρειάζεται η επαναφόρτιση ολόκληρης της σελίδας για να αλλάξουμε το περιεχόμενο σε ένα η περισσότερα σημεία της.

Σήμερα ο Ajax αποτελεί την πιο σύγχρονη τεχνολογία στο προγραμματισμό διαδικτύου και χρησιμοποιείτε ευρύτατα από τις ιστοσελίδες και τις διάφορες διαδικτυακές εφαρμογές. Ένα ζωντανό παράδειγμα χρήσης της τεχνικής Ajax είναι οι χάρτες της Google(Google Maps).

3.10 Η γλώσσα σήμανσης υπερκειμένου HTML

Η HTML(HyperText Markup Language) αποτελεί την κύρια γλώσσα σήμανση, για την δημιουργία ιστοσελίδων που το περιεχόμενό τους μπορεί να περιέχει κείμενο, εικόνες, ήχο, βίντεο, διαδραστικές φόρμες και αλλά πολυμεσικά στοιχεία. Επίσης επιτρέπει την εισαγωγή μέσα στον συντακτικό της κώδικα, να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως PHP και JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων και κάνουν εφικτή την ενθυλάκωση διαδικτυακών εφαρμογών. Το πρότυπο της γλώσσας HTML, παρέχει όλες τις κατάλληλες μεθόδους για την δημιουργία δομημένων εγγράφων, καθορίζοντας σημασιολογικά το περιεχόμενο ενός HTML αρχείου με κεφαλίδες, παραγράφους, κυρίως κείμενο, λίστες, συνδέσμους, παραθέσεις και άλλα. Η έννοια των δομημένων εγγράφων αναφέρεται στα έγγραφα που το περιεχόμενό τους περιγράφεται από κάποιο κώδικα σήμανσης και μορφοποίησης περιεχομένου.



Εικόνα 3-34 : HTML Λογότυπο

Η σύνταξη της γλώσσας HTML γράφεται υπό την μορφή στοιχείων HTML, τα οποία αποτελούνται από ετικέτες οι οποίες περικλείονται μέσα στα σύμβολα “μεγαλύτερο από” και “μικρότερο από” (π.χ. <html>). Ο ορισμός σύνταξης μιας HTML ετικέτας ακολουθεί την εξής δομή:



Εικόνα 3-35 : Ορισμός σύνταξης HTML ετικέτας

Όπου:

1. Ετικέτα έναρξης
2. Περιεχόμενο(κείμενο, εικόνα, βίντεο και άλλα)
3. Ετικέτα λήξης

Όπως βλέπουμε στην εικόνα 3-35 οι ετικέτες HTML πάνε κυρίως σε ζευγάρια, εκτός ορισμένων περιπτώσεων. Ανάμεσα στις ετικέτες οι σχεδιαστές ιστοσελίδων μπορούν να βάλουν κάποιο κείμενο, μια εικόνα, έναν ήχο κ.λπ. Έτσι η δημιουργία μια σελίδας HTML αποτελείται από ένα σύνολο ετικετών οι οποίες συνθέτουν τα δομικά στοιχεία της HTML. Για να μπορέσουμε να διαβάσουμε μια ιστοσελίδα HTML κάνουμε χρήση των φυλλομετρητών(web browser). Ο σκοπός των φυλλομετρητών είναι να διαβάζουν τα έγγραφα HTML και να τα συνθέτουν σε σελίδες διαδικτύου οι οποίες μπορούν να διαβαστούν, να ακουστούν και γενικά να αλληλεπιδράσουν με ένα χρήστη.

3.10.1 Η ιστορία της HTML

Το 1980 ο φυσικός Tim Berners-Lee, ο οποίος εργαζόταν στο εργαστήριο φυσικής CERN στη Γενεύη δημιούργησε το ENQUIRE^[73], ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN. Μετά από μερικά χρόνια το 1989, πρότεινε ένα σύστημα υπερκειμένου το οποίο βασιζόταν στο διαδίκτυο. Έτσι δημιουργείσαι την προδιαγραφή της HTML και έγραψε το πρώτο φυλλομετρητή και το λογισμικό εξυπηρετητή μέχρι τα τέλη του 1990.

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML, έγινε στα τέλη του 1991 από τον Tim Berners-Lee με το όνομα ετικέτες της HTML. Η πρώτη προδιαγραφή, περιέγραφε τα 20 στοιχεία που αποτελούσαν την HTML, η οποία ήταν έντονα επηρεασμένη από την γλώσσα SGMLguid που ήταν προέκταση της SGML. Μετά από τέσσερα χρόνια, το 1994 αναπτύσσετε το πρότυπο HTML 2.0 το οποίο κυκλοφόρησε στις 24 Νοεμβρίου του 1995. Η επόμενη έκδοση HTML 3.2 κυκλοφόρησε τον Ιανουάριο του 1997 και δημοσιεύτηκε ως πλήρη σύσταση από τον οργανισμό πρωτοτύπων W3C. Από την έκδοση 3.2 αφαιρέθηκε η υποστήριξη μαθηματικών τύπων και οι υπερκαλύψεις που μπορούσαν να υλοποιηθούν από τρίτους φορείς(Microsoft, Netscape). Στη συνέχεια το Δεκέμβριο του 1997 δημοσιεύτηκε η έκδοση HTML 4.0 η οποία εισήγαγε πολλά στοιχεία και ιδιότητες που είχαν προτείνει οι εταιρίες Microsoft και Netscape. Η νέα έκδοση HTML 5.0 η οποία παραμένει μέχρι και σήμερα, κυκλοφόρησε στις 28 Οκτωβρίου του 2004 από τον οργανισμό W3C και παρέχει πολλά νέα συντακτικά χαρακτηριστικά. Η HTML 5.0 περιλαμβάνει νέα στοιχεία όπως το <audio>, <video> και <canvas> και υποστηρίζει μαθηματικές φόρμουλες και scalable vector graphics.^[78] Τα νέα αυτά χαρακτηριστικά, είναι σχεδιασμένα ώστε να παρέχουν έναν εύκολο τρόπο να εισάγουμε και να χειριστούμε τα πολυμεσικά στοιχεία και τα γραφικά περιεχομένου στο διαδίκτυο, χωρίς να χρειάζεται να χρησιμοποιήσουμε ιδιωτικά plugins και APIs.

3.11 Η γλώσσα φύλλων στυλ CSS

Η γλώσσα CSS(Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) χρησιμοποιείται για την μορφοποίηση της εμφάνισης ενός εγγράφου που έχει γραφτεί με μία γλώσσα σήμανσης(markup language). Ο ρόλος της έγκειται σε μία πλουσιότερη στιλιστική σχεδίαση μίας ιστοσελίδας έναντι της γλώσσας HTML, διαμορφώνοντας με περισσότερες δυνατότητες χαρακτηριστικά όπως το χρώμα, η στοίχιση, η γραμματοσειρά και πολλά άλλα. Σήμερα χρησιμοποιείτε ευρύτητα για



Εικόνα 3-36 : CSS Λογότυπο

την μορφοποίηση HTML και XHTML εγγράφων αλλά μπορεί επίσης να χρησιμοποιηθεί και για την μορφοποίηση XML, SVG και XUL εγγράφων.

Η γλώσσα CSS έχει σχεδιαστεί ώστε να επιτρέπει το διαχωρισμό του περιεχομένου ενός εγγράφου που έχει γραφτεί σε μια γλώσσα σήμανσης, από την παρουσίαση του και τον τρόπο εμφάνισής του. Έτσι η σύνταξη κανόνων CSS μπορεί να γίνει μέσα στο ίδιο έγγραφο που περιέχει το περιεχόμενο και τα δομικά στοιχεία της εκάστοτε γλώσσας σήμανσης ή σε εξωτερικό έγγραφο το οποίο φέρει την κατάληξη CSS. Η χρήση κανόνων CSS κατά κόρον γίνεται σε εξωτερικά έγγραφα για λόγους λειτουργικότητας και ευελιξίας, επιτρέποντας παράλληλα σε πολλά έγγραφα να μοιράζονται την ίδια μορφοποίηση. Επίσης η γλώσσα CSS επιτρέπει την παρουσίαση του ίδιου περιεχομένου με διαφορετικούς τρόπους όπως είναι η εκτύπωση, η εμφάνιση στην οθόνη, η εμφάνιση σε συσκευές ανάγνωσης που βασίζονται στο σύστημα Braille^[79] και άλλα πολλά.

3.11.1 Ιστορία της γλώσσας CSS

Η γλώσσα CSS προτάθηκε από τον Håkon Wium Lie στις 10 Οκτωβρίου του 1994, όπου εκείνη την περίοδο εργαζόταν με τον Tim Berners-Lee στο εργαστήριο φυσικής CERN. Η πρόταση του έγινε δεκτή από τον οργανισμό πρωτοτύπων W3C, ο οποίος συνέχισε την σχεδίαση της γλώσσας και το 1996 κυκλοφόρησε την πρώτη σύσταση για το CSS1. Μετά το CSS1 δημοσιεύτηκαν άλλες δυο εκδόσεις ή επίπεδα όπως ονομάζονται, οι οποίες έφεραν πολλές αλλαγές, εισάγοντας νέες δυνατότητες σχεδίασης και μορφοποίησης στην παρουσίαση των ιστοσελίδων. Το 1998 έγινε η σύσταση για το CSS επιπέδου 2 και το 1999 ακολουθείσαι η σύσταση για το CSS επιπέδου 3. Η συμβολή του οργανισμού W3C ξεκίνησε από το επίπεδο της CSS1 όπου και συνεχίζει την ανάπτυξή της μέχρι και σήμερα.

ΚΕΦΑΛΑΙΟ 4

Υλοποίηση συστήματος για online δημοπρασίες και πωλήσεις

Στο κεφάλαιο αυτό θα παρουσιάσουμε τα βήματα που ακολουθήσαμε κατά την διαδικασία υλοποίησης του συστήματος για online δημοπρασίες και πωλήσεις. Το σύστημα μας, έχει την μορφή ενός ηλεκτρονικού καταστήματος το οποίο υποστηρίζει τις εξής διαδικασίες, δυνατότητες:

- Απλούς χρήστες (users)
- Έναν χρήστη-διαχειριστή (administrator)
- Πώληση προϊόντων μέσω δημοπρασίας που εκτελείτε σε πραγματικό χρόνο
- Αναζήτηση δημοπρασιών
- Δημιουργία συγκεντρωτικών αναφορών(page reports) για τα αποτελέσματα των δημοπρασιών που έλαβαν τέλος

Δυνατότητες χρήστη(user):

- Εγγραφή χρήστη
- Συμμετοχή σε μια πώληση προϊόντος μέσω δημοπρασίας σε πραγματικό χρόνο
- Δυνατότητα αναζήτησης δημοπρασιών με βάση το όνομα του προϊόντος, την κατηγορία τον προϊόντος και την εταιρία στην οποία ανήκει το προϊόν

Δυνατότητες χρήστη-διαχειριστή(administrator):

- Εισαγωγή νέων προϊόντων
- Εισαγωγή νέων δημοπρασιών
- Επίβλεψη συγκεντρωτικών αναφορών(page reports)

Η υλοποίηση του ηλεκτρονικού καταστήματος χωρίζετε σε 2 μέρη:

1. Σύστημα αποθήκευσης δεδομένων:

Για την αποθήκευση της χρήσιμης πληροφορίας που συμβάλει στην ορθή διεξαγωγή και λειτουργία του συστήματος έγινε χρήση της τεχνολογίας XML. Ο συνδυασμός της γλώσσας XML με τις τεχνολογίες XML Schema και XSL που υποστηρίζει, προσφέρει ένα αξιόπιστο περιβάλλον αποθήκευσης, ανάκτησης και παρουσίασης πληροφορίας με δομημένο τρόπο.

2. Περιβάλλον και δυνατότητες συστήματος:

Για την δημιουργία του ηλεκτρονικού καταστήματος χρησιμοποιήθηκαν συνδυαστικά οι γλώσσες PHP, JavaScript, HTML,CSS και η τεχνική Ajax. Πιο αναλυτικά:

- Η γλώσσα PHP χρησιμοποιήθηκε για την δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο από την πλευρά του διακομιστή. Επίσης αποτελεί το βασικό μηχανισμό για την διαχείριση και την επεξεργασία των XML εγγράφων.
- Η γλώσσα JavaScript χρησιμοποιήθηκε για τη παραγωγή δυναμικού περιεχομένου και την προσθήκη διαδραστικότητας στις ιστοσελίδες από την πλευρά του πελάτη.

- Η τεχνική AJAX χρησιμοποιήθηκε για την ασυγχρόνιστη επικοινωνία του συστήματος μας με τον διακομιστή, επιτρέποντας την αποστολή και την ανάκτηση δεδομένων στο προσκήνιο, χωρίς να είναι απαραίτητη η ανανέωση ολόκληρης της ιστοσελίδας.
- Η γλώσσα HTML χρησιμοποιήθηκε για την δημιουργία των ιστοσελίδων, των φορμών εισαγωγής δεδομένων, και των πινάκων παρουσίασης δεδομένων.
- Η γλώσσα φύλλων στυλ CSS χρησιμοποιήθηκε για την μορφοποίηση και τον καθορισμό της εμφάνισης κάθε ιστοσελίδας.

Λόγο της μεγάλης έκτασης κώδικα που δημιουργήθηκε για την ολοκλήρωση του ηλεκτρονικού καταστήματος, η περιγραφή υλοποίησης του συστήματος θα επικεντρωθεί περισσότερο στα βασικά σημεία, εξηγώντας την λογική ανάπτυξης και σχεδιασμού.

Πριν ξεκινήσουμε να αναλύσουμε και να περιγράψουμε τα βήματα υλοποίησης του πρακτικού μέρους της πτυχιακής, θα ακολουθήσει μια σύντομη αναφορά στα εργαλεία που χρησιμοποιήθηκαν, για την συγγραφή του απαραίτητου κώδικα ανάπτυξης του συστήματος καθώς και το λογισμικό εξυπηρετητή που φιλοξενεί(hosting) το ηλεκτρονικό κατάστημα μας.

4.1 Εργαλεία υλοποίησης

Για την υλοποίηση του πρακτικού έργου της πτυχιακής εργασίας απαιτείται η επιλογή των εξής δύο βασικών εργαλείων:

1. Ο τοπικός διακομιστής(server) που θα χρησιμοποιηθεί
2. Το περιβάλλον ανάπτυξης στο οποίο θα δουλέψουμε

Ο τοπικός διακομιστής που επιλέχτηκε για τις δοκιμές του ηλεκτρονικού καταστήματος είναι ο WampServer και το περιβάλλον ανάπτυξης της εφαρμογής πάνω στο οποίο δουλέψαμε είναι το Adobe Dreamweaver.

4.1.1 WampServer

Ο διακομιστής WAMP(Windows, Apache, MySQL, PHP) δημιουργήθηκε από τον Romain Bourdon και αποτελεί ένα λογισμικό εξυπηρετητή για λειτουργικά συστήματα Microsoft Windows. Χρησιμοποιείτε συχνά για την ανάπτυξη εφαρμογών διαδικτύου και τοπικών δοκιμών, ωστόσο μπορεί να χρησιμοποιηθεί και για την εξυπηρέτηση online ιστοσελίδων.

Ο WampServer αποτελείται από τρία βασικά πακέτα:

1. Apache web server
2. MySQL Database
3. PHP



Εικόνα 4-1 :
WampServer Λογότυπο

Το πακέτο Apache web server χρησιμοποιείτε για την λειτουργία του web διακομιστή, παρέχοντας HTTP υπηρεσίες. Η λειτουργία του Apache web server σε τοπικό επίπεδο, Λειτουργώντας σε τοπικό επίπεδο τον Apache web server, ένας προγραμματιστής(web developer) μπορεί να δοκιμάζει τις ιστοσελίδες του σε έναν φυλλομετρητή ιστού(web browser) χωρίς να χρειάζεται να τις δημοσιεύσει στο διαδίκτυο. Τα πακέτα MySQL και PHP, αποτελούν δύο από τις πιο κοινές τεχνολογίες που χρησιμοποιούνται ευρύτατα για την δημιουργία δυναμικών ιστοσελίδων. Η

MySQL είναι μία υψηλής ταχύτητας βάση δεδομένων, ενώ η PHP είναι μια γλώσσα σεναρίων που χρησιμοποιείτε για την πρόσβαση στα δεδομένα της βάσης. Με την εγκατάσταση αυτών των δύο πακέτων σε τοπικό επίπεδο, ένας προγραμματιστής μπορεί να κατασκευάσει και να δοκιμάσει ιστοσελίδες με δυναμικό περιεχόμενο πριν από την δημοσίευσή τους σε ένα δημόσιο web διακομιστή.

Τα πακέτα Apache, MySQL και PHP είναι ανοιχτού κώδικα (open source) τα οποία μπορούν να εγκατασταθούν ανεξάρτητα. Παρόλα αυτά επιλέγοντας την εγκατάσταση του WampServer στον υπολογιστή μας, εγκαθιστά και τα 3 πακέτα. Η έκδοση του WampServer που χρησιμοποιήσαμε για την υλοποίηση του ηλεκτρονικού καταστήματος είναι η έκδοση 2.2 και περιλαμβάνει τις εκδόσεις Apache 2.2.22, PHP 5.3.13 και MySQL 5.5.24. Στην ενότητα παράρτημα στο τέλος της αναφοράς της πτυχιακής εργασίας, παρουσιάζετε ο οδηγός εγκατάστασης του WampServer καθώς και οι επεκτάσεις (extensions) που πρέπει να ενεργοποιήσουμε στο πακέτο PHP για την υποστήριξη επιπλέον δυνατοτήτων.

Εφόσον έχουμε κάνει επιτυχώς την εγκατάσταση του WampServer στον υπολογιστή μας, μπορούμε να τον τρέξουμε από το εικονίδιο εγκατάστασης του. Όταν το εικονίδιο του WampServer τα οποίο εμφανίζεται κάτω δεξιά στην μπάρα εργασιών των Windows γίνει από κόκκινο πράσινο τότε μπορούμε να έχουμε πρόσβαση στις ιστοσελίδες που δημιουργούμε πληκτρολογώντας “http://localhost/όνομα ιστοσελίδας” ή “http://127.0.0.1/όνομα ιστοσελίδας” στη γραμμή διευθύνσεων του φυλλομετρητή ιστού μας.

4.1.2 Adobe Dreamweaver

Το Adobe Dreamweaver είναι ένα εργαλείο ανάπτυξης και σχεδιασμού ιστοσελίδων που αναπτύχθηκε από την εταιρία Adobe System. Αρχικά δημιουργήθηκε από την εταιρία Macromedia το 1997 και στη συνέχεια εξαγοράστηκε από την Adobe System το 2005, η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη του λογισμικού. Μετά την εξαγορά του Dreamweaver από την σουίτα προγραμμάτων της Macromedia, η εταιρία Adobe System προχώρησε σε σημαντικές αλλαγές, καθιστώντας το πρόγραμμα Dreamweaver πιο συμβατό με τα πρότυπα του W3C. Οι πρόσφατες εκδόσεις έχουν βελτιωμένη υποστήριξη για Web τεχνολογίες όπως HTML, JavaScript, PHP, CSS και framework όπως ASP (ASP JavaScript, ASP, ASP VBScript, ASP.NET C#, ASP.NET VB), ColdFusion, JavaServer Pages, Scriptlet.



Εικόνα 4-2 : Adobe Dreamweaver Λογότυπο

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε η έκδοση Adobe Dreamweaver CC 6 (v12) για τον σχεδιασμό και την ανάπτυξη του ηλεκτρονικού καταστήματος. Η έκδοση CC 6 έκτος από την υποστήριξη των παραπάνω τεχνολογιών που προαναφέραμε, παρέχει και μια ολοκληρωμένη λύση για την συγγραφή εγγράφων XML, XML Schemas και XSLT. Επίσης υποστηρίζει την δυνατότητα ελέγχου εγκυρότητας (validation) ενός XML εγγράφου έναντι ενός XML σχήματος και παρέχει δυνατότητες συντακτικού ελέγχου σε πραγματικό χρόνο, επισήμανση κώδικα και ολοκλήρωση κώδικα (code completion).

4.2 Δημιουργία κατάλληλων εγγράφων XML

Σύμφωνα με τις απαιτήσεις υλοποίησης του συστήματος για online δημοπρασίες και πωλήσεις, η αποθήκευση της χρήσιμης πληροφορίας θα γίνεται σε έγγραφα XML. Για την υλοποίηση του κατάλληλου περιβάλλοντος αποθήκευσης σχετιζόμενων δεδομένων δημιουργήθηκαν τα εξής αρχεία XML:

- **Αρχείο products.xml**

Το έγγραφο products.xml περιγράφει το σύστημα αποθήκευσης των προϊόντων που θα λάβουν μέρος σε μία δημοπρασία και αποτελείται από τα εξής στοιχεία για την περιγραφή ενός προϊόντος:

- products_id
Το στοιχείο products_id χρησιμοποιείτε για να περιγράψει μονοσήμαντα ένα προϊόν, μέσω ενός μοναδικού αλφαριθμητικού αναγνωριστικού. Η ανάθεση τιμής σε ένα στοιχείο products_id γίνεται αυτόματα μέσω PHP κώδικα, σύμφωνα πάντα με την καθορισμένη σειρά των προϊόντων που αποθηκεύονται μέσα στο έγγραφο products.xml. Θα μπορούσαμε να πούμε ότι το products_id έχει το ρόλο ενός κλειδιού, όπως γίνεται και στο σχεδιασμό μίας βάσης δεδομένων.
- price
Το στοιχείο price χρησιμοποιείτε για την αποθήκευση της τιμής(ευρώ) του κάθε προϊόντος. Η τιμή αυτή αφορά την αρχική τιμή του προϊόντος για μια δημοπρασία.
- product_name
Το στοιχείο product_name χρησιμοποιείτε για την αποθήκευση του ονόματος του κάθε προϊόντος.
- category
Το στοιχείο category χρησιμοποιείτε για την αποθήκευση της κατηγορίας που ανήκει το κάθε προϊόν(π.χ. computers, cars, watches).
- corporation
Το στοιχείο corporation χρησιμοποιείτε για την αποθήκευση της εταιρίας που ανήκει το κάθε προϊόν.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- αρχείο Xml για τα προϊόντα της σελίδας -->
3 <products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="products.xsd">
5   <product>
6     <product_id>p1</product_id>
7     <price>345</price>
8     <product_name>Compaq 6530g</product_name>
9     <category>Computers</category>
10    <corporation>Hewlett Packard</corporation>
11  </product>
12  <product>
13    <product_id>p2</product_id>
14    <price>340</price>
15    <product_name>230g</product_name>
16    <category>Computers</category>
17    <corporation>Dell</corporation>
18  </product>
19  <product>
20    <product_id>p3</product_id>
21    <price>700</price>
22    <product_name>N300g</product_name>
23    <category>Computers</category>
24    <corporation>Dell</corporation>
25  </product>
26 </products>

```

Εικόνα 4-3 : Στιγμιότυπο του αρχείου products.xml

Το περιεχόμενο των στοιχείων price, product_name, category και corporation, το καθορίζει μόνο ο διαχειριστής του συστήματος μέσω της διαδικασίας εισαγωγή νέων προϊόντων(βλ. παρακάτω). Το

αρχείο products.xml βρίσκετε στο φάκελο store\connections, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο pelates.xml**

Το έγγραφο pelates.xml περιγράφει το σύστημα αποθήκευσης για τους χρήστες που έχουν κάνει εγγραφή(registered) στην ιστοσελίδα μας και αποτελείται από τα εξής στοιχεία:

- user_id
Το στοιχείο user_id χρησιμοποιείτε για να περιγράψει μονοσήμαντα ένα χρήστη, μέσω ενός μοναδικού ακέραιου αριθμού(αναγνωριστικό). Η ανάθεση τιμής σε ένα στοιχείο user_id γίνεται αυτόματα μέσω PHP κώδικα, σύμφωνα πάντα με την καθορισμένη σειρά εγγραφής των χρηστών.
- username
Το στοιχείο username χρησιμοποιείτε για την αποθήκευση ενός μοναδικού ονόματος το οποίο επιλέγει ο χρήστης για να μπορεί σε συνδυασμό με τον κωδικό που θα ορίσει να συνδεθεί(log in) με την ιστοσελίδα μας.
- password
Το στοιχείο password το καθορίζει ο χρήστης και χρησιμοποιείτε για την αποθήκευση του κωδικού χρήστη. Ο συνδυασμός του password με το username αποτελεί έναν ασφαλή τρόπο συνδέσεις(log in) ενός χρήστη με την ιστοσελίδα μας. Η αποθήκευση ενός κωδικού γίνεται σε κωδικοποιημένη μορφή MD5 κατά την έγγραφη ενός χρήστη και παράγετε αυτόματα μέσω κώδικα PHP.
- name
Το στοιχείο name χρησιμοποιείτε για την αποθήκευση του πραγματικού ονόματος ενός χρήστη. Το περιεχόμενο του στοιχείου name το καθορίζει ο χρήστης κατά την έγγραφη του στην ιστοσελίδα μας.
- address
Το στοιχείο address χρησιμοποιείτε για την αποθήκευση της διεύθυνσης κατοικίας ενός χρήστη. Το περιεχόμενο του στοιχείου address το καθορίζει ο χρήστης κατά την έγγραφη του στην ιστοσελίδα μας.
- phone
Το στοιχείο phone χρησιμοποιείτε για την αποθήκευση του προσωπικού τηλεφώνου ενός χρήστη. Το περιεχόμενο του στοιχείου phone το καθορίζει ο χρήστης κατά την έγγραφη του στην ιστοσελίδα μας.
- credit_card_number
Το στοιχείο credit_card_number χρησιμοποιείτε για την αποθήκευση του προσωπικού αριθμού πιστωτικής κάρτας ενός χρήστη. Το περιεχόμενο του στοιχείου credit_card_number το καθορίζει ο χρήστης κατά την έγγραφη του στην ιστοσελίδα μας.
- Type
Το στοιχείο type χρησιμοποιείτε για την περιγραφή των δικαιωμάτων που μπορεί να έχει ένας χρήστης στην ιστοσελίδα μας. Ο ακέραιος αριθμός “0” προσδιορίζει έναν απλό χρήστη και ο ακέραιος αριθμός “1” προσδιορίζει έναν χρήστη-διαχειριστή. Τα δικαιώματα ενός διαχειριστή αναφέρονται στην εισαγωγική ενότητα του τέταρτου κεφαλαίου. Το περιεχόμενο του στοιχείου type, καθορίζετε μέσω κώδικα PHP αυτόματα με την έγγραφη ενός χρήστη στην ιστοσελίδα μας.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- αρχείο Xml με τους εγγεγραμμένους χρήστες της σελίδα μας -->
3 <pelates xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="pelates.xsd">
5   <pelaths>
6     <user_id>1</user_id>
7     <username>angel</username>
8     <password>e10adc3949ba59abbe56e057f20f883e</password>
9     <name>angel</name>
10    <address>Oxford 12</address>
11    <phone>6946354671</phone>
12    <credit_card_number>A0147582</credit_card_number>
13    <type>0</type>
14  </pelaths>
15  <pelaths>
16    <user_id>2</user_id>
17    <username>sofia</username>
18    <password>fcea920f7412b5da7be0cf42b8c93759</password>
19    <name>sofia</name>
20    <address>newstreet 15</address>
21    <phone>6946353765</phone>
22    <credit_card_number>035748E</credit_card_number>
23    <type>1</type>
24  </pelaths>
25 </pelates>

```

Εικόνα 4-4 : Στιγμιότυπο του αρχείου pelates.xml

Το αρχείο pelates.xml βρίσκετε στο φάκελο store\users, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο auctions.xml**

Το έγγραφο auctions.xml περιγράφει το σύστημα αποθήκευσης για τις δημοπρασίες που πρόκειται να λάβουν μέρος στο μέλλον και αποτελείται από τα εξής στοιχεία:

- auction id
 Το στοιχείο auction_id χρησιμοποιείτε για να περιγράψει μονοσήμαντα μία δημοπρασία, μέσω ενός μοναδικού αλφαριθμητικού αναγνωριστικού. Η ανάθεση τιμής σε ένα στοιχείο auction_id, γίνεται αυτόματα μέσω PHP κώδικα κατά την διαδικασία εισαγωγής μιας νέας δημοπρασίας από τον διαχειριστή του συστήματος (βλ. παρακάτω). Η τιμή που λαμβάνει το στοιχείο auction_id συνδέεται απολυτά με την τιμή του στοιχείου product_id στο αρχείο products.xml, παρουσιάζοντας έτσι την σχέση δυο κόμβων σε δύο διαφορετικά έγγραφα XML. Για παράδειγμα η τιμή “p1” στο στοιχείο auction_id της εικόνας 4-5(βλ. παρακάτω) συνδέει το αρχείο auctions.xml με το αρχείο products.xml μέσω του αντίστοιχου product_id με τιμή “p1”. Αυτή η σύνδεση αποτελεί μια λογική σχέση μεταξύ των δυο εγγράφων XML και υλοποιείτε στο σύστημα μας μέσω κατάλληλου κώδικα PHP για την διαχείρισης των XML εγγράφων.
- time_start
 Το στοιχείο time_start χρησιμοποιείτε για την αποθήκευση της ώρας έναρξης μιας δημοπρασίας και καθορίζεται από τον διαχειριστή του συστήματος κατά την διαδικασία εισαγωγής μίας νέας δημοπρασίας.

- start_price
Το στοιχείο `start_price` χρησιμοποιείτε για την αποθήκευση της αρχικής τιμής ενός προϊόντος σε μία δημοπρασία και καθορίζεται αυτόματα μέσω κώδικα PHP. Η ανάθεση τιμής του στοιχείου `start_price` γίνεται μέσω της επιλογής προϊόντος(drop down list) κατά την διαδικασία εισαγωγής μίας νέας δημοπρασίας. Το προϊόν που θα επιλέξει ο διαχειριστής αντιστοιχεί στο προϊόν του αρχείου `products.xml`, μέσω της μοναδικής αμφίδρομης σχέσης που έχουν τα στοιχεία `auction_id` και `product_id` στα αρχεία `auctions.xml` και `products.xml` αντίστοιχα. Ίσως σε αυτό το σημείο να μπορούμε να ονομάσουμε αυτά τα στοιχεία, στοιχεία-κλειδιά για να περιγράψουμε τον ρόλο τους στη σύνδεση δύο ή περισσότερων XML εγγράφων.
- auction_status
Το στοιχείο `auction_status` χρησιμοποιείτε για την αποθήκευση της κατάστασης που βρίσκετε μία δημοπρασία. Η τιμές κατάστασης που μπορεί να πάρει είναι “Not Start” και “Finished” και καθορίζονται αυτόματα μέσω κώδικα PHP κατά την διαδικασία εισαγωγής μίας νέας δημοπρασίας και στη λήξη μιας δημοπρασίας. Εάν το στοιχείο `auction_status` έχει την τιμή “Not Start” τότε μπορεί να εμφανίζετε στον πίνακα με της δημοπρασίες που πρόκειται να λάβουν μέρος στο μέλλον, εάν όμως έχει την τιμή “Finished” τότε δεν θα εμφανίζετε πλέον στον πίνακα των δημοπρασιών(βλ. παρακάτω). Επίσης το στοιχείο `auction_status` θα μπορούσε να χρησιμοποιηθεί για μελλοντικές επεκτάσεις του συστήματος για online δημοπρασίες και πωλήσεις.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- αρχείο Xml για τις δημοπρασίες που θα πάρουν μέρος -->
3 <auctions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="auctions.xsd">
5   <auction>
6     <auction_id>p1</auction_id>
7     <time_start>T18:43 23-07-2015</time_start>
8     <start_price>345</start_price>
9     <auction_status>Not Start</auction_status>
10  </auction>
11  <auction>
12    <auction_id>p2</auction_id>
13    <time_start>T23:34 26-09-2015</time_start>
14    <start_price>340</start_price>
15    <auction_status>Not Start</auction_status>
16  </auction>
17  <auction>
18    <auction_id>p3</auction_id>
19    <time_start>T23:00 26-07-2015</time_start>
20    <start_price>890</start_price>
21    <auction_status>Finished</auction_status>
22  </auction>
23 </auctions>

```

Εικόνα 4-5 : Στιγμιότυπο του αρχείου `auctions.xml`

Το αρχείο `auctions.xml` βρίσκεται στο φάκελο `store\connections`, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο `offers.xml`**

Το έγγραφο `offers.xml` περιγράφει το σύστημα αποθήκευσης για τις προσφορές των χρηστών κατά την διαδικασία μίας δημοπρασίας και αποτελείται από τα εξής στοιχεία:

- offer_id
Το στοιχείο `offer_id` χρησιμοποιείτε για να περιγράψει μονοσήμαντα μία προσφορά χρήστη, μέσω ενός μοναδικού αλφαριθμητικού αναγνωριστικού. Η ανάθεση τιμής στο στοιχείο `offer_id`, γίνεται αυτόματα μέσω PHP κώδικα κατά την υποβολή μιας προσφοράς από ένα χρήστη.
- id_product
Το στοιχείο `id_product` χρησιμοποιείτε για την αποθήκευση του μοναδικού αλφαριθμητικού αναγνωριστικού ενός συγκεκριμένου προϊόντος από το αρχείο `products.xml`. Η ανάθεση τιμής στο στοιχείο `id_product` γίνεται αυτόματα μέσω κώδικα PHP κατά την υποβολή μιας προσφοράς χρήστη σε μία δημοπρασία. Με αυτό τον τρόπο συνδέουμε την προσφορά ενός χρήστη με το αντίστοιχο προϊόν που αναφέρεται και παρέχουμε μέσω του στοιχείου αυτού πρόσβαση στα υπόλοιπα στοιχεία ενός συγκεκριμένου προϊόντος στο αρχείο `products.xml`.
- id_user
Το στοιχείο `id_user` χρησιμοποιείτε για την αποθήκευση του μοναδικού αναγνωριστικού ενός συγκεκριμένου χρήστη από το αρχείο `relates.xml`. Η ανάθεση τιμής στο στοιχείο `id_user` γίνεται αυτόματα μέσω κώδικα PHP κατά την υποβολή μιας προσφοράς χρήστη σε μία δημοπρασία. Με αυτό τον τρόπο συνδέουμε μία προσφορά με τον αντίστοιχο χρήστη που την υπέβαλε και παρέχουμε μέσω του στοιχείου αυτού πρόσβαση στα υπόλοιπα στοιχεία ενός συγκεκριμένου χρήστη στο αρχείο `relates.xml`.
- offer_time
Το στοιχείο `offer_time` χρησιμοποιείτε για την αποθήκευση της ώρας που έγινε μία προσφορά από ένα χρήστη κατά την υποβολή μιας προσφοράς χρήστη σε μία δημοπρασία.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- αρχείο Xml για τις προσφορές που γίνονται σε μια δημοπρασία -->
3 <offers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="offers.xsd">
5   <offer>
6     <offer_id>o1</offer_id>
7     <price>650</price>
8     <id_product>p11</id_product>
9     <id_user>4</id_user>
10    <offer_time>2015-09-26 16:25:12</offer_time>
11  </offer>
12  <offer>
13    <offer_id>o2</offer_id>
14    <price>490</price>
15    <id_product>p13</id_product>
16    <id_user>5</id_user>
17    <offer_time>2015-09-26 22:00:37</offer_time>
18  </offer>
19 </offers>

```

Εικόνα 4-6 : Στιγμιότυπο του αρχείου `offers.xml`

Το αρχείο `offers.xml` βρίσκεται στο φάκελο `store\connections`, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο finished_auctions.xml**

Το έγγραφο finished_auctions.xml περιγράφει το σύστημα αποθήκευσης για τις δημοπρασίες που έλαβαν τέλος και αποτελείται από τα εξής στοιχεία:

- product_id
Το στοιχείο product_id χρησιμοποιείτε για την αποθήκευση του μοναδικού αλφαριθμητικού αναγνωριστικού ενός συγκεκριμένου προϊόντος από το αρχείο products.xml. Η ανάθεση τιμής στο στοιχείο product_id γίνεται αυτόματα μέσω PHP κώδικα με την λήξη μίας δημοπρασίας. Με αυτό τον τρόπο συνδέουμε μία συγκεκριμένη δημοπρασία που έλαβε τέλος με το αντίστοιχο προϊόν στο οποίο αναφέρεται.
- user_id
Το στοιχείο user_id χρησιμοποιείτε για την αποθήκευση του μοναδικού αναγνωριστικού ενός συγκεκριμένου χρήστη από το αρχείο pelates.xml. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας. Στην περίπτωση που ένας χρήστης έχει κερδίσει μια δημοπρασία η τιμή του στοιχείου user_id θα είναι ίση με το μοναδικό αναγνωριστικό του ενώ αν η δημοπρασία έχει λήξει χωρίς κανένα νικητή(άγονη-barren) τότε το στοιχείο user_id θα είναι κενό.
- offer_id
Το στοιχείο offer_id χρησιμοποιείτε για την αποθήκευση του μοναδικού αναγνωριστικού μίας συγκεκριμένης προσφοράς χρήστη από το αρχείο offers.xml. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας. Στην περίπτωση που ένας χρήστης έχει κερδίσει μια δημοπρασία η τιμή του στοιχείου offer_id θα είναι ίση με το μοναδικό αναγνωριστικό της προσφοράς που έκανε ο συγκεκριμένος χρήστης ενώ αν η δημοπρασία λήξει χωρίς κανένα νικητή τότε το στοιχείο offer_id θα είναι κενό.
- price
Το στοιχείο price χρησιμοποιείτε, για την αποθήκευση της τιμής προσφοράς του χρήστη ο οποίος κέρδισε σε μία συγκεκριμένη δημοπρασία ή για την αποθήκευση της αρχικής τιμής προϊόντος στην περίπτωση που μια δημοπρασία λήξει άγονη. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας. Στην περίπτωση που ένας χρήστης έχει κερδίσει μια συγκεκριμένη δημοπρασία η τιμή του στοιχείου price θα είναι ίση με την τιμή της προσφοράς του χρηστή από το αρχείο offers.xml, ενώ αν η δημοπρασία λήξει άγονη τότε η τιμή του στοιχείου price θα είναι ίση με τιμή του προϊόντος ενός συγκεκριμένου στοιχείου price από το αρχείου products.xml.
- time_ended
Το στοιχείο time_ended χρησιμοποιείτε για την αποθήκευση της ώρας λήξης μίας συγκεκριμένης δημοπρασίας. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας.
- status
Το στοιχείο status χρησιμοποιείτε για την αποθήκευση της κατάστασης στην οποία έληξε μια δημοπρασία. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας. Στην περίπτωση που μία δημοπρασία λήξει άγονη, η τιμή του στοιχείου status θα είναι ίση με το αλφαριθμητικό “Barren” ενώ αν μία δημοπρασία λήξει με κάποιο νικητή τότε η τιμή του στοιχείου status θα είναι ίση με το αλφαριθμητικό “Knock Down”.

- delivered

Το στοιχείο delivered χρησιμοποιείται για την αποθήκευση της κατάστασης παράδοσης του προϊόντος μια δημοπρασίας που έλαβε τέλος. Η ανάθεση τιμής γίνεται αυτόματα μέσω PHP κώδικα μετά την λήξη μίας δημοπρασίας. Στην περίπτωση που μία δημοπρασία λήξει άγονη, η τιμή του στοιχείου delivered είναι κενή ενώ αν μία δημοπρασία λήξει με κάποιο νικητή τότε η τιμή του στοιχείου delivered θα είναι ίση με το αλφαριθμητικό “Yes” στην περίπτωση που το προϊόν παραδόθηκε στο χρήστη και “No” στην περίπτωση που δεν έχει παραδοθεί στο χρήστη.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- αρχείο XML για τις δημοπρασίες που έληξαν-->
3 <finished_auctions
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5   <finished_auction>
6     <product_id>p11</product_id>
7     <user_id>4</user_id>
8     <offer_id>o1</offer_id>
9     <price>650</price>
10    <time_ended>2015-09-26 16:40:00</time_ended>
11    <status>Knock Down</status>
12    <delivered>No</delivered>
13  </finished_auction>
14  <finished_auction>
15    <product_id>p12</product_id>
16    <user_id/>
17    <offer_id/>
18    <price>40000</price>
19    <time_ended>2015-09-26 21:49:00</time_ended>
20    <status>Barren</status>
21    <delivered/>
22  </finished_auction>
23 </finished_auctions>
```

Εικόνα 4-7 : Στιγμιότυπο του αρχείου finished_auctions.xml

Το αρχείο finished_auctions.xml βρίσκεται στο φάκελο store\connections, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο auction_products.xml**

Το έγγραφο auction_products.xml περιγράφει το σύστημα αποθήκευσης για τις δημοπρασίες που πρόκειται να λάβουν μέρος στο μέλλον. Παρότι φαίνεται να έχει την ίδια λειτουργία με το αρχείο auctions.xml διαφοροποιείται από αυτό για πολλούς λόγους. Τα στοιχεία του εγγράφου auction_products.xml λαμβάνουν το περιεχόμενο τους μέσω PHP κώδικα, συλλέγοντας δεδομένα από τα στοιχεία των εγγράφων products.xml και auctions.xml. Η λειτουργία του εγγράφου αυτού έγκειται στην συλλογή όλων των απαραίτητων στοιχείων που χρειαζόμαστε για να μπορέσουμε να δημιουργήσουμε ένα αντίστοιχο HTML έγγραφο, το οποίο θα παρουσιάζει ένα πίνακα(εικόνα 4-8) με τις δημοπρασίες που δεν έχουν ξεκινήσει ακόμα. Εφόσον έχουμε συγκεντρώσει όλη την απαραίτητη πληροφορία που θέλουμε μέσα στο έγγραφο auction_products.xml, τότε μπορούμε να το μετασχηματίσουμε σε ένα HTML έγγραφο, μέσω του κατάλληλου εγγράφου XSLT. Όλη η διαδικασία του μετασχηματισμού ενός εγγράφου XML, υλοποιείται μέσω προκαθορισμένων συναρτήσεων PHP κώδικα.

Το έγγραφο auction_products.xml αποτελείται από τα εξής στοιχεία:

- count_products
Το στοιχείο count_products χρησιμοποιείτε για την αποθήκευση ενός μοναδικού αναγνωριστικού για κάθε προϊόν που πρόκειται να λάβει μέρος σε μία δημοπρασία. Η λογική υλοποιήσεις του στοιχείου count_products έγκειται στην ποσοτική μέτρηση των προϊόντων αυτών. Με αυτό τον τρόπο κατά το μετασχηματισμό του εγγράφου auction_products.xml σε ένα HTML έγγραφο, θα μπορούμε να έχουμε μία στήλη που αριθμεί τα προϊόντα μέσα σε ένα HTML πινάκα(εικόνα 4-8, στήλη A/A). Η ανάθεση τιμής στο στοιχείο count_products γίνεται αυτόματα μέσω PHP κώδικα κατά την φόρτωση της HTML σελίδας auctions_display.php.
- product_name
Το στοιχείο product_name χρησιμοποιείτε για την αποθήκευση του ονόματος ενός συγκεκριμένου προϊόντος που πρόκειται να λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο product_name γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(product_name) από το έγγραφο products.xml κατά την φόρτωση της HTML σελίδα auctions_display.php.
- price
Το στοιχείο price χρησιμοποιείτε για την αποθήκευση της αρχικής τιμής ενός προϊόντος ή της μεταγενέστερης τιμής του εφόσον η δημοπρασία ενός προϊόντος έχει ξεκινήσει. Η ανάθεση τιμής στο στοιχείο price γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(start_price) από το έγγραφο auctions.xml κατά την φόρτωση της HTML σελίδα auctions_display.php.
- time_start
Το στοιχείο time_start χρησιμοποιείτε για την αποθήκευση της ώρας έναρξης μίας δημοπρασίας. Η ανάθεση τιμής στο στοιχείο time_start γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(time_start) από το έγγραφο auctions.xml κατά την φόρτωση της HTML σελίδα auctions_display.php.
- corporation
Το στοιχείο corporation χρησιμοποιείτε για την αποθήκευση της εταιρίας που ανήκει το κάθε προϊόν που θα λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο corporation γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(corporation) από το έγγραφο products.xml κατά την φόρτωση της HTML σελίδα auctions_display.php.
- category
Το στοιχείο category χρησιμοποιείτε για την αποθήκευση της κατηγορίας που ανήκει το κάθε προϊόν που θα λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο category γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(category) από το έγγραφο products.xml κατά την φόρτωση της HTML σελίδα auctions_display.php.
- product_id
Το στοιχείο product_id χρησιμοποιείτε για την αποθήκευση του μοναδικού αλφαριθμητικού αναγνωριστικού ενός συγκεκριμένου προϊόντος από το αρχείο products.xml. Η ανάθεση τιμής στο στοιχείο product_id γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(product_id) από το έγγραφο products.xml κατά την φόρτωση της HTML σελίδα auctions_display.php. Η τιμή του συγκεκριμένου στοιχείου δεν εμφανίζεται στον πίνακα με τις δημοπρασίες

και χρησιμοποιείτε για να έχουμε πρόσβαση στα στοιχεία του αντίστοιχου κόμβου με το ίδιο product_id από το αρχείο products.xml.

- auction_status

Το στοιχείο auction_status χρησιμοποιείτε για την αποθήκευση της κατάστασης που βρίσκετε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο auction_status γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(auction_status) από το έγγραφο auctions.xml κατά την φόρτωση της σελίδα auctions_display.php. Σε αυτό το σημείο πρέπει να αναφέρουμε ότι οι κόμβοι των οποίων το στοιχείο auction_status έχει τιμή ίση με “Finished” στο έγγραφο auctions.xml, δεν πρόκειται να αποθηκευτούν στο auction_products.xml λόγω της λειτουργίας που θέλουμε να έχει το συγκεκριμένο XML αρχείο. Με αυτό τον τρόπο ο πίνακας δημοπρασιών μένει ανανεωμένος πάντα με τις δημοπρασίες που δεν έχουν ξεκινήσει ακόμα. Με την συγκεκριμένη υλοποίηση του συστήματος μας το στοιχείο auction_status παίρνει προκαθορισμένες τιμές που καθορίζει το αντίστοιχο στοιχείο auction_status από το αρχείο auctions.xml. Ο σκοπός δημιουργίας ενός ίδιου στοιχείου στο έγγραφο auction_products.xml έγκειται σε μελλοντικές επεκτάσεις του συστήματος και θα μπορούσε να χρησιμοποιηθεί στη περιγραφή της κατάστασης μιας δημοπρασίας η οποία βρίσκεται σε εξέλιξη.

A/A	Product Name	Price(€)	Start Time	Corpration	Category
1	A1	3063	T17:23 27-09-2015	Audi	Cars
2	L200	10700	T15:01 20-10-2015	Mitsubishi	Cars
3	L400	15700	T20:00 01-02-2013	Mitsubishi	Cars
4	Compaq 6530g	345	T18:43 23-07-2015	Hewlett Packard	Computers
5	230g	340	T23:34 26-09-2015	Dell	Computers
6	N300g	700	T21:15 13-11-2015	Dell	Computers
7	N400g	498.50	T20:43 10-10-2015	Dell	Computers
8	G-Shock XL	130.60	T20:24 27-12-2015	Casio	Watches
9	New Melenioum	100	T20:00 18-10-2015	Casio	Watches
10	Vogue City	350.60	T16:00 19-09-2015	Casio	Watches

Εικόνα 4-8 : Στιγμιότυπο XSLT μετασχηματισμού του εγγράφου auction_products.xml

Το στιγμιότυπο της εικόνας 4-8 παρουσιάζει ένα HTML πίνακα, του οποίου η κάθε μία στήλη εμφανίζει το περιεχόμενο των αντίστοιχων στοιχείων από το αρχείο auction_products.xml, μετά τον μετασχηματισμό του με το αντίστοιχο XSLT αρχείο. Το περιεχόμενο του πίνακα μεταβάλετε κάθε φορά που ένας χρήστης ζητά να δει την σελίδα auctions_display.php, κρατώντας έτσι ενημερωμένη την ιστοσελίδα με τις δημοπρασίες που πρόκειται να λάβουν μέρος στο μέλλον.

Στη συνέχεια ακολουθεί ένα στιγμιότυπο με τα στοιχεία και τα περιεχόμενα τους από το αρχείο auction_products.xml.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <auction_products>
3   <product>
4     <count_products>1</count_products>
5     <product_name>Compaq 6530g</product_name>
6     <price>345</price>
7     <time_start>T18:43 23-07-2015</time_start>
8     <corporation>Hewlett Packard</corporation>
9     <category>Computers</category>
10    <product_id>p1</product_id>
11    <auction_status>Not Start</auction_status>
12  </product>
13  <product>
14    <count_products>2</count_products>
15    <product_name>230g</product_name>
16    <price>340</price>
17    <time_start>T23:34 26-09-2015</time_start>
18    <corporation>Dell</corporation>
19    <category>Computers</category>
20    <product_id>p2</product_id>
21    <auction_status>Not Start</auction_status>
22  </product>
23 </auction_products>

```

Εικόνα 4-9 : Στιγμιότυπο του αρχείου auction_products.xml

Το αρχείο auction_products.xml βρίσκετε στο φάκελο store\ offers, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο search.xml**

Το έγγραφο search.xml περιγράφει το σύστημα αποθήκευσης για την διαδικασία αναζήτησης δημοπρασιών που πρόκειται να λάβουν μέρος στο μέλλον. Σε κάθε αναζήτηση δημοπρασίας το αρχείο search.xml ενημερώνετε αυτόματα με τα κατάλληλα κριτήρια που έχει επιλέξει ο χρήστης. Σύμφωνα με τις απαιτήσεις του συστήματος μας, ένας χρήστης θα μπορεί να κάνει αναζήτηση δημοπρασιών, ανά κατηγορία, ανά εταιρία και ανά συγκεκριμένο προϊόν. Το έγγραφο search.xml αποτελείται από τα εξής στοιχεία:

- select_search
Το στοιχείο select_search χρησιμοποιείτε για να αποθηκεύσουμε τα προκαθορισμένα κριτήρια αναζήτησης που μπορεί να επιλέξει ένας χρήστης. Τα κριτήρια αυτά περιλαμβάνουν την δυνατότητα αναζήτησης ανά προϊόν, ανά κατηγορία και ανά εταιρία(εικόνα 4-10). Η ανάθεση τιμής στο στοιχείο select_search γίνεται αυτόματα μέσω PHP κώδικα, ανάλογα με τα κριτήρια αναζήτησης δημοπρασιών που έχει επιλέξει ο χρήστης.
- search_text
Το στοιχείο search_text χρησιμοποιείτε για την αποθήκευση των κριτηρίων αναζήτησης που μπορεί να θέσει ένας χρήστης πληκτρολογώντας το ανάλογο κείμενο. Τα κριτήρια αυτά μπορεί να αναφέρονται σε μια συγκεκριμένη εταιρία, ένα συγκεκριμένο προϊόν ή μια συγκεκριμένη κατηγορία προϊόντων. Η ανάθεση τιμής στο στοιχείο search_text γίνεται αυτόματα μέσω PHP κώδικα, ανάλογα με τα κριτήρια αναζήτησης δημοπρασιών που έχει πληκτρολογήσει ο χρήστης.

Εικόνα 4-10 : Κριτήρια αναζήτησης δημοπρασιών

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <search_form>
3   <user_search>
4     <select_search>Corporation</select_search>
5     <search_text>dell</search_text>
6   </user_search>
7 </search_form>

```

Εικόνα 4-11 : Στιγμιότυπο του αρχείου search.xml

Το αρχείο search.xml βρίσκεται στο φάκελο store\search, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο search_results.xml**

Το έγγραφο search_results.xml περιγράφει το σύστημα αποθήκευσης με τα αποτελέσματα μίας αναζήτησης δημοπρασιών. Σε κάθε νέα αναζήτηση δημοπρασίας τα στοιχεία ενημερώνονται αυτόματα με τα αποτελέσματα των κριτηρίων αναζήτησης του αρχείου search.xml και παρουσιάζονται στο χρήστη μέσω του κατάλληλου XSLT μετασχηματισμού(εικόνα 4-15, 4-16, 4-17). Το έγγραφο search_results.xml αποτελείται από τα εξής στοιχεία:

- **Πρώτη περίπτωση:**
 Η αναζήτηση δημοπρασίας με βάση την εταιρία παραγωγής, δημιουργεί τα εξής στοιχεία μέσα στο έγγραφο search_results.xml:
 - **corporation**
 Το στοιχείο corporation χρησιμοποιείτε για την αποθήκευση της εταιρίας που ανήκει το κάθε προϊόν που θα λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο corporation γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(corporation) από το έγγραφο products.xml κατά την ενέργεια αναζήτησης ενός χρήστη.

- product_name
Το στοιχείο product_name χρησιμοποιείτε για την αποθήκευση του ονόματος ενός συγκεκριμένου προϊόντος που πρόκειται να λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο product_name γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(product_name) από το έγγραφο products.xml κατά την ενέργεια αναζήτησης ενός χρήστη.
- time_start
Το στοιχείο time_start χρησιμοποιείτε για την αποθήκευση της ώρας έναρξης μίας δημοπρασίας. Η ανάθεση τιμής στο στοιχείο time_start γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(time_start) από το έγγραφο auctions.xml κατά την ενέργεια αναζήτησης ενός χρήστη.
- start_price
Το στοιχείο start_price χρησιμοποιείτε για την αποθήκευση της αρχικής τιμής ενός προϊόντος ή της μεταγενέστερης τιμής του εφόσον η δημοπρασία ενός προϊόντος έχει ξεκινήσει. Η ανάθεση τιμής στο στοιχείο start_price γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(start_price) από το έγγραφο auctions.xml κατά την ενέργεια αναζήτησης ενός χρήστη.

➤ Δεύτερη περίπτωση:

Η αναζήτηση δημοπρασίας με βάση την κατηγορία προϊόντος δημιουργεί τα εξής στοιχεία μέσα στο έγγραφο search_results.xml:

- product_name
- time_start
- start_price
- category
Το στοιχείο category χρησιμοποιείτε για την αποθήκευση της κατηγορίας που ανήκει το κάθε προϊόν που θα λάβει μέρος σε μία δημοπρασία. Η ανάθεση τιμής στο στοιχείο category γίνεται αυτόματα μέσω PHP κώδικα που αποθηκεύει το περιεχόμενο του κατάλληλου στοιχείου(category) από το έγγραφο products.xml, κατά την ενέργεια αναζήτησης ενός χρήστη.

➤ Τρίτη περίπτωση:

Η αναζήτηση δημοπρασίας με βάση ένα συγκεκριμένο, δημιουργεί τα εξής στοιχεία μέσα στο έγγραφο search_results.xml:

- product_name
- time_start
- start_price

Το στιγμιότυπο του αρχείου search_results.xml σύμφωνα με την κάθε περίπτωση αναζήτησης δημοπρασίας, αποτελείται από τα εξής στοιχεία:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <search_results>
3   <search_data>
4     <corporation>Dell</corporation>
5     <product_name>230g</product_name>
6     <time_start>T23:34 26-09-2015</time_start>
7     <start_price>340</start_price>
8   </search_data>
9   <search_data>
10    <corporation>Dell</corporation>
11    <product_name>N300g</product_name>
12    <time_start>T21:15 13-11-2015</time_start>
13    <start_price>700</start_price>
14  </search_data>
15  <search_data>
16    <corporation>Dell</corporation>
17    <product_name>N400g</product_name>
18    <time_start>T20:43 10-10-2015</time_start>
19    <start_price>498.50</start_price>
20  </search_data>
21 </search_results>

```

Εικόνα 4-12 : Αποτελέσματα αναζήτησης με βάση την εταιρία προϊόντος

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <search_results>
3   <search_data>
4     <category>Watches</category>
5     <corporation>Casio</corporation>
6     <product_name>G-Shock XL</product_name>
7     <time_start>T20:24 27-12-2015</time_start>
8     <start_price>130.60</start_price>
9   </search_data>
10  <search_data>
11    <category>Watches</category>
12    <corporation>Casio</corporation>
13    <product_name>New Melenioum</product_name>
14    <time_start>T20:00 18-10-2015</time_start>
15    <start_price>100</start_price>
16  </search_data>
17  <search_data>
18    <category>Watches</category>
19    <corporation>Casio</corporation>
20    <product_name>Vogue City</product_name>
21    <time_start>T16:00 19-09-2015</time_start>
22    <start_price>350.60</start_price>
23  </search_data>
24 </search_results>

```

Εικόνα 4-13 : Αποτελέσματα αναζήτησης με βάση την κατηγορία προϊόντος

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <search_results>
3   <search_data>
4     <product_name>Compaq 6530g</product_name>
5     <time_start>T18:43 23-07-2015</time_start>
6     <start_price>345</start_price>
7   </search_data>
8 </search_results>

```

Εικόνα 4-14 : Αποτελέσματα αναζήτησης με βάση ένα συγκεκριμένο προϊόν

Η διαφοροποίηση των στοιχείων που μπορούν να δημιουργηθούν στο έγγραφο search_results.xml έγκειται στην λογική ανάπτυξη της διαδικασίας αναζήτησης δημοπρασιών και υλοποιείτε με το κατάλληλο XSLT μετασχηματισμό του εγγράφου search_results.xml (βλ. παρακάτω). Στην περίπτωση που ο χρήστης ψάξει ένα προϊόν με βάση την εταιρία παράγωγης, παραλείπουμε από την εμφάνιση των αποτελεσμάτων αναζήτησης την κατηγορία προϊόντος, λόγω του ότι ο χρήστης έχει ήδη γνώση της κατηγορία στην οποία ψάχνει. Για παράδειγμα αν ψάχνουμε μια εταιρία παραγωγής υπολογιστών όπως είναι η Dell, είναι λογικό να γνωρίζουμε ήδη την κατηγορία στην οποία ψάχνουμε. Στην αντίστοιχη περίπτωση που ο χρήστης ψάξει ένα προϊόν με βάση την κατηγορία προϊόντος, δεν παραλείπουμε από την εμφάνιση των αποτελεσμάτων αναζήτησης την εταιρία και αυτό γιατί μέσα στα αποτελέσματα της αναζήτησης του χρήστη μπορεί να υπάρχουν διάφορες εταιρίες προϊόντων. Στην περίπτωση τώρα που ο χρήστης κάνει μία αναζήτηση δημοπρασίας με βάση το όνομα του προϊόντος, από την εμφάνιση των αποτελεσμάτων αναζήτησης παραλείπουμε την εταιρία παραγωγής και την κατηγορία προϊόντος και αυτό γιατί είναι λογικό ο χρήστης να έχει γνώση των δυο αυτών στοιχείων εκ των προτέρων. Το αρχείο search_results.xml βρίσκεται στο φάκελο store\search, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

A/A	Corporation	Product Name	Time Start	Start Price(€)
1	Dell	230g	T23:34 26-09-2015	340
2	Dell	N300g	T21:15 13-11-2015	700
3	Dell	N400g	T20:43 10-10-2015	498.50

Εικόνα 4-15 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση την εταιρία προϊόντος

A/A	Category	Corporation	Product Name	Time Start	Start Price(€)
1	Cars	Audi	A1	T17:23 27-09-2015	3063
2	Cars	Mitsubishi	L200	T15:01 20-10-2015	10700
3	Cars	Mitsubishi	L400	T20:00 01-02-2013	15700
4	Cars	BMW	M800	T21:39 26-09-2015	40000

Εικόνα 4-16 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση την κατηγορία προϊόντος

A/A	Product Name	Time Start	Start Price(€)
1	Compaq 6530g	T18:43 23-07-2015	345

Εικόνα 4-17 : Εμφάνιση αποτελεσμάτων αναζήτησης με βάση ένα συγκεκριμένο προϊόν

- **Αρχείο report_barren.xml**

Το έγγραφο report_barren.xml περιγράφει το σύστημα αποθήκευσης για τις δημοπρασίες που έληξαν άγονες. Το συγκεκριμένο έγγραφο χρησιμοποιείτε αποκλειστικά για την αναφορά των άγονων δημοπρασιών συλλέγοντας το περιεχόμενο από τα κατάλληλα στοιχεία των αρχείων finished_auctions.xml και products.xml. Η ανάθεση τιμής στα στοιχεία του εγγράφου report_barren.xml γίνεται αυτόματα μέσω PHP κώδικα ο οποίος εκτελείτε σε κάθε αίτηση(Full report for barren auctions) του διαχειριστή συστήματος στην σελίδα auctions_reports.php. Τα αποτελέσματα των άγονων δημοπρασιών παρουσιάζονται στη συνέχεια στο διαχειριστή μέσω του XSLT μετασχηματισμού του εγγράφου report_barren.xml(εικόνα 4-18).

Το έγγραφο report_barren.xml αποτελείται από τα εξής στοιχεία:

- barren_id
Το στοιχείο barren_id χρησιμοποιείτε για την αποθήκευση ενός μοναδικού αναγνωριστικού για κάθε δημοπρασία που έληξε άγονη. Η λογική υλοποιήσεις του στοιχείου barren_id έγκειται στην ποσοτική μέτρηση των δημοπρασιών αυτών. Με αυτό τον τρόπο κατά το μετασχηματισμό του εγγράφου report_barren.xml σε ένα HTML έγγραφο για την δημιουργία αναφορών(page reports) του συστήματος, θα μπορούμε να έχουμε μία στήλη που αριθμεί τις άγονες δημοπρασίες μέσα σε ένα HTML πίνακα(εικόνα 4-18, στήλη A/A). Η ανάθεση τιμής στο στοιχείο barren_id γίνεται αυτόματα μέσω PHP κώδικα κατά την επιλογή του διαχειριστή του συστήματος .
- product_name
Το στοιχείο product_name χρησιμοποιείτε για την αποθήκευση του ονόματος ενός συγκεκριμένου προϊόντος το οποίο δεν πουλήθηκε στην δημοπρασία που έλαβε μέρος(άγονη δημοπρασία). Η ανάθεση τιμής στο στοιχείο product_name γίνεται αυτόματα μέσω PHP κώδικα που συνδυάζει το περιεχόμενο των κατάλληλων στοιχείων από τα αρχεία finished_auctions.xml και products.xml. Μέσω του σωστού κλειδιού από το στοιχείο product_id και την κατάσταση στην οποία έληξε μια δημοπρασία από το στοιχείο status στο αρχείο finished_auctions.xml, γίνεται η ανάθεση τιμής του κατάλληλου περιεχόμενου από το στοιχείο product_name μέσω του στοιχείου product_id από το αρχείο products.xml.
- start_price
Το στοιχείο start_price χρησιμοποιείτε για την αποθήκευση της αρχικής τιμής ενός προϊόντος. Η ανάθεση τιμής στο στοιχείο start_price γίνεται αυτόματα μέσω PHP κώδικα που συνδυάζει το περιεχόμενο του κατάλληλου στοιχείου price από το αρχείο finished_auctions.xml. Όπως αναφέραμε και παραπάνω, στην περίπτωση που λήξει άγονη μία δημοπρασία, το στοιχείο price(αρχείο finished_auctions.xml) θα είναι ίσο με την τιμή του στοιχείου price από το αρχείο products.xml.
- biggest_offer
Το στοιχείο biggest_offer χρησιμοποιείτε, για την αποθήκευση της μεγαλύτερης τιμής προσφοράς ενός χρήστη για μια δημοπρασία που έληξε άγονη. Στην περίπτωση που

δεν έχει γίνει καμία προσφορά μέχρι την λήξη της δημοπρασίας, το στοιχείο biggest_offer λαμβάνει την τιμή “No offer”.

A/A	Product Name	Start Price(€)	Bigger Offer(€)
1	Aspire E1-78	890	No offer
2	M800	40000	No offer
3	T600	500	490

Εικόνα 4-18 : Στιγμιότυπο αναφοράς των δημοπρασιών που έληξαν άγονες

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <barren_auctions>
3    <barren_auction>
4      <barren_id>1</barren_id>
5      <product_name>M800</product_name>
6      <start_price>40000</start_price>
7      <biggest_offer>No offer</biggest_offer>
8    </barren_auction>
9    <barren_auction>
10     <barren_id>2</barren_id>
11     <product_name>T600</product_name>
12     <start_price>500</start_price>
13     <biggest_offer>490</biggest_offer>
14   </barren_auction>
15   <barren_auction>
16     <barren_id>3</barren_id>
17     <product_name>Aspire E1-78</product_name>
18     <start_price>890</start_price>
19     <biggest_offer>No offer</biggest_offer>
20   </barren_auction>
21 </barren_auctions>

```

Εικόνα 4-19 : Στιγμιότυπο του αρχείου report_barren.xml

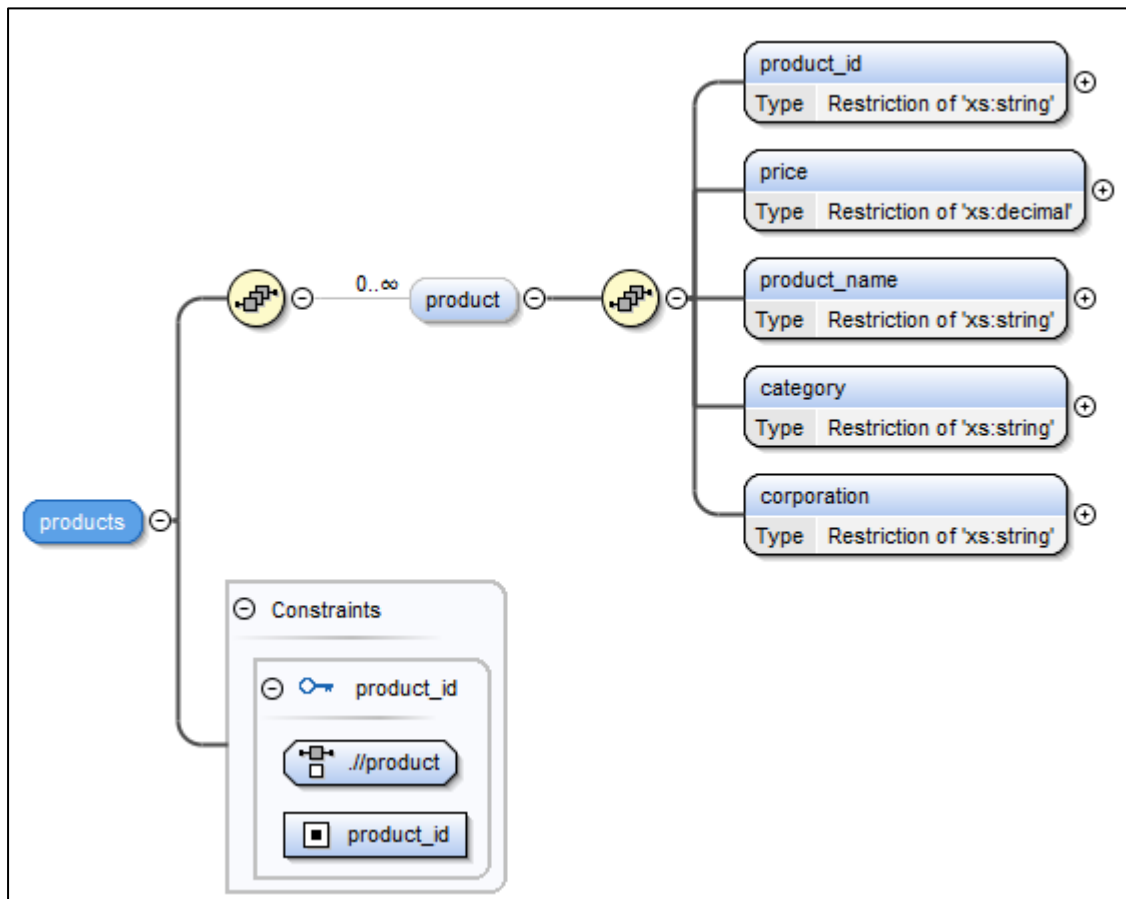
Το αρχείο search.xml βρίσκεται στο φάκελο store\ reports, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

4.3 Δημιουργία κατάλληλων εγγράφων XML Schema

Για την ορθή διεξαγωγή και υλοποίηση του συστήματος για online δημοπρασίες και πωλήσεις, είναι απαραίτητο το σύστημα αποθήκευσης σε XML αρχεία που έχουμε σχεδιάσει, να παρέχει τους κατάλληλους κανόνες και περιορισμούς για την αποφυγή προβλημάτων στην αποθήκευση δεδομένων. Για τον καθορισμό ενός συνόλου κανόνων που περιγράφουν τα στοιχεία και το περιεχόμενο των XML αρχείων, δημιουργήθηκαν τα εξής XML Schema:

- **Αρχείο products.xsd**

Το XML Schema products.xsd περιγράφει τους κανόνες και τους περιορισμούς των δεδομένων που θα πρέπει να πλήρη το σύστημα αποθήκευσης products.xml. Σύμφωνα με την περιγραφή που ορίζει το XML Schema, τα στοιχεία που αποτελούν το έγγραφο products.xml πρέπει να συμμορφώνονται με τους εξής κανόνες:



Εικόνα 4-20 : Σχηματική αποτύπωση του αρχείου products.xsd

- Το στοιχείο-ρίζα products(γραμμή 3, εικόνα 4-21):
 - Ορίζει ως κλειδί το στοιχείο product_id (10^η γραμμή)
 - Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (4^η γραμμή)
 - Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="products">
4    <xs:complexType>
5      <xs:sequence>
6        ....
7        ....
8      </xs:sequence>
9    </xs:complexType>
10   <xs:key name="product_id">
11     <xs:selector xpath="//product"/>
12     <xs:field xpath="product_id"/>
13   </xs:key>
14 </xs:element>
15 ...

```

Εικόνα 4-21 : Περιγραφή του στοιχείου-ρίζα products. Αρχείο products.xsd

Στη δέκατη γραμμή της εικόνας 4-21 ορίζεται ως κλειδί το στοιχείο `product_id`. Το κλειδί προσδιορίζει το περιεχόμενο ενός στοιχείου μοναδικά και αποτρέπει την υπάρξει άλλου στοιχείου με το ίδιο περιεχόμενο. Το στοιχείο που ορίζεται ως κλειδί πρέπει να έχει πάντα περιεχόμενο. Η δήλωση ενός κλειδιού περιέχει με συγκεκριμένη σειρά τα εξής στοιχεία:

- Το στοιχείο `selector`:
Καθορίζει μέσω της έκφρασης XPath το σύνολο των στοιχείων στο οποίο εφαρμόζεται το κλειδί. Το στοιχείο `selector` ορίζεται μία φορά.
- Το στοιχείο `field`:
Καθορίζει μέσω της έκφρασης XPath, ότι οι τιμές των στοιχείων που προσδιορίζει το στοιχείο `selector` πρέπει να είναι μοναδικές. Το στοιχείο `field` μπορεί να ορίζεται παραπάνω από μια φορά.

➤ Το στοιχείο `product(element node)`(γραμμή 3, εικόνα 4-22):

- Μπορεί να παρουσιάζετε από καμία έως άπειρες φορές μέσα στο στοιχείο-ρίζα `products` (3^η γραμμή)
- Ορίζει ένα σύνθετο τύπο δεδομένων(`complexType`) (4^η γραμμή)
- Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)
- Έχει 5 στοιχεία-παιδιά (6^η, 8^η, 10^η, 12^η, 14^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="product" minOccurs="0" maxOccurs="unbounded">
4   <xs:complexType>
5     <xs:sequence>
6       <xs:element name="product_id" minOccurs="1">
7         ....
8       <xs:element name="price" minOccurs="1">
9         ....
10      <xs:element name="product_name" minOccurs="1">
11        ....
12      <xs:element name="category" minOccurs="1">
13        ....
14      <xs:element name="corporation" minOccurs="1">
15        ....
16    </xs:sequence>
17  </xs:complexType>
18 </xs:element>
19 ...
```

Εικόνα 4-22 : Περιγραφή του στοιχείου `product`. Αρχείο `products.xsd`

➤ Το στοιχείο `product_id(data-node)`(γραμμή 3, εικόνα 4-23):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `product` και πρέπει να έχει περιεχόμενο. Η τιμή του `maxOccurs` είναι ίση με την τιμή του `minOccurs` από την στιγμή που δεν παρουσιάζετε (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα περιορισμό(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `product_id` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="product_id" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="^[^+=\[\]\*\^\n;:|%.\\\"?//{} (),>]+">
7      </xs:pattern>
8    </xs:restriction>
9  </xs:simpleType>
10 </xs:element>
11 ...
12 ...

```

Εικόνα 4-23 : Περιγραφή του στοιχείου product_id. Αρχείο products.xsd

Στη πέμπτη γραμμή της εικόνας 4-23, ορίζεται μέσω του στοιχείου restriction ένας περιορισμός στον τύπο δεδομένων που αποθηκεύονται στο στοιχείο product_id. Το στοιχείο restriction ακολουθείται από το χαρακτηριστικό base το οποίο προσδιορίζει τον τύπο δεδομένων πάνω στον οποίο θα εφαρμοστεί ο περιορισμός. Στη συνέχεια μέσω του στοιχείου pattern και του χαρακτηριστικού value καθορίζεται ένα πρότυπο περιορισμού δεδομένων. Σύμφωνα με το πρότυπο αυτό, το περιεχόμενο του στοιχείου product_id δεν μπορεί να περιέχει κάποιους ειδικούς χαρακτήρες όπως είναι τα σύμβολα %, +, =, ; κ.τ.λ.. Η τιμή του χαρακτηριστικού value μπαίνει πάντα μέσα σε διπλά εισαγωγικά.

➤ Το στοιχείο price(data-node)(γραμμή 3, εικόνα 4-24):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερτα product και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα περιορισμό(restriction) στον δεκαδικό τύπο δεδομένων του στοιχείου price (5^η γραμμή)
- Καθορίζει το κατώτατο όριο(minExclusive) της δεκαδικής τιμής που μπορεί να πάρει το στοιχείο price. price > 0 (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="price" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:decimal">
6        <xs:minExclusive value="0"/>
7      </xs:restriction>
8    </xs:simpleType>
9  </xs:element>
10 ...
11 ...

```

Εικόνα 4-24 : Περιγραφή του στοιχείου price. Αρχείο products.xsd

Στην έκτη γραμμή της εικόνας 4-24, καθορίζεται μέσω του στοιχείου minExclusive ένα πρότυπο περιορισμού το οποίο εφαρμόζεται σε αριθμητικές παραστάσεις. Το χαρακτηριστικό value δηλώνει το κατώτερο όριο των αριθμητικών τιμών που μπορεί να πάρει το στοιχείο price(η τιμή του στοιχείου πρέπει να είναι μεγαλύτερη της τιμής του χαρακτηριστικού). Με αυτό τον τρόπο περιορίσαμε το περιεχόμενο του στοιχείου price να μην μπορεί να πάρει μηδενικές και αρνητικές

τιμές. Οι τιμές που μπορεί να δέχεται είναι από το ένα μέχρι το άπειρο. Η μαθηματική έκφραση του περιορισμού ορίζεται με τον εξής τρόπο: $price > 0$

➤ Το στοιχείο `product_name`(data-node)(γραμμή 3, εικόνα 4-25):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `product` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `product_name` (5^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου `product_name` (6^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου `product_name` (7^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `product_name`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (8^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="product_name" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:minLength value="1"/>
7       <xs:maxLength value="16"/>
8       <xs:pattern value="^[^+=\[\]\*\^\n;:|%.\\?//{}()>]+>">
9     </xs:restriction>
10  </xs:simpleType>
11 </xs:element>
12 ...
13 ...
14 ...
```

Εικόνα 4-25 : Περιγραφή του στοιχείου `product_name`. Αρχείο `products.xsd`

Στην έκτη και έβδομη γραμμή της εικόνας 4-25, καθορίζεται μέσω των στοιχείων `minLength` και `maxLength`, τα όρια του μήκους χαρακτήρων στον αλφαριθμητικό τύπο δεδομένων που μπορεί να αποθηκευτεί στο περιεχόμενο του στοιχείου `product_name`. Μέσω του χαρακτηριστικού `value` του στοιχείου `minLength` δηλώνετε το κατώτατο όριο μήκους χαρακτήρων και μέσω του χαρακτηριστικού `value` του στοιχείου `maxLength` δηλώνετε το ανώτατο όριο μήκους χαρακτήρων. Με αυτό τον τρόπο περιορίσαμε το περιεχόμενο του στοιχείου `product_name` να μην μπορεί να έχει λιγότερο από ένα χαρακτήρα και περισσότερους από 16. Η μαθηματική έκφραση του περιορισμού ορίζεται με τον εξής τρόπο: $1 \geq \text{μήκος χαρακτήρων} \leq 16$.

➤ Το στοιχείο `category`(data-node)(γραμμή 3, εικόνα 4-26):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `product` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `category` (5^η γραμμή)

- Καθορίζει ένα σύνολο αποδεκτών τιμών για το περιεχόμενο που μπορεί να έχει το στοιχείο category (6^η, 7^η, 8^η, 9^η, 10^η, 11^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="category" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:enumeration value="Computers" />
7        <xs:enumeration value="Clothes" />
8        <xs:enumeration value="Watches" />
9        <xs:enumeration value="Art" />
10       <xs:enumeration value="Cars" />
11       <xs:enumeration value="Bikes" />
12     </xs:restriction>
13   </xs:simpleType>
14 </xs:element>
15 ...
16 ...

```

Εικόνα 4-26 : Περιγραφή του στοιχείου category. Αρχείο products.xsd

Στην 6^η, 7^η, 8^η, 9^η, 10^η, 11^η γραμμή της εικόνας 4-26, καθορίζεται ένα σύνολο αποδεκτών αλφαριθμητικών τιμών μέσω των στοιχείων enumeration. Το χαρακτηριστικό value των στοιχείων enumeration προσδιορίζει τις τιμές που μπορεί να πάρει το στοιχείο category. Σύμφωνα με τον περιορισμό που ορίζει το κάθε στοιχείο enumeration οι τιμές που μπορεί να πάρει το στοιχείο category είναι: Computers, Cars, Watches, Clothes, Art, Bikes.

➤ Το στοιχείο corporation(data-node)(γραμμή 3, εικόνα 4-27):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα product και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(restriction) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου corporation (5^η γραμμή)
- Καθορίζει το πρότυπο(pattern) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου corporation. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου corporation. Το κατώτατο όριο είναι ένας χαρακτήρας (8^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου corporation. Το ανώτατο όριο είναι έως δεκαέξι χαρακτήρες (9^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="corporation" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="[^-+=\$\[\]\*\^\n;|%.\\\"?/{ } ( ) , >]+">
7        </xs:pattern>
8        <xs:minLength value="1"/>
9        <xs:maxLength value="16"/>
10       </xs:restriction>
11     </xs:simpleType>
12   </xs:element>
13   ...
14   ...

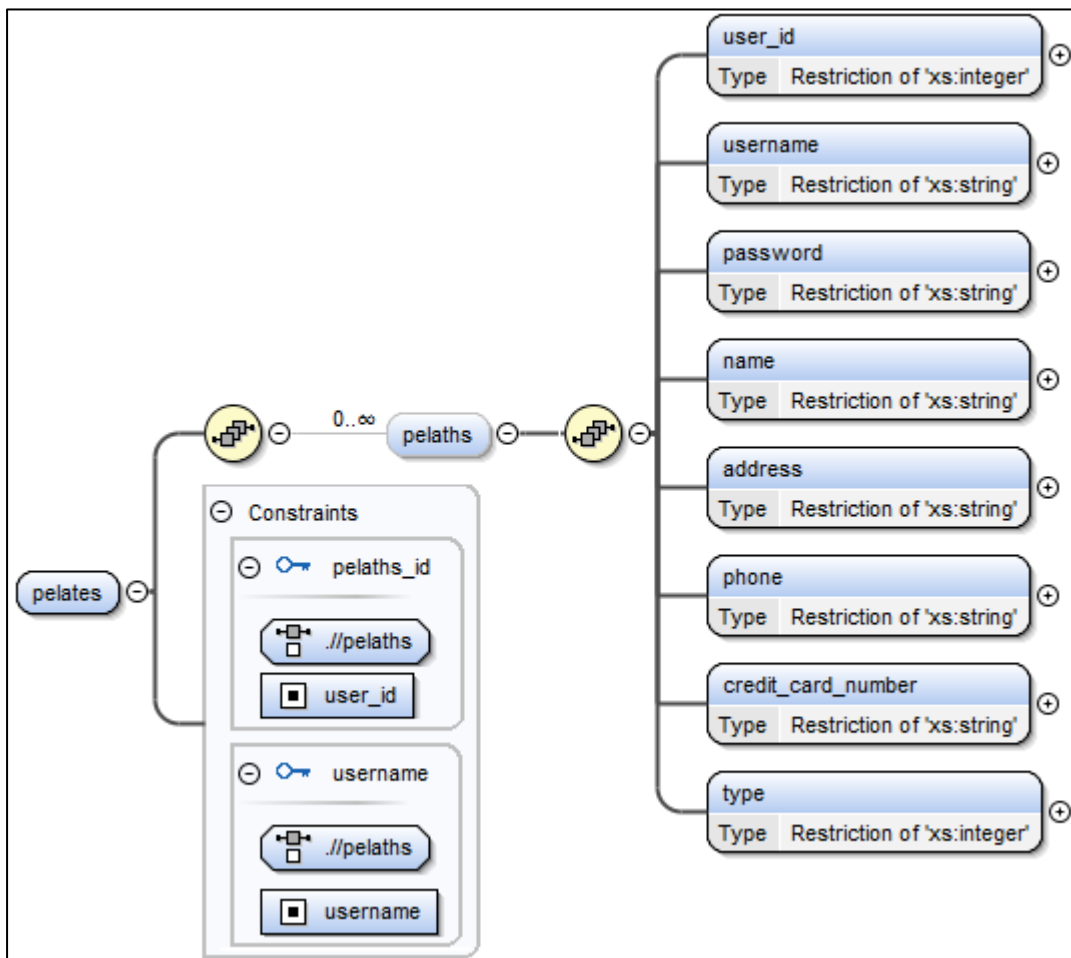
```

Εικόνα 4-27 : Περιγραφή του στοιχείου corporation. Αρχείο products.xsd

Το αρχείο products.xsd βρίσκετε στο φάκελο store\connections, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο pelates.xsd**

Το XML Schema pelates.xsd περιγράφει τους κανόνες και τους περιορισμούς των δεδομένων που θα πρέπει να πλήρη το σύστημα αποθήκευσης pelates.xml. Συμφώνα με την περιγραφή που ορίζει το XML Schema, τα στοιχεία που αποτελούν το έγγραφο pelates.xml πρέπει να συμμορφώνονται με τους εξής κανόνες:



Εικόνα 4-28 : Σχηματική αποτύπωση του αρχείου pelates.xsd

➤ Το στοιχείο-ρίζα pelates(γραμμή 3, εικόνα 4-29):

- Ορίζει ως κλειδί το στοιχείο user_id και το στοιχείο username (10^η και 14^η γραμμή)
- Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (4^η γραμμή)
- Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="pelates">
4   <xs:complexType>
5     <xs:sequence>
6       ....
7       ....
8     </xs:sequence>
9   </xs:complexType>
10  <xs:key name="pelaths_id">
11    <xs:selector xpath="//pelaths"/>
12    <xs:field xpath="user_id"/>
13  </xs:key>
14  <xs:key name="username">
15    <xs:selector xpath="//pelaths"/>
16    <xs:field xpath="username"/>
17  </xs:key>
18 </xs:element>
19 ...
20 ...
```

Εικόνα 4-29 : Περιγραφή του στοιχείου-ρίζα pelates. Αρχείο pelates.xsd

➤ Το στοιχείο pelaths(element node)(γραμμή 2, εικόνα 4-30):

- Μπορεί να παρουσιάζετε από καμία έως άπειρες φορές μέσα στο στοιχείο-ρίζα pelates (2^η γραμμή)
- Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (3^η γραμμή)
- Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (4^η γραμμή)
- Έχει 8 στοιχεία-παιδιά (5^η, 6^η, 7^η, 8^η, 9^η, 10^η, 11^η, 13^η γραμμή)

```
1 ...
2 <xs:element name="pelaths" minOccurs="0" maxOccurs="unbounded">
3   <xs:complexType>
4     <xs:sequence>
5       <xs:element name="user_id" minOccurs="1">..</xs:element>
6       <xs:element name="username" minOccurs="1">..</xs:element>
7       <xs:element name="password" minOccurs="1">..</xs:element>
8       <xs:element name="name" minOccurs="1">...</xs:element>
9       <xs:element name="address" minOccurs="1">..</xs:element>
10      <xs:element name="phone" minOccurs="1">..</xs:element>
11      <xs:element name="credit_card_number" minOccurs="1">...
12    </xs:sequence>
13    <xs:element name="type" minOccurs="1">..</xs:element>
14  </xs:complexType>
15 </xs:element>
16 ...
17 ...
```

Εικόνα 4-30 : Περιγραφή του στοιχείου pelaths. Αρχείο pelates.xsd

➤ Το στοιχείο `user_id(data-node)`(γραμμή 3, εικόνα 4-31):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `relaths` και πρέπει να έχει περιεχόμενο. Η τιμή του `maxOccurs` είναι ίση με την τιμή του `minOccurs` από την στιγμή που δεν παρουσιάζετε. (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα περιορισμό(`restriction`) στον ακέραιο αριθμό τύπου δεδομένων του στοιχείου `user_id` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού (6^η γραμμή)
- Καθορίζει το κατώτατο όριο(`minExclusive`) τιμής που μπορεί να πάρει το στοιχείο `user_id`. `user_id > 0` (8^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="user_id" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:integer">
6       <xs:pattern value="^[^-=\$\[\]\*\^\^n;:|\%.s\?\//{}(),>]+">
7     </xs:pattern>
8     <xs:minExclusive value="0"/>
9   </xs:restriction>
10  </xs:simpleType>
11 </xs:element>
12 ...
13 ...
```

Εικόνα 4-31 : Περιγραφή του στοιχείου `user_id`. Αρχείο `pelates.xsd`

➤ Το στοιχείο `username(data-node)`(γραμμή 3, εικόνα 4-32):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `relaths` και πρέπει να έχει περιεχόμενο. (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `username` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `username`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως `%`, `+`, `=`, `;` κ.τ.λ. (6^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου `username`. Οι λιγότεροι χαρακτήρες που μπορεί να έχει είναι τρεις. `username ≥ 3` (8^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου `username`. Οι περισσότεροι χαρακτήρες που μπορεί να έχει είναι μέχρι και 16. `username ≤ 16` (7^η γραμμή)

```

1   ...
2   ...
3   <xs:element name="username" minOccurs="1">
4     <xs:simpleType>
5       <xs:restriction base="xs:string">
6         <xs:pattern value="^[^= $\ [ \ ] \* \^ \n ; : | % . \s \\ \? // { } ( ) , > ] + ">
7       </xs:pattern>
8       <xs:minLength value="3"/>
9       <xs:maxLength value="16"/>
10      </xs:restriction>
11    </xs:simpleType>
12  </xs:element>
13  ...
14  ...

```

Εικόνα 4-32 : Περιγραφή του στοιχείου username. Αρχείο pelates.xsd

➤ Το στοιχείο password(data-node)(γραμμή 3, εικόνα 4-33):

- Μπορεί να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα pelaths και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(restriction) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου password (5^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου password. Οι λιγότεροι χαρακτήρες που μπορεί να έχει είναι 6. $\text{password} \geq 6$ (6^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου password. Οι περισσότεροι χαρακτήρες που μπορεί να έχει είναι μέχρι και 32. $\text{password} \leq 32$ (7^η γραμμή)

```

1   ...
2   ...
3   <xs:element name="password" minOccurs="1">
4     <xs:simpleType>
5       <xs:restriction base="xs:string">
6         <xs:minLength value="6"/>
7         <xs:maxLength value="32"/>
8       </xs:pattern>
9     </xs:restriction>
10    </xs:simpleType>
11  </xs:element>
12  ...
13  ...

```

Εικόνα 4-33 : Περιγραφή του στοιχείου password. Αρχείο pelates.xsd

➤ Το στοιχείο name(data-node)(γραμμή 3, εικόνα 4-34):

- Μπορεί να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα pelaths και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)

- Ορίζει ένα σύνολο περιορισμών(restriction) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου name (5^η γραμμή)
- Καθορίζει το πρότυπο(pattern) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου name. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου name. Οι λιγότεροι χαρακτήρες που μπορεί να έχει είναι τρεις. name ≥ 3 (8^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου name. Οι περισσότεροι χαρακτήρες που μπορεί να έχει είναι μέχρι και 16. name ≤ 16 (9^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="name" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="^[^+=\$\[\]\*\^\n;:#!%.\s\?\{\}(),>]+">
7      </xs:pattern>
8      <xs:minLength value="3"/>
9      <xs:maxLength value="16"/>
10     </xs:restriction>
11   </xs:simpleType>
12 </xs:element>
13 ...
14 ...

```

Εικόνα 4-34 : Περιγραφή του στοιχείου name. Αρχείο pelates.xsd

- Το στοιχείο address(data-node)(γραμμή 2, εικόνα 4-35):
 - Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα pelaths και μπορεί να μην έχει περιεχόμενο (2^η γραμμή)
 - Ορίζει ένα απλό τύπο δεδομένων(simpleType) (3^η γραμμή)
 - Ορίζει ένα σύνολο περιορισμών(restriction) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου address (4^η γραμμή)
 - Καθορίζει το πρότυπο(pattern) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου address. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (5^η γραμμή)

```

1  ...
2  <xs:element name="address" minOccurs="1">
3    <xs:simpleType>
4      <xs:restriction base="xs:string">
5        <xs:pattern value="^[^+=#\$!@/\^\^;:|%.\\?\{\}(),>]*">
6      </xs:pattern>
7    </xs:restriction>
8  </xs:simpleType>
9 </xs:element>
10 ...

```

Εικόνα 4-35 : Περιγραφή του στοιχείου address. Αρχείο pelates.xsd

➤ Το στοιχείο `phone(data-node)`(γραμμή 3, εικόνα 4-36):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `relaths` και μπορεί να μην έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `phone` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `phone`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως `%`, `+`, `=`, `;` κ.τ.λ. (6^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου `phone`. Οι περισσότεροι χαρακτήρες που μπορεί να έχει είναι μέχρι και 13. `phone ≤ 13` (8^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="phone" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:pattern value="[^\*#\$\!@/\^;:|\%.\\\"?//{}() ,>]*" />
7     </xs:pattern>
8     <xs:maxLength value="13" />
9   </xs:restriction>
10  </xs:simpleType>
11 </xs:element>
12 ...
13 ...
```

Εικόνα 4-36 : Περιγραφή του στοιχείου `phone`. Αρχείο XML `relates.xsd`

➤ Το στοιχείο `credit_card_number(data-node)`(γραμμή 3, εικόνα 4-37):

- Μπορεί να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `relaths` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `credit_card_number` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `credit_card_number`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως `%`, `+`, `=`, `;` κ.τ.λ. (6^η γραμμή)
- Καθορίζει το κατώτατο όριο μήκους χαρακτήρων του στοιχείου `credit_card_number`. Οι λιγότεροι χαρακτήρες που μπορεί να έχει είναι τρεις. `credit_card_number ≥ 3` (8^η γραμμή)
- Καθορίζει το ανώτατο όριο μήκους χαρακτήρων του στοιχείου `credit_card_number`. Οι περισσότεροι χαρακτήρες που μπορεί να έχει είναι μέχρι και 32. `credit_card_number ≤ 32` (9^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="credit_card_number" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="[^*=#$!@^;:|%.\\?{}() ,>]*">
7        </xs:pattern>
8        <xs:minLength value="3"/>
9        <xs:maxLength value="32"/>
10       </xs:restriction>
11     </xs:simpleType>
12   </xs:element>
13   ...
14   ...

```

Εικόνα 4-37 : Περιγραφή του στοιχείου credit_card_number. Αρχείο pelates.xsd

➤ Το στοιχείο type(data-node)(γραμμή 3, εικόνα 4-38):

- Μπορεί να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα pelates και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(restriction) στον ακέραιο τύπο δεδομένων του στοιχείου type (5^η γραμμή)
- Καθορίζει το κατώτατο όριο(minInclusive) στην ακέραια τιμή που μπορεί να πάρει το στοιχείο type. $type \geq 0$ (6^η γραμμή)
- Καθορίζει το ανώτατο όριο(maxInclusive) στην ακέραια τιμή που μπορεί να πάρει το στοιχείο type. $type \leq 1$ (7^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="type" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:integer">
6        <xs:minInclusive value="0"/>
7        <xs:maxInclusive value="1"/>
8      </xs:restriction>
9    </xs:simpleType>
10   </xs:element>
11   ...
12   ...

```

Εικόνα 4-38 : Περιγραφή του στοιχείου type. Αρχείο pelates.xsd

Στην έκτη και έβδομη γραμμή της εικόνας 4-38, καθορίζεται μέσω του στοιχείου minInclusive και maxInclusive ένα πρότυπο περιορισμού το οποίο εφαρμόζεται σε αριθμητικές παραστάσεις. Η λειτουργία των δυο αυτών στοιχείων είναι ίδια με την λειτουργία των στοιχείων minExclusive και maxExclusive με την διαφορά ότι η τιμή του στοιχείου type, σύμφωνα με την εικόνα 4-38 μπορεί να είναι ίση με το μηδέν και το ένα αντίστοιχα. Αν για παράδειγμα είχαμε

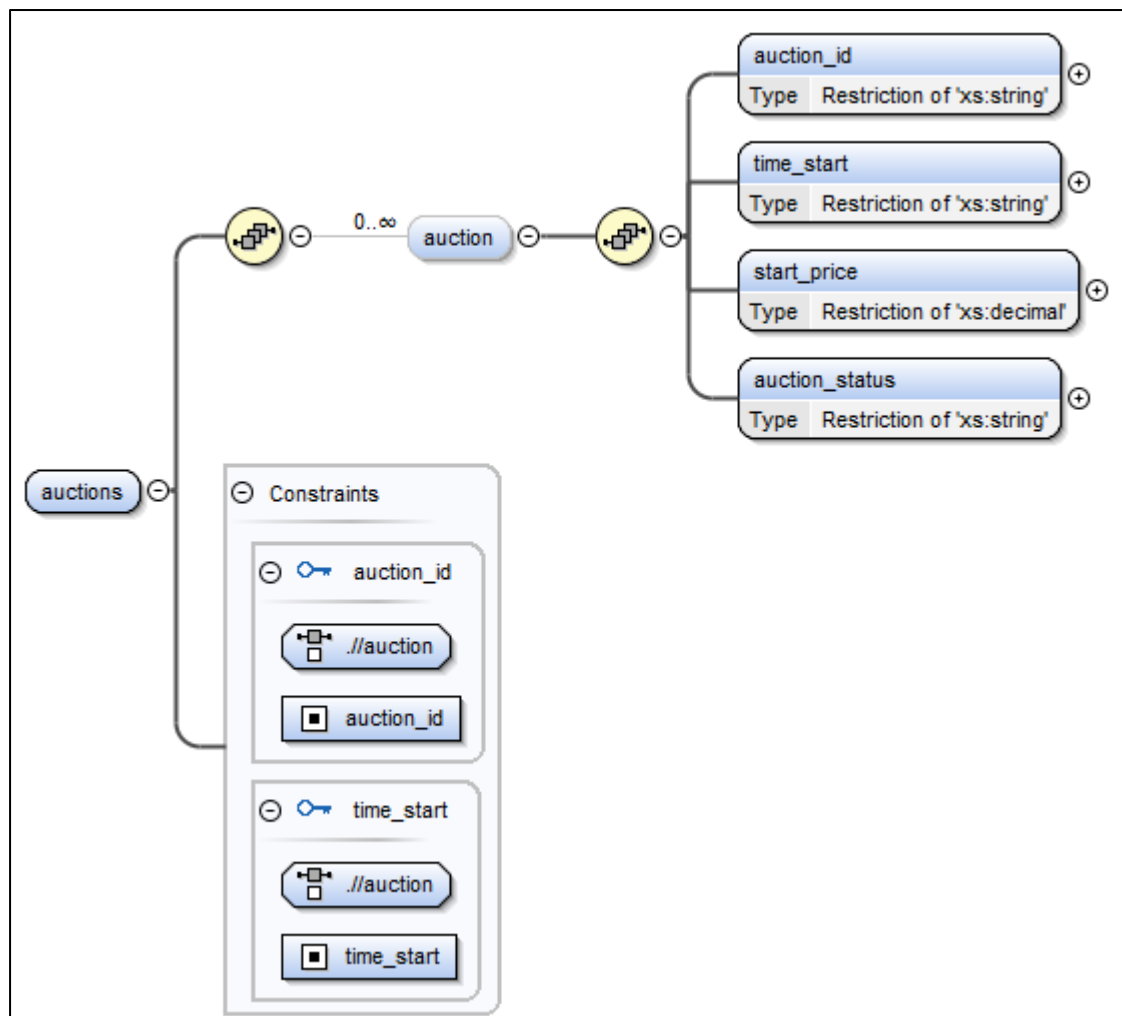
καθορίσει τα όρια των τιμών που μπορεί να πάρει το στοιχείο type με minExclusive και maxExclusive τότε το στοιχείο δεν θα μπορούσε να πάρει τις τιμές ένα και μηδέν. Με μαθηματική εξήγηση:

- minExclusive και maxExclusive
 $0 > \text{type} < 1$
- minInclusive και maxInclusive
 $0 \geq \text{type} \leq 1$

Το αρχείο relates.xsd βρίσκετε στο φάκελο store\users, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο auctions.xsd**

Το XML Schema auctions.xsd περιγράφει τους κανόνες και τους περιορισμούς των δεδομένων που θα πρέπει να πλήρη το σύστημα αποθήκευσης auctions.xml. Σύμφωνα με την περιγραφή που ορίζει το XML Schema, τα στοιχεία που αποτελούν το έγγραφο auctions.xml πρέπει να συμμορφώνονται με τους εξής κανόνες:



Εικόνα 4-39 : Σχηματική αποτύπωση του αρχείου auctions.xsd

- Το στοιχείο-ρίζα auctions(γραμμή 3, εικόνα 4-40):
 - Ορίζει ως κλειδί το στοιχείο auction_id (10^η γραμμή)
 - Ορίζει ως κλειδί το στοιχείο time_start (14^η γραμμή)

- Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (4^η γραμμή)
- Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="auctions">
4    <xs:complexType>
5      <xs:sequence>
6        ....
7        ....
8      </xs:sequence>
9    </xs:complexType>
10   <xs:key name="auction_id">
11     <xs:selector xpath="//auction"/>
12     <xs:field xpath="auction_id"/>
13   </xs:key>
14   <xs:key name="time_start">
15     <xs:selector xpath="//auction"/>
16     <xs:field xpath="time_start"/>
17   </xs:key>
18 </xs:element>
19 ...
20 ...

```

Εικόνα 4-40 : Περιγραφή του στοιχείου-ρίζα auctions. Αρχείο auctions.xsd

- Το στοιχείο auction(element node)(γραμμή 3, εικόνα 4-41):
 - Μπορεί να παρουσιάζετε από καμία έως άπειρες φορές μέσα στο στοιχείο-ρίζα auctions (3^η γραμμή)
 - Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (4^η γραμμή)
 - Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)
 - Έχει 4 στοιχεία-παιδιά (6^η, 8^η, 10^η, 12^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="auction" minOccurs="0" maxOccurs="unbounded">
4    <xs:complexType>
5      <xs:sequence>
6        <xs:element name="auction_id" minOccurs="1">
7          ...</xs:element>
8        <xs:element name="time_start" minOccurs="1">
9          ...</xs:element>
10       <xs:element name="start_price" minOccurs="1">
11         ...</xs:element>
12       <xs:element name="auction_status" minOccurs="1">
13         ...</xs:element>
14     </xs:sequence>
15   </xs:complexType>
16 </xs:element>
17 ...
18 ...

```

Εικόνα 4-41 : Περιγραφή του στοιχείου auction. Αρχείο auctions.xsd

➤ Το στοιχείο `auction_id`(data-node)(γραμμή 3, εικόνα 4-42):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `auction` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `auction_id` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `auction_id`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="auction_id" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:pattern value="^[^=\\$\\[\\]\\*\\^\\n;:|%.\\s\\?\\/\\{\\} \\(\\),>]+">
7     </xs:pattern>
8   </xs:restriction>
9 </xs:simpleType>
10 </xs:element>
11 ...
12 ...
```

Εικόνα 4-42 : Περιγραφή του στοιχείου `auction_id`. Αρχείο `auctions.xsd`

➤ Το στοιχείο `time_start`(data-node)(γραμμή 3, εικόνα 4-43):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `auction` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `time_start` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `time_start`. (6^η γραμμή)

Το στοιχείο `time_start` έχει οριστεί ως κλειδί στην δήλωση του στοιχείου-ρίζα `auctions` γιατί δεν πρέπει δυο δημοπρασίες να συμπίπτουν χρονικά. Όταν ένα στοιχείο ορίζεται ως κλειδί πρέπει πάντα να έχει περιεχόμενο.

```
1 ...
2 ...
3 <xs:element name="time_start" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:pattern value=
7 "T[0-9][0-9]:[0-9][0-9] [0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]"/>
8     </xs:restriction>
9   </xs:simpleType>
10 </xs:element>
11 ...
12 ...
```

Εικόνα 4-43 : Περιγραφή του στοιχείου `time_start`. Αρχείο `auctions.xsd`

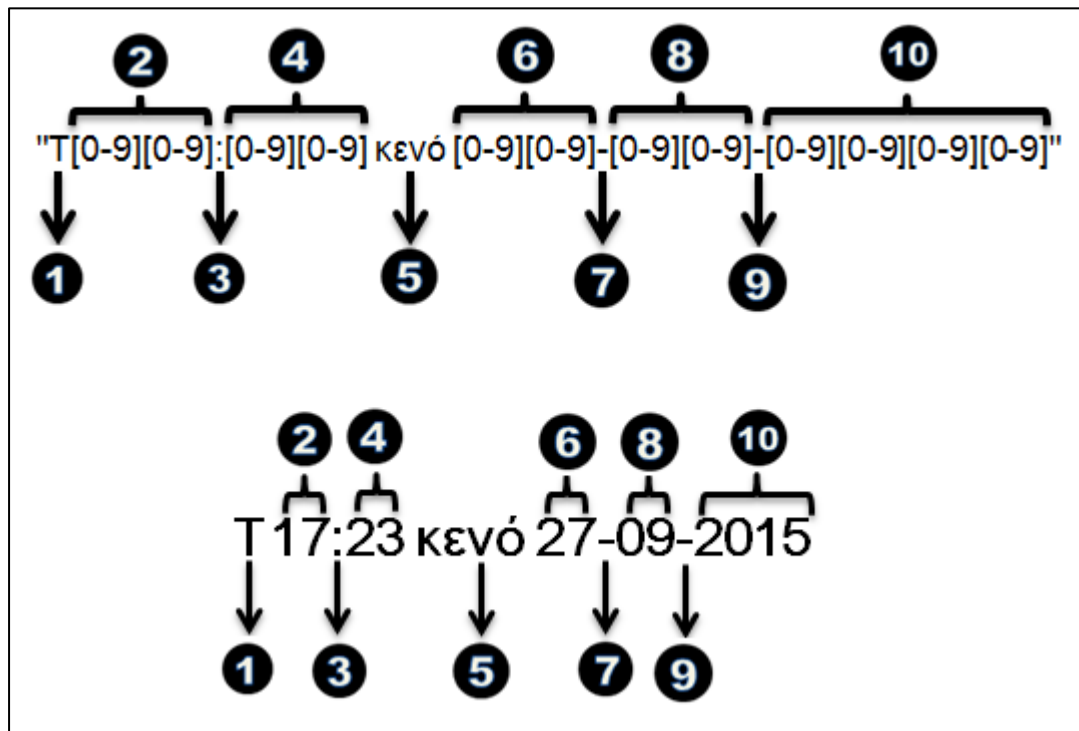
Για την δημιουργία του κατάλληλου προτύπου της ώρας έναρξης μίας δημοπρασίας, χρησιμοποιήθηκε το στοιχείο pattern για το περιορισμό του αλφαριθμητικού τύπο δεδομένων του στοιχείου time_start.

A/A	Product Name	Price(€)	Start Time	Corpration	Category
1	A1	3063	T17:23 27-09-2015	Audi	Cars

Εικόνα 4-44 : Στιγμιότυπο πίνακα δημοπρασιών

Σύμφωνα με το στιγμιότυπο της εικόνας 4-44 το πρότυπο της ώρα πρέπει να ακολουθεί την εξής μορφή:

- Πρότυπο ώρας : T17:23 κενό 27-09-2015
- Δημιουργία του κατάλληλου αλφαριθμητικού pattern:



Εικόνα 4-45 : Δημιουργία pattern για το στοιχείο time_start. Αρχείο XMLSchema auctions.xsd

Όπου:

1. Η περιγραφή της ώρας ξεκινά με το κεφαλαίο χαρακτήρα "T"
2. Η ώρα αποτελείται από δύο ακέραιους αριθμούς που μπορούν να πάρουν τιμές από το ένα μέχρι το εννέα. Η δήλωση τιμής στο στοιχείο pattern: [0-9][0-9]
3. Ξεχωρίζουμε την ώρα από τα λεπτά με το χαρακτήρα της άνω και κάτω τελείας. Η δήλωση τιμής στο στοιχείο pattern: ":"
4. Η περιγραφή των λεπτών αποτελείται από δύο ακέραιους αριθμούς που μπορούν να πάρουν τιμές από το ένα μέχρι το εννέα. Η δήλωση τιμής στο στοιχείο pattern: [0-9][0-9]
5. Θέτουμε ένα κενό ανάμεσα στην ώρα και την ημερομηνία.

6. Η περιγραφή της ημέρας αποτελείται από δύο ακέραιους αριθμούς που μπορούν να πάρουν τιμές από το ένα μέχρι το εννέα. Η δήλωση τιμής στο στοιχείο pattern: **[0-9][0-9]**
7. Ξεχωρίζουμε τις ημέρες από τους μήνες με το χαρακτήρα της άνω και κάτω τελείας. Η δήλωση τιμής στο στοιχείο pattern: “:”
8. Η περιγραφή του μήνα αποτελείται από δύο ακέραιους αριθμούς που μπορούν να πάρουν τιμές από το ένα μέχρι το εννέα. Η δήλωση τιμής στο στοιχείο pattern: **[0-9][0-9]**
9. Ξεχωρίζουμε τους μήνες από τα χρόνια με το χαρακτήρα της άνω και κάτω τελείας. Η δήλωση τιμής στο στοιχείο pattern: “:”
10. Η περιγραφή της χρονολογίας αποτελείται από 4 ακέραιους αριθμούς που μπορούν να πάρουν τιμές από το ένα μέχρι το εννέα. Η δήλωση τιμής στο στοιχείο pattern: **[0-9][0-9][0-9][0-9]**

Η δημιουργία του κατάλληλου πρότυπου μας εξασφαλίζει ότι το περιεχόμενο του στοιχείου time_start θα ακολουθεί πάντα την συγκεκριμένη σύνταξη για την αποφυγή προβλημάτων στην παρουσίαση χρήσιμων πληροφοριών σε ένα χρήστη.

➤ Το στοιχείο start_price(data-node)(γραμμή 3, εικόνα 4-46):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα auction και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα περιορισμό(restriction) στον δεκαδικό τύπο δεδομένων του στοιχείου start_price (5^η γραμμή)
- Καθορίζει το κατώτατο όριο(minExclusive) της δεκαδικής τιμής που μπορεί να πάρει το στοιχείο start_price. start_price > 0 (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="start_price" minOccurs="1">
4  <xs:simpleType>
5  <xs:restriction base="xs:decimal">
6  <xs:minExclusive value="0"/>
7  </xs:restriction>
8  </xs:simpleType>
9  </xs:element>
10 ...
11 ...

```

Εικόνα 4-46 : Περιγραφή του στοιχείου start_price. Αρχείο auctions.xsd

➤ Το στοιχείο auction_status(data-node)(γραμμή 3, εικόνα 4-47):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα auction και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(simpleType) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(restriction) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου auction_status (5^η γραμμή)
- Καθορίζει το πρότυπο(pattern) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου auction_status. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="auction_status" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="^[^*+=#$!@/^\^;:|%.\\?//{}()>]*">
7        </xs:pattern>
8      </xs:restriction>
9    </xs:simpleType>
10   </xs:element>
11   ...
12   ...

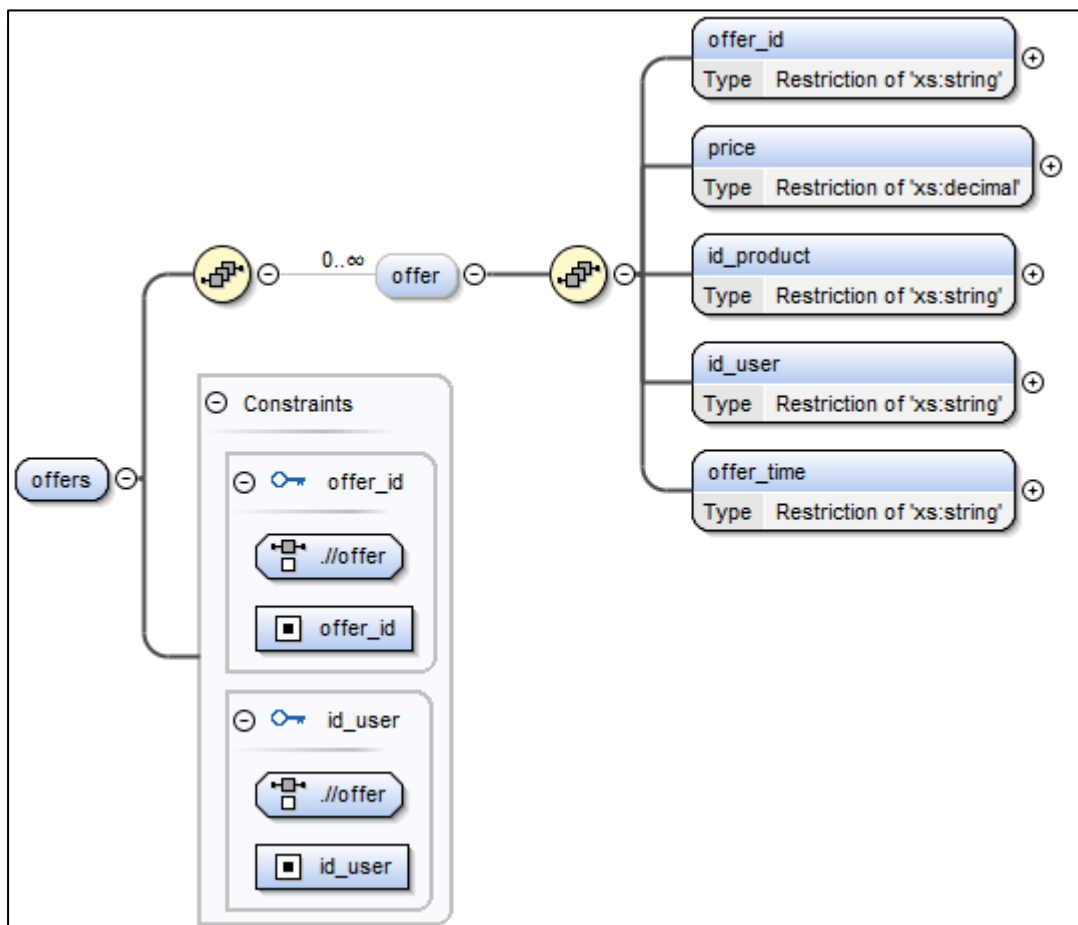
```

Εικόνα 4-47 : Περιγραφή του στοιχείου auction_status. Αρχείο auctions.xsd

Το αρχείο auctions.xsd βρίσκεται στο φάκελο store\connections, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

- **Αρχείο offers.xsd**

Το XML Schema offers.xsd περιγράφει τους κανόνες και τους περιορισμούς των δεδομένων που θα πρέπει να πληρή το σύστημα αποθήκευσης offers.xml. Συμφώνα με την περιγραφή που ορίζει το XML Schema, τα στοιχεία που αποτελούν το έγγραφο offers.xml πρέπει να συμμορφώνονται με τους εξής κανόνες:



Εικόνα 4-48 : Σχηματική αποτύπωση του αρχείου offers.xsd

- Το στοιχείο-ρίζα offers(γραμμή 3, εικόνα 4-49):
 - Ορίζει ως κλειδί το στοιχείο offer_id (9^η γραμμή)
 - Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (4^η γραμμή)
 - Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (5^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="offers">
4  <xs:complexType>
5  <xs:sequence>
6  ....
7  </xs:sequence>
8  </xs:complexType>
9  <xs:key name="offer_id">
10 <xs:selector xpath="./offer"/>
11 <xs:field xpath="offer_id"/>
12 </xs:key>
13 </xs:element>
14 ...
15 ...

```

Εικόνα 4-49 : Περιγραφή του στοιχείου-ρίζα offers. Αρχείο offers.xsd

- Το στοιχείο offer(element node)(γραμμή 2, εικόνα 4-50):
 - Μπορεί να παρουσιάζετε από καμία έως άπειρες φορές μέσα στο στοιχείο-ρίζα offers (2^η γραμμή)
 - Ορίζει ένα σύνθετο τύπο δεδομένων(complexType) (3^η γραμμή)
 - Καθορίζει την σειρά εμφάνισης των στοιχείων που περικλείει (4^η γραμμή)
 - Έχει 5 στοιχεία-παιδιά (5^η, 6^η, 7^η, 8^η, 9^η γραμμή)

```

1  ...
2  <xs:element name="offer" minOccurs="0" maxOccurs="unbounded">
3  <xs:complexType>
4  <xs:sequence>
5  <xs:element name="offer_id" minOccurs="1">...</xs:element>
6  <xs:element name="price" minOccurs="1">...</xs:element>
7  <xs:element name="id_product" minOccurs="1">...</xs:element>
8  <xs:element name="id_user" minOccurs="1">...</xs:element>
9  <xs:element name="offer_time" minOccurs="1">...</xs:element>
10 </xs:sequence>
11 </xs:complexType>
12 </xs:element>
13 ...

```

Εικόνα 4-50 : Περιγραφή του στοιχείου offer. Αρχείο offers.xsd

- Το στοιχείο `offer_id`(data-node)(γραμμή 3, εικόνα 4-51):
 - Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `offer` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
 - Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
 - Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `offer_id` (5^η γραμμή)
 - Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `offer_id`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="offer_id" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:string">
6        <xs:pattern value="[^-=$\[\]\*\^\n;:|\%.\s\\?//{}() ,>]+">
7      </xs:pattern>
8    </xs:restriction>
9  </xs:simpleType>
10 </xs:element>
11 ...
12 ...

```

Εικόνα 4-51 : Περιγραφή του στοιχείου `offer_id`. Αρχείο `offers.xsd`

- Το στοιχείο `price`(data-node)(γραμμή 3, εικόνα 4-52):
 - Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `offer` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
 - Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
 - Ορίζει ένα περιορισμό(`restriction`) στον δεκαδικό τύπο δεδομένων του στοιχείου `price` (5^η γραμμή)
 - Καθορίζει το κατώτατο όριο(`minExclusive`) της δεκαδικής τιμής που μπορεί να πάρει το στοιχείο `price`. `price > 0` (6^η γραμμή)

```

1  ...
2  ...
3  <xs:element name="price" minOccurs="1">
4    <xs:simpleType>
5      <xs:restriction base="xs:decimal">
6        <xs:minExclusive value="0"/>
7      </xs:restriction>
8    </xs:simpleType>
9  </xs:element>
10 ...
11 ...

```

Εικόνα 4-52 : Περιγραφή του στοιχείου `price`. Αρχείο `offers.xsd`

➤ Το στοιχείο `id_product(data-node)`(γραμμή 3, εικόνα 4-53):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `offer` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `id_product` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `id_product`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="id_product" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:pattern value="[^\d+=#\$!@/^\^;:|%.\\?//{}(),>]*">
7     </xs:pattern>
8   </xs:restriction>
9 </xs:simpleType>
10 </xs:element>
11 ...
12 ...
```

Εικόνα 4-53 : Περιγραφή του στοιχείου `id_product`. Αρχείο `offers.xsd`

➤ Το στοιχείο `id_user(data-node)`(γραμμή 3, εικόνα 4-54):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `offer` και πρέπει να έχει περιεχόμενο (3^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (4^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `id_user` (5^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `id_user`. Δεν επιτρέπονται ειδικοί χαρακτήρες όπως %, +, =, ; κ.τ.λ. (6^η γραμμή)

```
1 ...
2 ...
3 <xs:element name="id_user" minOccurs="1">
4   <xs:simpleType>
5     <xs:restriction base="xs:string">
6       <xs:pattern value="[^\d+=#\$!@/^\^;:|%.\\?//{}(),>]*">
7     </xs:pattern>
8   </xs:restriction>
9 </xs:simpleType>
10 </xs:element>
11 ...
12 ...
```

Εικόνα 4-54 : Περιγραφή του στοιχείου `id_user`. Αρχείο `offers.xsd`

➤ Το στοιχείο `offer_time(data-node)`(γραμμή 2, εικόνα 4-55):

- Πρέπει να παρουσιάζετε μία και μόνο μια φορά μέσα στο στοιχείο-πάτερα `offer` και πρέπει να έχει περιεχόμενο (2^η γραμμή)
- Ορίζει ένα απλό τύπο δεδομένων(`simpleType`) (3^η γραμμή)
- Ορίζει ένα σύνολο περιορισμών(`restriction`) στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `offer_time` (4^η γραμμή)
- Καθορίζει το πρότυπο(`pattern`) του περιορισμού που θα εφαρμοστεί στον αλφαριθμητικό τύπο δεδομένων του στοιχείου `offer_time`. (5^η, 6^η γραμμή)

Όταν γίνετε μια προσφορά από ένα πελάτη, η ώρα που έγινε η προσφορά αποθηκεύεται στο στοιχείο `offer_time` στην εξής μορφή: YY-MM-DD HH-MM-SS π.χ. 2015-09-26 22:00:37. Το πρότυπο που εφαρμόζεται στη πέμπτη και έκτη γραμμή ακολουθεί το συγκεκριμένο πρότυπο ώρας(YY-MM-DD HH-MM-SS).

```
1 ...
2 <xs:element name="offer_time" minOccurs="1">
3   <xs:simpleType>
4     <xs:restriction base="xs:string">
5       <xs:pattern value="[0-9][0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]
6         [0-9][0-9]:[0-9][0-9]:[0-9][0-9]"
7     />
8   </xs:restriction>
9 </xs:simpleType>
10 </xs:element>
11 ...
```

Εικόνα 4-55 : Περιγραφή του στοιχείου `offer_time`. Αρχείο `offers.xsd`

Το αρχείο `offers.xsd` βρίσκετε στο φάκελο `store\connections`, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

Συνοπτικά, μέσω του συνδυασμού των παραπάνω XML Schemas που περιγράψαμε, καταφέραμε να επιτύχουμε τα εξής:

- Περιγράψαμε τον τύπο δεδομένων που πρέπει να έχει το κάθε στοιχείο
- Περιορίσαμε τον τύπο δεδομένων του κάθε στοιχείου σύμφωνα με τις απαιτήσεις μας
- Προσδιορίσαμε μοναδικά το περιεχόμενο συγκεκριμένων στοιχείων μέσω του στοιχείου κλειδιού
- Ορίσαμε τα κατάλληλα στοιχεία-κλειδιά για να υλοποιήσουμε μέσω PHP κώδικα ένα ασφαλές και γρήγορο σύστημα αποθήκευσης δεδομένων σε XML έγγραφα
- Ορίσαμε ότι ένας πελάτης έχει δικαίωμα να λάβει μέρος σε μία δημοπρασία ανά χρονική στιγμή
- Ορίσαμε ότι μια προσφορά γίνεται από ένα πελάτη και αφορά ένα μόνο προϊόν
- Ορίσαμε ότι κάθε δημοπρασία αναφέρεται σε ένα και μόνο ένα προϊόν
- Περιορίσαμε τις αριθμητικό πεδίο τιμών για συγκεκριμένα στοιχεία που περιέχουν αριθμητικό περιεχόμενο

Για την δημιουργία των σχημάτων που αποτυπώνονται στις εικόνες 4-20, 4-28, 4-39 και 4-48 χρησιμοποιήθηκε το εργαλείο Oxygen XML Editor έκδοση 16.

4.4 Δυνατότητες χρήστη

Το ηλεκτρονικό κατάστημα για online δημοπρασίες και πωλήσεις υποστηρίζει τις εξής δυνατότητες για ένα χρήστη:

- Εγγραφή χρήστη
- Συμμετοχή σε μια πώληση προϊόντος μέσω δημοπρασίας σε πραγματικό χρόνο
- Δυνατότητα αναζήτησης δημοπρασιών με βάση το όνομα του προϊόντος, την κατηγορία του προϊόντος και την εταιρία στην οποία ανήκει το προϊόν

4.4.1 Εγγραφή χρήστη

Μέσω της ιστοσελίδας εγγραφή χρήστη, παρέχεται η δυνατότητα σε ένα επισκέπτη να γίνει μέλος του ιστότοπου μας. Κάνοντας εγγραφή ένας χρήστης, θα μπορεί να πάρει μέρος σε δημοπρασίες πώλησης προϊόντων και να κάνει αναζήτηση δημοπρασιών στη σελίδα μας. Η φόρμα αποστολής προσωπικών δεδομένων χρήστη αποτελείται από τα εξής πεδία:

1. Πεδίο Username
2. Πεδίο Password
3. Πεδίο Password verification
4. Πεδίο Name
5. Πεδίο address
6. Πεδίο Phone
7. Πεδίο Credit card number

Εικόνα 4-56 : Ιστοσελίδα εγγραφής χρήστη

Τα πεδία αυτά αντιστοιχούν στα στοιχεία του αρχείου pelates.xml που αναλύσαμε στο υποκεφάλαιο 4.2. Εφόσον ο χρήστης έχει δώσει ορθά δεδομένα στα πεδία της φόρμας, τα αποθηκεύουμε στα αντίστοιχα στοιχεία του αρχείου pelates.xml μέσω κατάλληλων συναρτήσεων PHP. Το πεδίο password verification δεν αποτελεί στοιχείο του έγγραφου pelates.xml και χρησιμοποιείται μόνο για την επιβεβαίωση του σωστού κωδικού που πληκτρολόγησε ο χρήστης στο πεδίο password.

Η διαδικασία ελέγχου των προσωπικών δεδομένων που έδωσε ένας χρήστης στη φόρμα εγγραφής(εικόνα 4-56), λειτουργεί από την πλευρά του διακομιστή και αποτελείται από δύο διαδικασίες:

➤ Πρώτη διαδικασία ελέγχου:

Η πρώτη διαδικασία αφορά τον έλεγχο των δεδομένων που έστειλε ο χρήστης μέσω PHP κώδικα. Για την ασφαλή και ορθή λήψη δεδομένων προς αποθήκευση στο αρχείο pelates.xml υλοποιούνται οι εξής κανόνες:

- Όλα τα πεδία που είναι σημειωμένα με αστερίσκο πρέπει να είναι συμπληρωμένα από το χρήστη
- Δεν πρέπει ο χρήστης να δώσει κάποιο username που ήδη υπάρχει στο σύστημα αποθήκευσης pelates.xml
- Το username δεν πρέπει να περιέχει κενά
- Το πεδίο password πρέπει να είναι ίσο ή μεγαλύτερο των 6 χαρακτήρων
- Το πεδίο password verification πρέπει να ταιριάζει με το πεδίο password

➤ Δεύτερη διαδικασία ελέγχου:

Εφόσον τα δεδομένα που έστειλε ο χρήστης περάσουν επιτυχώς από την πρώτη διαδικασία ελέγχου τότε περνούν στην δεύτερη διαδικασία η οποία αφορά τον έλεγχο εγκυρότητας του συστήματος αποθήκευσης pelates.xml με βάση τους κανόνες που ορίζει το αντίστοιχο XML Schema pelates.xsd (αναλύθηκε στο υποκεφάλαιο 4.3). Για να μπορέσουμε να κάνουμε τον έλεγχο με ασφάλεια χωρίς να προκαλέσουμε αλλαγές που δεν θέλουμε στο αρχείο pelates.xml, αποθηκεύουμε τα δεδομένα πρώτα σε ένα προσωρινό αρχείο το οποίο ονομάζεται temp.xml. Στη συνέχεια ελέγχουμε το προσωρινό αρχείο με το αντίστοιχο XML Schema pelates.xsd και εφόσον κριθεί έγκυρο τότε αποθηκεύουμε τα ορθά δεδομένα στο αρχείο pelates.xml. Η δεύτερη διαδικασία ελέγχου παρέχει ένα επιπλέον επίπεδο προστασίας στο σύστημά αποθήκευσής μας, παρέχοντας τους εξής κανόνες:

- Όλα τα πεδία που είναι σημειωμένα με αστερίσκο πρέπει να είναι συμπληρωμένα από το χρήστη
- Δεν επιτρέπεται να υπάρχουν username με το ίδιο όνομα
- Το πεδίο username μπορεί να αποτελείται από τρεις έως και δεκαέξι χαρακτήρες
- Δεν επιτρέπεται η εισαγωγή ειδικών χαρακτήρων σε όλα τα πεδία που αντιστοιχούν username, name, address, phone, Credit card number
- Το πεδίο name μπορεί να αποτελείται από τρεις έως και δεκαέξι χαρακτήρες
- Το πεδίο password μπορεί να αποτελείται από έξι έως και τριάντα δύο χαρακτήρες
- Το πεδίο phone μπορεί να έχει μέχρι και δέκα τρεις χαρακτήρες
- Το πεδίο Credit card number μπορεί να αποτελείται από τρεις έως και τριάντα δύο χαρακτήρες

4.4.2 Συμμετοχή χρήστη σε μια δημοπρασία

Για να μπορέσει ένας εγγεγραμμένος χρήστης να συμμετέχει σε μία δημοπρασία θα πρέπει πρώτα να επισκεφτεί την σελίδα http://localhost/store/auctions_display.php και να επιλέξει από τον

πίνακα δημοπρασιών την αντίστοιχη δημοπρασία που τον ενδιαφέρει. Ένα στιγμιότυπο της σελίδας παρουσιάζεται παρακάτω για την περαιτέρω ανάλυση της.

The screenshot shows the 'Super Auctions' website interface. At the top, there is a navigation menu with 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock displays '8:54:40pm' and the date 'October 07, 2015'. The main content area is titled 'Auctions' and features a table with 10 rows of auction data. To the right of the table is a sidebar with a user greeting 'Hello, Takis!', a 'Logout' link, and a 'Users' section stating 'We have 5 registerd user.' (note the typo).

A/A	Product Name	Price(€)	Start Time	Corpration	Category
1	A1	3063	T17:23 27-09-2015	Audi	Cars
2	L200	10700	T15:01 20-10-2015	Mitsubishi	Cars
3	L400	15700	T20:00 01-02-2013	Mitsubishi	Cars
4	Compaq 6530g	345	T18:43 23-07-2015	Hewlett Packard	Computers
5	230g	340	T23:34 26-09-2015	Dell	Computers
6	N300g	700	T21:15 13-11-2015	Dell	Computers
7	N400g	498.50	T20:43 10-10-2015	Dell	Computers
8	G-Shock XL	130.60	T20:24 27-12-2015	Casio	Watches
9	New Melenioum	100	T20:00 18-10-2015	Casio	Watches
10	Vogue City	350.60	T16:00 19-09-2015	Casio	Watches

Εικόνα 4-57 : Ιστοσελίδα παρουσίασης δημοπρασιών

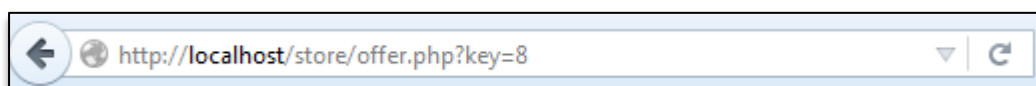
Η ιστοσελίδα `auctions_display.php` παρουσιάζει ένα πίνακα με τις δημοπρασίες που πρόκειται να λάβουν μέρος στο μέλλον. Ο πίνακας αντλεί πληροφορίες από το αρχείο `auction_products.xml` του οποίου το περιεχόμενο ανανεώνεται κάθε φορά που ο χρήστης μπαίνει ή ανανεώνει την σελίδα. Οι λειτουργίες, η περιγραφή και ο τρόπος παρουσίασης των στοιχείων του εγγράφου `auction_products.xml` αναλύεται στο κεφάλαιο 4.2.

Ο πίνακας που εμφανίζεται στην εικόνα 4-57 είναι αποτέλεσμα του XSLT μετασχηματισμού, του εγγράφου `auction_products.xml` σε HTML μορφή. Η ενέργεια αυτή υλοποιείται μέσω PHP συνάρτησης που μετασχηματίζει το έγγραφο `auction_products.xml`, σύμφωνα με το μοντέλο μετασχηματισμού που ορίζει το έγγραφο `auction_products.xsl`. Τα δύο αυτά αρχεία βρίσκονται στο φάκελο `store\offers`, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

Η διαδικασία του XSLT μετασχηματισμού παρέχει τις εξής δυνατότητες στην έξοδο του νέου εγγράφου:

- Παρουσίαση των στοιχείων του XML εγγράφου σε HTML μορφή
- Δημιουργία HTML πίνακα για την παρουσίαση του περιεχομένου των στοιχείων που συνθέτουν το έγγραφο `auction_products.xml`
- Επιλογή των στοιχείων που θα εμφανίζονται στον πίνακα
- Ταξινόμηση των στοιχείων που εμφανίζονται στον πίνακα με βάση την κατηγορία προϊόντος
- Δημιουργία κατάλληλων συνδέσμων(link) για την μετάβαση πληροφοριών μέσω της URL διεύθυνσης, στην σελίδα `offer.php`
- Γραφική εμφάνιση του περιεχομένου των στοιχείων, μέσω CSS μορφοποίησης

Όταν ένας χρήστης επιλέξει σε μια δημοπρασία θέλει να συμμετέχει από τον πίνακα δημοπρασιών, δημιουργείτε αυτόματα το κατάλληλο URL της ιστοσελίδα που θα φιλοξενήσει την συγκεκριμένη δημοπρασία. Ένα παράδειγμα μίας τέτοιας διεύθυνσης είναι το εξής:



Εικόνα 4-58 : URL διεύθυνση μιας συγκεκριμένης δημοπρασίας

Όπου η σύνταξη της URL διεύθυνσης αποτελείται από δυο πληροφορίες:

1. Την διεύθυνση του αρχείου που παρέχει την βασική δομή σχεδίασης της ιστοσελίδας που θα γίνει μία δημοπρασία. Πάντα το πρώτο μέρος της URL διεύθυνσης μίας δημοπρασίας θα είναι `http://localhost/store/offer.php`.
2. Το κλειδί(`?key=`"τιμή κλειδιού"`")` που προσδιορίζει την δημοπρασία που επέλεξε να συμμετέχει ο χρήστης, Το κλειδί αυτό προσδιορίζει μονοσήμαντα μια δημοπρασία και η τιμή του εξαρτάτε πάντα από την επιλογή του χρηστή μέσα από το πίνακα δημοπρασιών. Μέσω του κλειδιού αυτού δημιουργείτε ένα στιγμιότυπο της σελίδας `http://localhost/store/offer.php` με διεύθυνση `http://localhost/store/offer.php?key=8` (εικόνας 4-58). Η τελική διεύθυνσης URL που σχηματίστηκε αποτελεί την ιστοσελίδα που θα φιλοξενήσει την δημοπρασία που επέλεξε ο χρήστης.

Σύμφωνα με το παράδειγμα της εικόνας 4-58, όταν ένας χρήστης μπει στην σελίδα `http://localhost/store/offer.php?key=8`, αυτόματα μέσω της τιμής του κλειδιού(`?key=8`) που παρέχεται στη URL διεύθυνση, θα φορτωθούν τα δεδομένα του κόμβου με την αντίστοιχη τιμή κλειδιού από το αρχείο `auction_products.xml`. Το αρχείο που παρέχει την βασική δομή σχεδίασης της σελίδας `http://localhost/store/offer.php`, περιλαμβάνει τον κατάλληλο PHP κώδικα για την ανάκτηση πληροφοριών από το σύστημα αποθήκευσης XML αρχείων, σύμφωνα με την τιμή του κλειδιού. Παράλληλα ο συνδυασμός κώδικα Javascript και Ajax παρέχει στην σελίδα μία αμφίδρομη επικοινωνία με το χρήστη, παρουσιάζοντας με ρεαλιστικό τρόπο ένα συνεχώς ανανεωμένο περιεχόμενο. Το στιγμιότυπο της σελίδα `http://localhost/store/offer.php?key=8` που ακολουθεί, παρουσιάζει τη φόρμα μαζί με όλα τα πεδία που υλοποιούν το περιβάλλον διεξαγωγής μίας δημοπρασίας.

The screenshot shows the 'Super Auctions' website interface. At the top, there is a navigation menu with 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock displays '10:20:31pm' and the date 'October 09, 2015'. The main content area features an auction for a 'G-Shock XL' watch. The auction will start in '1 M. 0 D. 12 H. 40 M. 24 S'. The start price is '130.60 euros' and the corporation is 'Casio'. A description box contains the text: 'Product description. We can load here the product description for current auction'. Below the description is a form for giving an offer, with a 'Submit Offer' button. A user greeting 'Hello, Takis!' and a 'Logout' link are visible on the right. A 'Users' section indicates 'We have 5 registerd user.' Numbered arrows (1-6) point to various elements: 1 points to the auction title, 2 to the countdown timer, 3 to the start price, 4 to the corporation name, 5 to the offer input field, and 6 to the 'Submit Offer' button.

Εικόνα 4-59 : Στιγμιότυπο δημοπρασίας πριν την έναρξη της

Όπου:

1. Το πεδίο που εμφανίζει το όνομα του προϊόντος σε μία δημοπρασία

2. Το πεδίο που εμφανίζει το χρονοδιακόπτη μίας δημοπρασίας

Ο χρονοδιακόπτης δημοπρασίας είναι προγραμματισμένος για να παρέχει τρεις διαφορετικές λειτουργίες χρονομέτρου για την ομαλή διεξαγωγή, εκκίνηση και λήξη μίας δημοπρασίας:

- I. Υπολογίζει το χρονικό διάστημα που απομένει για να ξεκινήσει μια δημοπρασία. Το χρονικό διάστημα υπολογίζεται και παρουσιάζεται μέσω ενός χρονομέτρου σε μήνες, μέρες, ώρες, λεπτά και δευτερόλεπτα(εικόνα 4-59).
- II. Θέτει ένα δεκάλεπτο χρονόμετρο που μετρά αντίστροφα κατά εκκίνηση μιας δημοπρασίας. Όταν τελειώσει η αντίστροφη μέτρηση που υπολογίζετε για την εκκίνηση μιας δημοπρασίας ο χρονοδιακόπτης αλλάζει λειτουργία και θέτει ένα νέο μετρητή που μετρά αντίστροφα για δέκα λεπτά. Ο ρόλος του χρονομέτρου εξαρτάτε άμεσα από την προσφορά ενός χρήστη και στην περίπτωση που δεν έχει γίνει καμία προσφορά μέσα στα δέκα πρώτα λεπτά τότε η δημοπρασία θα λήξει με το τέλος του μετρητή. Στην περίπτωση που γίνει μια προσφορά μέσα στα δέκα λεπτά, η λειτουργία του χρονοδιακόπτη θα αλλάξει με το τέλος του μετρητή.
- III. Θέτει ένα πεντάλεπτο χρονόμετρο που μετρά αντίστροφα, για κάθε παράταση της διάρκειας διεξαγωγής μίας δημοπρασίας. Η τρίτη λειτουργία του χρονοδιακόπτη ξεκινά στο τέλος του δεκάλεπτου χρονομέτρου και μόνο στην περίπτωση που έχει γίνει μια προσφορά από ένα χρήστη μέσα στο χρόνο αυτό. Στο τέλος της αντίστροφης μέτρησης του πεντάλεπτου χρονομέτρου εφόσον ένας χρήστης δεν έχει δώσει άλλη προσφορά ο χρονοδιακόπτης λήγει την δημοπρασία. Στην περίπτωση όμως που έχει δοθεί έστω και μια προσφορά μέσα στα πέντε λεπτά που ορίζει η τρίτη λειτουργία του χρονοδιακόπτη, στο τέλος του μετρητή το χρονόμετρο επανακαθορίζεται με άλλα πέντε λεπτά, δίνοντας παράταση στο χρόνο διάρκειας μίας δημοπρασίας. Η πεντάλεπτη παράταση επανακαθορίζεται για κάθε νέα προσφορά που γίνεται από ένα χρήστη στο τέλος του μετρητή.

3. Το πεδίο που εμφανίζει την τιμή του προϊόντος σε μία δημοπρασία

Η λειτουργία παρουσίασης της τιμής ενός προϊόντος χωρίζετε σε τρεις χρονικές καταστάσεις:

- I. Το πεδίο τιμής παρουσιάζει την αρχική τιμή του προϊόντος πριν την εκκίνηση μίας δημοπρασίας. Η τιμή του πεδίου δεν μπορεί να αλλάξει αν δεν ξεκινήσει δημοπρασίας.
- II. Το πεδίο τιμής παρουσιάζει την τιμή της τελευταίας μεγαλύτερης προσφοράς που έγινε από ένα χρήστη κατά την διάρκεια διεξαγωγής μίας δημοπρασίας. Εάν δεν έχει γίνει καμία προσφορά που να είναι μεγαλύτερη της αρχικής τιμής κατά την διεξαγωγή μιας δημοπρασίας, τότε το πεδίο παρουσιάζει την τιμή του προϊόντος σύμφωνα με την αρχική τιμή του περιεχομένου του στοιχείου start_price, από το αρχείο αποθήκευσης δημοπρασιών auctions.xml. Η ανανέωση της τιμής προϊόντος κατά την διεξαγωγή μίας δημοπρασίας γίνεται ανά 20 δευτερόλεπτα μέσω αιτήματος AJAX, προς το σύστημα αποθήκευσης αρχείων XML. Επίσης το πεδίο τιμής προϊόντος ανανεώνετε σε κάθε νέα προσφορά που γίνεται από ένα χρήστη. Οι προσφορές που είναι μικρότερες τις αρχικής τιμής ή της τρέχουσας τιμής δεν εμφανίζονται στο χρήστη παρά μόνο αποθηκεύονται.
- III. Το πεδίο τιμής παρουσιάζει την τελική τιμή του προϊόντος στην λήξη μίας δημοπρασίας. Η τελική τιμή του προϊόντος είναι ίση με την αρχική τιμή προϊόντος, στην περίπτωση που μια δημοπρασία λήξει άγονη. Στην περίπτωση που το προϊόν πουληθεί σε κάποιο χρήστη τότε η τελική τιμή προϊόντος θα είναι ίση με την μεγαλύτερη προσφορά του χρήστη που αγόρασε το προϊόν.

4. Το πεδίο που εμφανίζει την εταιρία του προϊόντος σε μία δημοπρασία

5. Το πεδίο εισαγωγής μίας προσφοράς

Μέσω του πεδίου εισαγωγής μίας προσφοράς, ένας χρήστης έχει την δυνατότητα να πληκτρολογήσει τα χρήματα που ποντάρει για το προϊόν μίας δημοπρασίας. Στο πεδίο δεν επιτρέπονται γράμματα και ειδικοί χαρακτήρες, παρά μόνο αριθμητικές τιμές.

6. Το κουμπί της φόρμας για την υποβολή μίας προσφοράς

Ένας χρήστης έχει την δυνατότητα να υποβάλει μια προσφορά, μέσω του κουμπιού Submit Offer ή πατώντας το πλήκτρο enter από το πληκτρολόγιο. Κατά την διαδικασία υποβολής μίας προσφοράς, αυτόματα μέσω PHP κώδικα που είναι υπεύθυνος για την διαχείριση του συστήματος αποθήκευσης XML αρχείων, αποθηκεύετε η προσφορά μέσα στο αρχείο offers.xml. Ένας χρήστης έχει δικαίωμα να υποβάλει μία ή περισσότερες προσφορές πριν την έναρξη και κατά την διάρκεια εξέλιξης μιας δημοπρασίας. Μετά την λήξη μίας δημοπρασίας το κουμπί γίνεται ανενεργό και οι χρήστες δεν έχουν δικαίωμα να ποντάρουν άλλες τιμές.

Συνοπτικά για μια δημοπρασία και τις δυνατότητες συμμετοχής ενός χρήστη:

- 1.** Πριν την εκκίνηση μιας δημοπρασίας, ένας χρήστης θα έχει την δυνατότητα να μπει στην αντίστοιχη σελίδα δημοπρασίας που θέλει να συμμετέχει και κάνει μια ή περισσότερες προσφορές. Οι προσφορές οι οποίες γίνονται πριν την έναρξη μια δημοπρασίας δεν θα εμφανίζονται στους χρήστες παρά μόνο η αρχική τιμή προϊόντος που έχει οριστεί από το διαχειριστή του συστήματος. Μέσω του χρονομέτρου εκκίνησης μιας δημοπρασίας που παρουσιάζετε οι χρήστες μπορούν να έχουν γνώση του εναπομείναντα χρόνου για την έναρξη της δημοπρασίας που θέλουν να συμμετέχουν.
- 2.** Κατά την εκκίνηση μίας δημοπρασίας γίνεται αυτόματα ανανέωση της τιμής του προϊόντος σύμφωνα με τις προσφορές των χρηστών. Αν δεν έχει γίνει καμία μεγαλύτερη πρόσφορα τότε αυτόματα γίνεται ανανέωση της τιμής του προϊόντος σύμφωνα με την αρχική τιμή. Επίσης μετά την έναρξη μίας δημοπρασίας οι χρήστες έχουν την δυνατότητα να δουν τις αλλαγές στις τιμές των προϊόντων από τον πίνακα δημοπρασιών στην σελίδα http://localhost/store/auctions_display.php (εικόνα 4-57) ή από την φόρμα διεξαγωγής μίας δημοπρασίας(εικόνα 4-60).
- 3.** Για την εκκίνηση, διεξαγωγή και παράταση του χρόνου εξέλιξης μιας δημοπρασίας χρησιμοποιείτε ένα χρονόμετρο από την πλευρά του χρήστη για να συντονίζει τις λειτουργίες της σελίδας σύμφωνα με τις προσφορές που γίνονται. Η εκκίνηση μίας δημοπρασίας θέτει ένα δεκάλεπτο χρονόμετρο(εικόνα 4-60) που μετρά αντίστροφα μέχρι το μηδέν. Στην περίπτωση που δεν έχει γίνει καμία προσφορά χρήστη με μεγαλύτερη τιμή από την αρχική, πριν και μετά την έναρξη μίας δημοπρασίας για τουλάχιστον δέκα λεπτά, όταν το χρονόμετρο φτάσει στο μηδέν η δημοπρασία λήγει. Στην περίπτωση που έχει γίνει μια προσφορά χρήστη με μεγαλύτερη τιμή από την αρχική, όταν το χρονόμετρο φτάσει στο μηδέν επανακαθορίζεται χρονικά για πέντε λεπτά και ο χρόνος δημοπρασία παρατείνεται(εικόνα 4-61). Κάθε νέα προσφορά χρήστη με μεγαλύτερη τιμή από την τρέχουσα τιμή μέσα στο διάστημα αυτό, επανακαθορίζει χρονικά το χρονόμετρο για άλλα πέντε λεπτά όταν ο μετρητής φτάσει στο μηδέν.
- 4.** Μια δημοπρασία λήγει με κατακύρωση εφόσον έχει γίνει έστω και μία προσφορά χρήστη με μεγαλύτερη τιμή από τη αρχική τιμή προϊόντος, για το χρονικό διάστημα πριν και μετά την έναρξη μίας δημοπρασίας, για τουλάχιστον δέκα λεπτά.
- 5.** Ένας χρήστης Α κερδίζει μια δημοπρασία έναντι των άλλων χρηστών εφόσον:
 - Δεν έχει γίνει καμία μεγαλύτερη προσφορά χρήστη από την τελευταία μεγαλύτερη προσφορά του χρήστη Α, για το χρονικό διάστημα πριν και μετά την έναρξη μίας δημοπρασίας, για τουλάχιστον δέκα λεπτά.

- Δεν έχει γίνει καμία μεγαλύτερη προσφορά χρήστη από την τελευταία μεγαλύτερη προσφορά του χρήστη A, μέσα στο χρονικό διάστημα μίας παράτασης δημοπρασίας και για τουλάχιστον πέντε λεπτά.
- 6.** Μια δημοπρασία λήγει άγωνα εφόσον δεν έχει γίνει καμία προσφορά χρήστη με μεγαλύτερη τιμή από την αρχική τιμή προϊόντος, για το χρονικό διάστημα πριν και μετά την έναρξη μίας δημοπρασίας, για τουλάχιστον δέκα λεπτά.

Στην συνέχεια παρουσιάζονται τρία στιγμιότυπα της φόρμας διεξαγωγής μίας δημοπρασίας που περιγράφουν την έναρξη, την παράταση και τη λήξη μίας δημοπρασίας.

- Έναρξη δημοπρασίας

Εικόνα 4-60 : Στιγμιότυπο δημοπρασίας μετά την έναρξη. Χρονόμετρο δέκα λεπτών

- Παράταση δημοπρασίας

Super Auctions

Home Auctions Search Contact us **12:10:06am** October 11, 2015

Auction for G-Shock XL

Auction started

0M.0D.0H.4M.54S

Start Price : **131** euros

Corporation : Casio

Description:

Product description. We can load here the product decription for current auction

Give your offer please:

Hello, Takis!

Logout

Users

We have 5 registerd user.

Εικόνα 4-61 : Παράταση χρόνου δημοπρασίας. Χρονόμετρο πέντε λεπτών

- Λήξη δημοπρασίας

Super Auctions

Home Auctions Search Contact us **12:15:17am** October 11, 2015

Auction for G-Shock XL

Auction finished

0M.0D.0H.0M.0S

Product sold for : **131** euros

Corporation : Casio

Description:

Product description. We can load here the product decription for current auction

Give your offer please:

Hello, angell!

Logout

Users

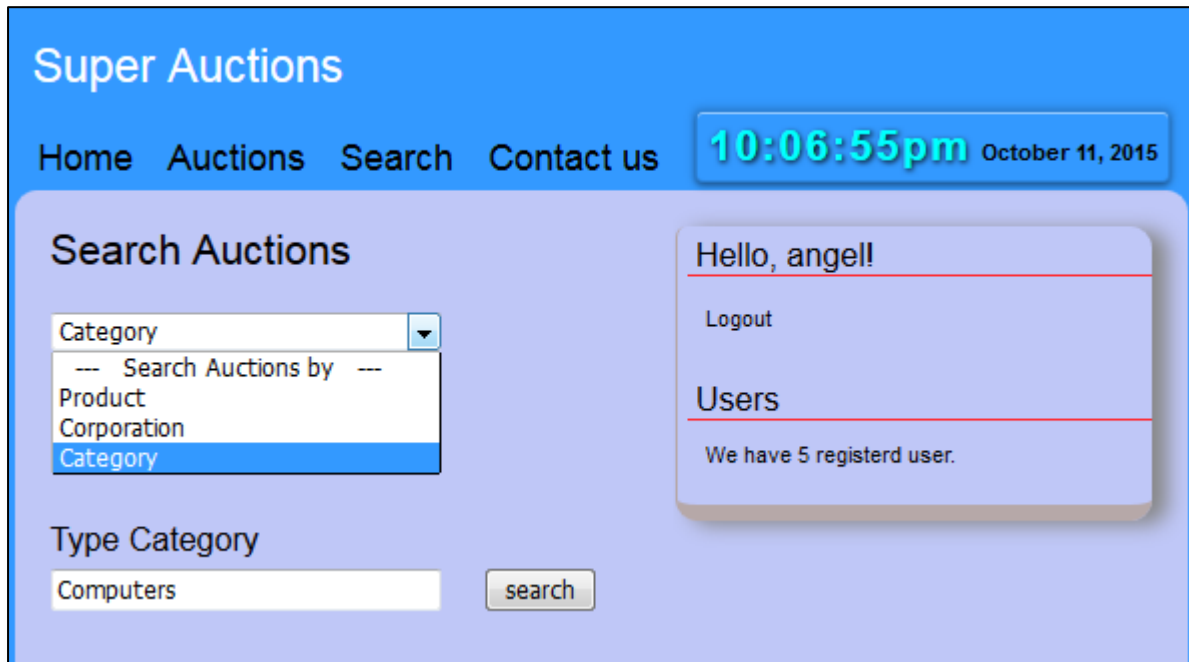
We have 5 registerd user.

Εικόνα 4-62 : Στιγμιότυπο λήξης μίας δημοπρασίας

4.4.3 Διαδικασία αναζήτησης δημοπρασιών

Ένας χρήστης ο οποίος έχει κάνει εγγραφή στη σελίδα μας, έχει την δυνατότητα να κάνει αναζήτηση δημοπρασιών με βάση τριών κριτηρίων:

- Αναζήτηση δημοπρασίας με βάση το όνομα προϊόντος
- Αναζήτηση δημοπρασίας με βάση τη κατηγορία προϊόντος
- Αναζήτηση δημοπρασίας με βάση την εταιρία προϊόντος



Εικόνα 4-63 : Ιστοσελίδα αναζήτησης δημοπρασιών

Η φόρμα αναζήτησης δημοπρασιών όπως φαίνεται και στην εικόνα 4-63, αποτελείται από τρία βασικά στοιχεία:

1. Το πεδίο αναδυόμενου μενού/λίστα(drop down menu/list) το οποίο χρησιμοποιείται για την επιλογή συγκεκριμένων κριτηρίων αναζήτησης που παρέχει το σύστημα
2. Το πεδίο εισαγωγής κριτηρίων αναζήτησης χρήστη.
 - Το κείμενο αναζήτησης που πληκτρολογεί ένας χρήστης μπορεί να αποτελείται από ένα έως και δέκα έξι χαρακτήρες
 - Το κείμενο αναζήτησης που πληκτρολογεί ένας χρήστης δεν πρέπει να περιλαμβάνει ειδικούς χαρακτήρες όπως %, +, =, ; κ.τ.λ..
3. Το κουμπί για την υποβολή μίας αναζήτησης στο σύστημα

Για αν μπορέσει ένας χρήστης να κάνει μία αναζήτηση δημοπρασίας, θα πρέπει πρώτα να συμπληρώσει την φόρμα που παρουσιάζεται στην εικόνα 4-63 και στην συνέχεια να πατήσει το κουμπί υποβολής μίας αναζήτησης. Με την υποβολή της φόρμας στο σύστημα, ακολουθούν οι εξής προγραμματισμένες ενέργειες από την πλευρά του διακομιστή για την εμφάνιση των αποτελεσμάτων αναζήτησης στην οθόνη του χρήστη:

1. Ελέγχετε μέσω PHP κώδικα η ορθότητα των δεδομένων που πληκτρολόγησε ο χρήστης στη φόρμα αναζήτησης, πριν τα δεδομένα αποθηκευτούν στο αρχείο search.xml (κεφάλαιο 4-2)
2. Ενημερώνετε το αρχείο search.xml με τα δεδομένα που πληκτρολόγησε ο χρήστης (κεφάλαιο 4.2)
3. Γίνετε έλεγχος των δεδομένων του αρχείου search.xml σύμφωνα με το αντίστοιχο αρχείο XML Schema search.xsd
4. Εφόσον τα δεδομένα αναζήτησης που υπέβαλε στο σύστημα ένας χρήστης είναι έγκυρα, σύμφωνα με τους κανόνες που ορίζει το αντίστοιχο XML Schema οδηγούμαστε στην σελίδα http://localhost/store/search_results.php
5. Γίνετε αναζήτηση της δημοπρασίας στο σύστημα αποθήκευσης XML αρχείων σύμφωνα με τα κριτήρια του χρήστη που είναι αποθηκευμένα στο αρχείο search.xml
6. Αποθηκεύονται τα αποτελέσματα αναζήτησης στο αρχείο search_results.xml
7. Μετασχηματίζετε το αρχείο search_results.xml με το XSLT έγγραφο style.xsl
8. Τα αποτελέσματα του μετασχηματισμού παρουσιάζονται στο χρήστη σε HTML μορφή, εμφανίζοντας τα αποτελέσματα της αναζήτησης του χρήστη μέσα σε ένα πινάκα
9. Ο χρήστης μπορεί να επιλέξει οποιαδήποτε δημοπρασία στην οποία θέλει να συμμετέχει από εμφανιζόμενο πίνακα

Super Auctions

Home Auctions Search Contact us

10:07:04pm October 11, 2015

Search Results

A/A	Category	Corporation	Product Name	Time Start	Start Price(€)
1	Computers	Hewlett Packard	Compaq 6530g	T18:43 23-07-2015	345
2	Computers	Dell	230g	T23:34 26-09-2015	340
3	Computers	Dell	N300g	T21:15 13-11-2015	700
4	Computers	Dell	N400g	T20:43 10-10-2015	498.50

Hello, angel!

Logout

Users

We have 5 registered user.

Εικόνα 4-64 : Αποτελέσματα μίας αναζήτησης δημοπρασιών

Το XSLT αρχείο μετασχηματισμού style.xsl βρίσκεται στο φάκελο store\search, από το πακέτο αρχείων που αποτελούν το ηλεκτρονικό μας κατάστημα.

4.5 Δυνατότητες διαχειριστή συστήματος

Το ηλεκτρονικό κατάστημα για online δημοπρασίες και πωλήσεις υποστηρίζει ένα χρήστη-διαχειριστή με τις εξής δυνατότητες:

- Εισαγωγή νέων προϊόντων
- Εισαγωγή νέων δημοπρασιών
- Επίβλεψη συγκεντρωτικών αναφορών (page reports)

Για να συνδεθούμε ως διαχειριστής στο σύστημα για online δημοπρασίες και πωλήσεις πληκτρολογούμε για username το όνομα petros και για κωδικό βάζουμε το 12345678. Εφόσον συμπληρώσουμε σωστά τα πεδία στη σελίδα index θα εμφανιστεί το κατάλληλο μήνυμα που μας ενημερώνει ότι έχουμε συνδεθεί ως διαχειριστές.

4.5.1 Εισαγωγή νέων προϊόντων

Ο διαχειριστής του συστήματος έχει την δυνατότητα να εισάγει νέα προϊόντα στο σύστημα αποθηκεύσεις του ιστότοπου. Τα νέα προϊόντα καταχωρούνται στο αρχείο products.xml το οποίο αναλύεται στο κεφάλαιο 4.2. Η φόρμα εισαγωγής νέων προϊόντων, όπως φαίνεται παρακάτω στην εικόνα 4-65, αποτελείται από πέντε βασικά στοιχεία:

1. Πεδίο Product Name
2. Πεδίο Corporation
3. Πεδίο Price
4. Πεδίο Category
5. Το κουμπί αποστολής δεδομένων

Τα πεδία Product Name, Corporation και Price, απαιτούν από το διαχειριστή του συστήματος να πληκτρολογήσει τα δεδομένα που θέλει να αποθηκεύσει. Το πεδίο Category έχει τη μορφή μίας αναδύμενης λίστας με προκαθορισμένες τιμές και εμφανίζει τις εξής επιλογές:

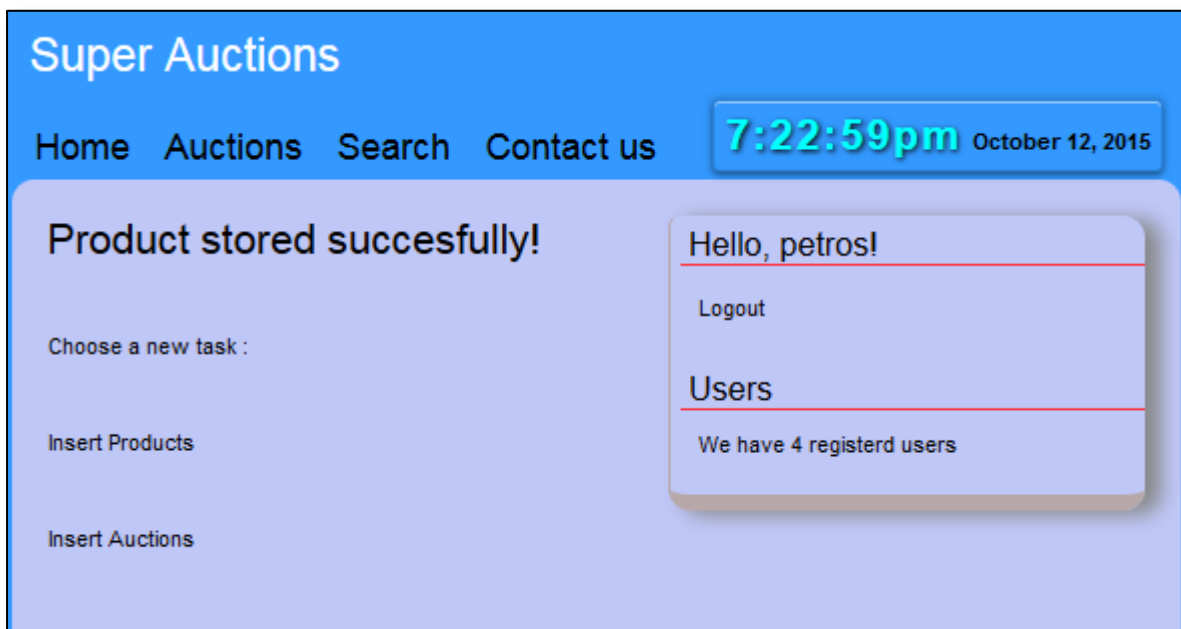
1. Computers
2. Cars
3. Watches
4. Bikes
5. Art
6. Clothes

The screenshot shows a web interface for 'Super Auctions'. The top navigation bar is blue and contains links for 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock displays '6:12:45pm' and the date 'October 12, 2015'. The main content area is light blue and features a section titled 'Administrator Privileges'. Under this section, there is a form titled 'Insert Product' with the following fields: 'Product Name' (text input), 'Corporation' (text input), 'Price' (text input), and 'Category' (dropdown menu with '--- Select Category ---'). An 'Insert Product' button is located at the bottom of the form. To the right of the form, there is a user greeting 'Hello, petros!' with a 'Logout' link below it, and a 'Users' section showing 'We have 4 registerd users'.

Εικόνα 4-65 : Ιστοσελίδα εισαγωγής νέων προϊόντων. Δικαιώματα διαχειριστή

Με την υποβολή της φόρμας εισαγωγής ενός νέου προϊόντος στο σύστημα, ακολουθούν οι εξής προγραμματισμένες ενέργειες από την πλευρά του διακομιστή:

1. Ελέγχετε μέσω PHP κώδικα η ορθότητα των δεδομένων που πληκτρολόγησε ο διαχειριστής του συστήματος στη φόρμα εισαγωγής νέων προϊόντων, πριν τα δεδομένα αποθηκευτούν στο αρχείο products.xml (κεφάλαιο 4-2)
2. Αποθηκεύονται τα δεδομένα που έστειλε ο διαχειριστής σε ένα προσωρινό αρχείο XML με όνομα temp_products.xml
3. Γίνετε έλεγχος των δεδομένων του αρχείου temp_products.xml σύμφωνα με το αντίστοιχο αρχείο XML Schema products.xsd (κεφάλαιο 4-3)
4. Εφόσον τα δεδομένα που έστειλε ο διαχειριστής στο σύστημα είναι έγκυρα, σύμφωνα με τους κανόνες που ορίζει το αντίστοιχο XML Schema, στη συνέχεια αποθηκεύονται τα δεδομένα στο σύστημα αποθήκευσης προϊόντων products.xml
5. Η επιτυχής εισαγωγή νέου προϊόντος στο αρχείο products.xml, οδηγεί το διαχειριστή στην σελίδα http://localhost/store/success_insert.php εμφανίζοντας το κατάλληλο μήνυμα επιτυχίας (εικόνα 4-66)



Εικόνα 4-66 : Επιτυχής εισαγωγή νέου προϊόντος στο αρχείο products.xml

4.5.2 Εισαγωγή νέων δημοπρασιών

Ο διαχειριστής του συστήματος έχει την δυνατότητα να εισάγει μία νέα δημοπρασία στο σύστημα αποθηκεύσεις του ιστότοπου. Οι νέες δημοπρασίες καταχωρούνται στο αρχείο auctions.xml το οποίο αναλύεται στο κεφάλαιο 4.2. Η φόρμα εισαγωγής νέων δημοπρασιών, όπως φαίνεται παρακάτω στην εικόνα 4-67, αποτελείται από πέντε βασικά στοιχεία:

1. Τρία πεδία που περιγράφουν την ημερομηνία της δημοπρασίας (Day-Month-Year)
2. Πεδίο Time start
3. Πεδίο Product
4. Το κουμπί αποστολής δεδομένων

Τα πεδία ημερομηνίας, Time start και Product, απαιτούν από το διαχειριστή του συστήματος να πληκτρολογήσει τα δεδομένα που θέλει να αποθηκεύσει. Το πεδίο Category έχει τη μορφή μίας αναδύομενης λίστας που παρουσιάζει τα προϊόντα του εγγράφου products.xml τα οποία δεν έχουν προσδιοριστεί από μία δημοπρασία.

Super Auctions

Home Auctions Search Contact us

7:59:55pm October 12, 2015

Administrator Privileges

Insert Auctions

Day : Month : Year :
ex : 29/08/2015

Time start :
ex : 19:30

Choose product :

--- Select product ---
--- Select product ---
Acer Aspire E-71

Insert Auction

Hello, petros!

Logout

Users

We have 4 registerd users

Εικόνα 4-67 : Ιστοσελίδα εισαγωγής νέων δημοπρασιών. Δικαιώματα διαχειριστή

Με την υποβολή της φόρμας εισαγωγής μίας νέας δημοπρασίας στο σύστημα, ακολουθούν οι εξής προγραμματισμένες ενέργειες από την πλευρά του διακομιστή:

1. Ελέγχετε μέσω PHP κώδικα η ορθότητα των δεδομένων που πληκτρολόγησε ο διαχειριστής του συστήματος στη φόρμα εισαγωγής νέων δημοπρασιών, πριν τα δεδομένα αποθηκευτούν στο αρχείο auctions.xml (κεφάλαιο 4-2)
2. Αποθηκεύονται τα δεδομένα που έστειλε ο διαχειριστής σε ένα προσωρινό αρχείο XML με όνομα temp_auctions.xml
3. Γίνετε έλεγχος των δεδομένων του αρχείου temp_auctions.xml σύμφωνα με το αντίστοιχο αρχείο XML Schema auctions.xsd (κεφάλαιο 4-3)
4. Εφόσον τα δεδομένα που έστειλε ο διαχειριστής στο σύστημα είναι έγκυρα, σύμφωνα με τους κανόνες που ορίζει το αντίστοιχο XML Schema, στη συνέχεια αποθηκεύονται τα δεδομένα στο σύστημα αποθήκευσης δημοπρασιών auctions.xml
5. Η επιτυχής εισαγωγή νέου προϊόντος στο αρχείο auctions.xml, οδηγεί το διαχειριστή στην σελίδα http://localhost/store/success_insert.php εμφανίζοντας το κατάλληλο μήνυμα επιτυχίας (εικόνα 4-68)

The screenshot shows the 'Super Auctions' website interface. At the top, there is a blue header with the site name and navigation links: Home, Auctions, Search, and Contact us. A digital clock displays '8:35:29pm' and the date 'October 12, 2015'. The main content area features a message 'Auction stored successfully!' and a section titled 'Choose a new task :'. Under this section, there are three links: 'Insert Products', 'Insert Auctions', and 'Insert Auctions'. On the right side, there is a user menu for 'petros' with a 'Logout' link and a 'Users' section indicating 'We have 4 registered users'.

Εικόνα 4-68 : Επιτυχής εισαγωγή μίας νέας δημοπρασίας στο αρχείο auction.xml

4.5.3 Συγκεντρωτικές αναφορές

Ο διαχειριστής του συστήματος έχει την δυνατότητα να επιβλέπει τις συγκεντρωτικές αναφορές που εκδίδει το σύστημα για την περιγραφή των δημοπρασιών που έχουν λήξει. Το σύστημα υλοποιεί και παρουσιάζει τις παρακάτω συγκεντρωτικές αναφορές:

1. Πόσες δημοπρασίες έληξαν άγονες
2. Πόσο είναι ο μέσος χρόνος όλων των δημοπρασιών που έχουν λήξει
3. Ποία προϊόντα δεν πουλήθηκαν λόγω άγονων δημοπρασιών, ποία ήταν η μεγαλύτερη προσφορά για το κάθε ένα προϊόν και ποία είναι η εναρκτήρια τιμή της αντίστοιχης δημοπρασίας.

The screenshot shows the 'Super Auctions' website interface at a later time, '9:17:41pm' on 'October 12, 2015'. The main content area is titled 'Reports' and displays summary statistics: 'Barren auctions : 3' and 'Average time for auctions : 0 day, 0 hour, 13 minutes, 48 seconds'. Below this, it says 'Full report for barren auctions :'. A table with 4 columns (A/A, Product Name, Start Price(€), Bigger Offer(€)) and 4 rows (including headers) is shown. The table lists three products: Aspire E1-78 (Start Price: 890, Bigger Offer: No offer), M800 (Start Price: 40000, Bigger Offer: No offer), and T600 (Start Price: 500, Bigger Offer: No offer). The user menu on the right remains the same as in the previous screenshot.

A/A	Product Name	Start Price(€)	Bigger Offer(€)
1	Aspire E1-78	890	No offer
2	M800	40000	No offer
3	T600	500	No offer

Εικόνα 4-69 : Παρουσίαση συγκεντρωτικών αναφορών συστήματος

1. Πόσες δημοπρασίες έληξαν άγονες(barren):

Το σύστημα ψάχνει και μέτρα πόσες δημοπρασίες περιγράφονται ως Barren από το στοιχείο status, στο σύστημα αποθήκευσης δημοπρασιών που έχουν λήξη finished_auctions.xml. Τα αποτελέσματα παρουσιάζονται στο διαχειριστή μέσω αιτήματος Ajax. (εικόνα 4-69)

2. Πόσο είναι ο μέσος χρόνος όλων των δημοπρασιών που έχουν λήξει:

Το σύστημα υπολογίζει το μέσο χρόνο μέσω της τιμής του στοιχείου time_ended από το αρχείο finished_auctions.xml και μέσω του στοιχείου time_start από το αρχείο auctions.xml. Τα δύο αυτά στοιχεία περιγράφουν την ώρα που τελείωσε μια δημοπρασία και την ώρα που ξεκίνησε. Όπου ο μέσος χρόνος υπολογίζεται από μαθηματικό υπολογισμό που περιγράφεται:

1. Αφαιρούμε την ώρα λήξης από την ώρα έναρξης μίας δημοπρασίας
2. Προσθέτουμε τους χρόνους κάθε δημοπρασίας
3. Διαιρούμε το συνολικό χρόνο των δημοπρασιών με τον συνολικό αριθμό των δημοπρασιών

Τα αποτελέσματα παρουσιάζονται στο διαχειριστή μέσω αιτήματος Ajax. (εικόνα 4-69)

3. Ποία προϊόντα δεν πουλήθηκαν λόγω άγονων δημοπρασιών, ποία ήταν η μεγαλύτερη προσφορά για το κάθε ένα προϊόν και ποία είναι η εναρκτήρια τιμή της αντίστοιχης δημοπρασίας:

Τα αποτελέσματα της αναφοράς για τις δημοπρασίες που έληξαν άγονες βρίσκονται στο σύστημα αποθήκευσης άγονων δημοπρασιών report_barren.xml(κεφάλαιο 4.2). Το σύστημα αποθηκεύει τις κατάλληλες πληροφορίες στο αρχείο report_barren.xml με την λήξη κάθε δημοπρασίας. Για την παρουσίαση των αποτελεσμάτων της αναφοράς στο διαχειριστή, υλοποιούνται οι εξής προγραμματισμένες ενέργειες:

1. Στέλνετε ένα αίτημα μέσω Ajax στο διακομιστή με την επιλογή της συγκεκριμένης αναφοράς από το διαχειριστή
2. Μετασχηματίζετε το αρχείο report_barren.xml σύμφωνα με το XSLT μετασχηματισμό του αρχείου barren_style.xsl
3. Τα αποτελέσματα του μετασχηματισμού παρουσιάζονται στο διαχειριστή σε HTML μορφή, εμφανίζοντας τα αποτελέσματα της αναφοράς μέσα σε ένα πινάκα (εικόνα 4-69)

4.6 Αρχεία συστήματος

Για την υλοποίηση του συστήματος για online δημοπρασίες και πωλήσεις υπό την μορφή ενός ιστότοπου, δημιουργήθηκαν δέκα πέντε φάκελοι και εβδομήντα έξι αρχεία. Σύμφωνα με το φάκελο store\ που περιλαμβάνει όλα τα αρχεία για την ανάπτυξη του συστήματος:

- Τα αρχεία με το σώμα(body) της κάθε HTML σελίδας που αποτελεί το ηλεκτρονικό κατάστημα, βρίσκονται στη διαδρομή store\
- Στο φάκελο store\connections βρίσκονται τα XML και XSD αρχεία που υλοποιούν ένα μέρος του συστήματος αποθηκεύσεις XML αρχείων
- Στο φάκελο store\core βρίσκονται όλα τα αρχεία κώδικα PHP, Javascript και Ajax που υλοποιούν κομμάτια ενεργειών όπως είναι η διαχείριση των XML αρχείων, η δημιουργία ενός κεντρικού ρολογιού, η δημιουργία ενός χρονομέτρου για κάθε δημοπρασία, η δυνατότητα αποστολής ασυγχρόνιστων αιτημάτων προς το διακομιστή κ.τ.λ.

- Στο φάκελο **store\css** βρίσκονται τα αρχεία CSS που συμβάλουν στην μορφοποίηση και τον καθορισμό της εμφάνισης κάθε ιστοσελίδας
- Στο φάκελο **store\img** βρίσκονται τα αρχεία εικόνες που χρησιμοποιήθηκαν στις ιστοσελίδες
- Στο φάκελο **store\includes** βρίσκονται τα αρχεία της επικεφαλίδα, του υποσέλιδου, το βασικό μενού, το πλαϊνό μενού και άλλα στοιχεία που αποτελούν μέρος της υλοποίησης του ιστότοπου. Τα αρχεία αυτά συμπεριλαμβάνονται μέσω της PHP επέκτασης include στο σώμα της κάθε ιστοσελίδας. Με την συγκεκριμένη υλοποίηση μειώνουμε την ανάπτυξη διπλότυπου κώδικα
- Στο φάκελο **store\offers** βρίσκεται το αρχείο auction_products.xml και το αρχείο μετασχηματισμού XSLT auction_products.xsl, που συμβάλλουν στην αποθήκευση και στην παρουσίαση των δημοπρασιών που πρόκειται να λάβουν μέρος στο μέλλον
- Στο φάκελο **store\reports** βρίσκεται το αρχείο report_barren.xml και το αρχείο μετασχηματισμού XSLT barren_style.xsl, που συμβάλλουν στην αποθήκευση και στην παρουσίαση των δημοπρασιών που έληξαν άγονες
- Στο φάκελο **store\search** βρίσκονται τα XML, XSD και XSL αρχεία που υλοποιούν ένα μέρος της διαδικασίας αναζήτησης μίας δημοπρασίας
- Στο φάκελο **store\users** βρίσκονται τα XML και XSD αρχεία που υλοποιούν το σύστημα αποθήκευσης χρηστών

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα και μελλοντικές επεκτάσεις του συστήματος

Στην παρούσα πτυχιακή εργασία σχεδιαστική, αναπτύχθηκε και υλοποιήθηκε ένα σύστημα για online δημοπρασίες και πωλήσεις. Στο διαδίκτυο σήμερα υπάρχουν πάρα πολλές εφαρμογές που υλοποιούν ένα τέτοιο σύστημα και η πλειοψηφία αυτών, παρέχει την δυνατότητα σε πολλούς χρήστες να συμμετέχουν σε δημοπρασίες προϊόντων μέσω ενός ιστότοπου που έχει την μορφή ενός ηλεκτρονικού καταστήματος. Ένα από τα πιο σημαντικά κομμάτια στην προσέγγιση υλοποίησης οποιουδήποτε ηλεκτρονικού καταστήματος, είναι η επιλογή του συστήματος αποθήκευσης δεδομένων που θα υποστηρίξει. Στην ουσία μια τέτοια επιλογή αποτελεί μονόδρομο, γιατί σχεδόν πάντα η ανάπτυξη μίας τέτοιας διαδικτυακής εφαρμογής, ακολουθεί το μοντέλο αποθήκευσης σχεσιακών βάσεων δεδομένων. Σε αυτό το κομμάτι η υλοποίηση του συστήματος που αναπτύχθηκε στη παρούσα πτυχιακή εργασία διαφοροποιείται σε σχέση με την πλειοψηφία αυτών των συστημάτων. Για την αποθήκευση όλων των απαραίτητων και ζωτικών δεδομένων που είναι αναγκαία για την λειτουργία του ιστότοπου επιλέχθηκε η τεχνολογία XML.

Είναι πολύ σημαντικό σε αυτό το σημείο να καταλάβουμε ότι η τεχνολογία XML, δεν αποτελεί κάποιο μέρος της τεχνολογίας των βάσεων δεδομένων ούτε είναι η ίδια μια βάση δεδομένων. Οι βάσεις δεδομένων, είναι μια δοκιμασμένη τεχνολογία με περισσότερα από 20 χρόνια εμπειρίας, παρέχοντας ένα γρήγορο, σταθερό και ασφαλή τρόπο αποθήκευσης και ανάκτησης δεδομένων. Στην τελική, η τεχνολογία XML είναι μια γλώσσα ενώ οι βάσεις δεδομένων αποτελούν ένα σύστημα. Τα κοινά σημεία της τεχνολογίας XML και των βάσεων δεδομένων είναι ότι και οι δυο τεχνολογίες μπορούν και αποθηκεύουν δεδομένα όπως και οι δύο τεχνολογίες παρέχουν καθιερωμένες τεχνικές στην ανάκτηση των δεδομένων. Ίσως για αυτό και οι άνθρωποι συγχέουν τις περισσότερες φορές τις τεχνολογίες αυτές. Στην πραγματικότητα όμως αποτελούν κάτι ξεχωριστό και χρησιμοποιούνται για πολύ διαφορετικούς σκοπούς.

Οι σχεσιακές βάσεις δεδομένων χρησιμοποιούνται πολύ καλύτερα για την διαχείριση μεγάλου όγκου πληροφορίας μέσα σε ένα σύστημα, και παρέχουν ώριμα μοντέλα διαχειρίσεις που μπορούν αποτελεσματικά και αξιόπιστα να διατηρούν μεγάλες ποσότητες δομημένων δεδομένων. Τα δεδομένα που διατηρούν, μπορούν να ενημερώνονται πολύ γρήγορα μέσω συναλλαγών, διασφαλίζοντας πάντα την ακεραιότητα της βάσης. Με αυτά τα κριτήρια, δεν υπάρχει άμεσα κάποιο αποτελεσματικό και αξιόπιστο σύστημα διαχείρισης XML εγγράφων που να μπορεί να πληροί τις δυνατότητες αποθήκευσης και διατήρησης μεγάλου όγκου δεδομένων. Αντίστοιχα, η τεχνολογία των XML εγγράφων χρησιμοποιείται πολύ καλύτερα για την ηλεκτρονική μεταφορά δεδομένων κειμένου ανάμεσα σε διαφορετικά συστήματα, όπου μαζί με τα αποθηκευμένα δεδομένα μεταφέρεται και η περιγραφή της δομής του XML εγγράφου. Αυτός είναι ο βασικός και ίσως ο μοναδικός λόγος για τον οποίο η τεχνολογία XML χρησιμοποιείται κατά κόρον για αυτό το σκοπό. Όλη η πληροφορία δεδομένων μαζί με την περιγραφή της δομής της, είναι αποθηκευμένη σε ένα XML έγγραφο που μπορεί πολύ εύκολα να διαβαστεί από διαφορετικά συστήματα, από διαφορετικές τεχνολογίες ακόμα και από τον άνθρωπο. Παρόλο που και οι βάσεις δεδομένων μπορούν να αυτό-περιγράφονται, γενικά στο σύνολο τους δεν μεταδίδονται μεταξύ διαφορετικών συστημάτων.

Ας σκεφτούμε τώρα ένα φυσικό τρόπο συνδυασμού αυτών των δύο τεχνολογιών. Ας υποθέσουμε ότι το σύστημα για online δημοπρασίες και πωλήσεις υποστηρίζει μία βάση δεδομένων για την διαχείριση και τη διατήρηση μεγάλου όγκου δεδομένων. Το σύστημα αυτό, επίσης υποστηρίζει τη χρήση XML εγγράφων για την μετάδοση δεδομένων σε προγράμματα πελάτη όπως και σε άλλες βάσεις δεδομένων για την ταυτοποίηση των προσωπικών στοιχείων των εγγεγραμμένων πελατών του ηλεκτρονικού καταστήματος. Αύτη η σκέψη θα μπορούσε να αποτελέσει μια μελλοντική επέκταση του συστήματος, όπου ο συνδυασμός αυτών των δύο τεχνολογιών αποτελεί έναν από τους πιο βέλτιστους τρόπους ανάπτυξη διαδικτυακών εφαρμογών με γρήγορη, ασφαλή και αξιόπιστη διαχείριση δεδομένων όπως επίσης και με δυνατότητες επέκτασης του συστήματος σε άλλες βάσεις δεδομένων.

Βιβλιογραφία

Βιβλία:

- [1]. Holzner, Steven. *Inside XML*. Indianapolis, Ind.: New Riders, 2001. Print.
- [2]. Holzner, Steven. *XML*. New York: McGraw-Hill, 2009. Print.
- [3]. Fawcett, Joe, Liam Quin, and Danny Ayers. *Beginning XML*. Indianapolis, IN: Wiley, 2012. Print.
- [4]. Tidwell, Doug. *XSLT*. Cambridge [Mass.]: O'Reilly, 2001. Print.
- [5]. Holzner, Steven. *PHP*. New York: McGraw-Hill, 2008. Print.
- [6]. Richards, Robert. *Pro PHP XML And Web Services*. Berkeley, CA: Apress, 2006. Print.

Πηγές από το διαδίκτυο:

Mysql:

- [7]. <http://dev.mysql.com/doc/refman/5.1/en/history.html>
- [8]. https://en.wikipedia.org/wiki/MySQL#cite_note-12
- [9]. <http://buytaert.net/the-history-of-mysql-ab>

PostgreSQL:

- [10]. <http://www.postgresql.org/about/history/>
- [11]. <https://en.wikipedia.org/wiki/PostgreSQL>

SQLite:

- [12]. <https://en.wikipedia.org/wiki/SQLite>

SQL Server:

- [13]. https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [14]. https://el.wikipedia.org/wiki/Microsoft_SQL_Server

File Maker:

- [15]. <https://en.wikipedia.org/wiki/FileMaker>

Oracle database:

- [16]. https://en.wikipedia.org/wiki/Oracle_database

Java:

- [17]. https://en.wikipedia.org/wiki/Java_programming_language

ASP.NET:

- [18]. <https://el.wikipedia.org/wiki/ASP.NET>
- [19]. <https://en.wikipedia.org/wiki/ASP.NET>
- [20]. https://en.wikipedia.org/wiki/Active_Server_Pages
- [21]. <http://www.brighthub.com/internet/web-development/articles/5009.aspx>

JavaServer Pages:

- [22]. https://en.wikipedia.org/wiki/JavaServer_Pages

Perl:

- [23]. <https://en.wikipedia.org/wiki/Perl>
- [24]. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Perl.html>

Ruby:

- [25]. <https://el.wikipedia.org/wiki/Ruby>
- [26]. https://en.wikipedia.org/wiki/Ruby_on_Rails
- [27]. https://en.wikipedia.org/wiki/Ruby_%28programming_language%29

Python:

- [28]. <https://el.wikipedia.org/wiki/Python>
- [29]. https://en.wikipedia.org/wiki/Python_%28programming_language%29

XML, XML Schema, DTD, XSLT, X-Path:

- [30]. <https://en.wikipedia.org/wiki/XML>
- [31]. <https://el.wikipedia.org/wiki/XML>
- [32]. http://www.fme.aegean.gr/sites/default/files/dsampsom_xml_lectures-notes-dec2003.pdf
- [33]. <http://www.rpbouret.com/xml/>
- [34]. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-XML.html>
- [35]. <http://www.w3.org/TR/REC-xml/>
- [36]. <http://www.w3schools.com/xml/>
- [37]. https://en.wikipedia.org/wiki/Markup_language
- [38]. <https://msdn.microsoft.com/en-us/library/ms256048%28v=vs.110%29.aspx>
- [39]. http://www.w3schools.com/schema/el_documentation.asp
- [40]. http://www.w3schools.com/schema/el_appinfo.asp
- [41]. http://www.w3schools.com/schema/el_annotation.asp
- [42]. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-XSL.html>
- [43]. <https://en.wikipedia.org/wiki/XSL>
- [44]. <http://www.w3schools.com/xsl/default.asp>
- [45]. <https://en.wikipedia.org/wiki/XPath>
- [46]. https://en.wikipedia.org/wiki/Query_language

PHP:

- [47]. <http://php.net/>
- [48]. <https://en.wikipedia.org/wiki/PHP>
- [49]. <https://el.wikipedia.org/wiki/PHP>
- [50]. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>

JavaScript:

- [51]. <https://en.wikipedia.org/wiki/JavaScript>
- [52]. <https://el.wikipedia.org/wiki/JavaScript>

Ajax:

- [53]. https://en.wikipedia.org/wiki/Ajax_%28programming%29

HTML:

- [54]. <https://en.wikipedia.org/wiki/HTML>
- [55]. <https://el.wikipedia.org/wiki/HTML>
- [56]. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-HTML.html>

[57]. <http://www.w3.org/MarkUp/>

CSS:

[58]. https://en.wikipedia.org/wiki/Cascading_Style_Sheets

[59]. <https://el.wikipedia.org/wiki/CSS>

WampServer:

[60]. <http://techterms.com/definition/wamp>

[61]. <http://techterms.com/definition/wamp>

Adobe Dreamweaver:

[62]. https://en.wikipedia.org/wiki/Adobe_Dreamweaver

Τεχνολογικές έννοιες:

[63]. https://en.wikipedia.org/wiki/First-class_function

[64]. <https://en.wikipedia.org/wiki/MIME>

[65]. https://en.wikipedia.org/wiki/Zend_Engine

[66]. https://el.wikipedia.org/wiki/Common_Gateway_Interface

[67]. <http://php.net/manual/en/class.arrayiterator.php>

[68]. <https://en.wikipedia.org/wiki/Iterator>

[69]. https://en.wikipedia.org/wiki/Name_binding

[70]. https://en.wikipedia.org/wiki/Garbage_collection_%28computer_science%29

[71]. <https://en.wikipedia.org/wiki/Goto>

[72]. https://en.wikipedia.org/wiki/Programming_language#Type_system

[73]. <https://en.wikipedia.org/wiki/ENQUIRE>

[74]. <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

[75]. <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

[76]. https://en.wikipedia.org/wiki/Site-specific_browser

[77]. https://en.wikipedia.org/wiki/Asynchronous_I/O

[78]. https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

[79]. <https://en.wikipedia.org/wiki/Braille>

[80]. <https://en.wikipedia.org/wiki/XMLHttpRequest>

[81]. https://en.wikipedia.org/wiki/Closure_%28computer_programming%29

[82]. https://en.wikipedia.org/wiki/Document_Object_Model

ΠΑΡΑΡΤΗΜΑ ΠΤΥΧΙΑΚΗΣ
ΣΥΣΤΗΜΑ ΓΙΑ ONLINE ΔΗΜΟΠΡΑΣΙΕΣ ΚΑΙ ΠΩΛΗΣΕΙΣ

ΠΕΡΙΕΧΟΜΕΝΟ

**ΠΑΡΑΡΤΗΜΑ Α : ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΚΑΙ
ΑΡΧΕΙΩΝ ΣΥΣΤΗΜΑΤΟΣ**

ΠΑΡΑΡΤΗΜΑ Β : ΔΙΑΦΑΝΙΕΣ ΠΑΡΟΥΣΙΑΣΗΣ

ΠΑΡΑΡΤΗΜΑ Α

ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΚΑΙ ΑΡΧΕΙΩΝ ΣΥΣΤΗΜΑΤΟΣ

1.1 Εγκατάσταση WampServer 2.2

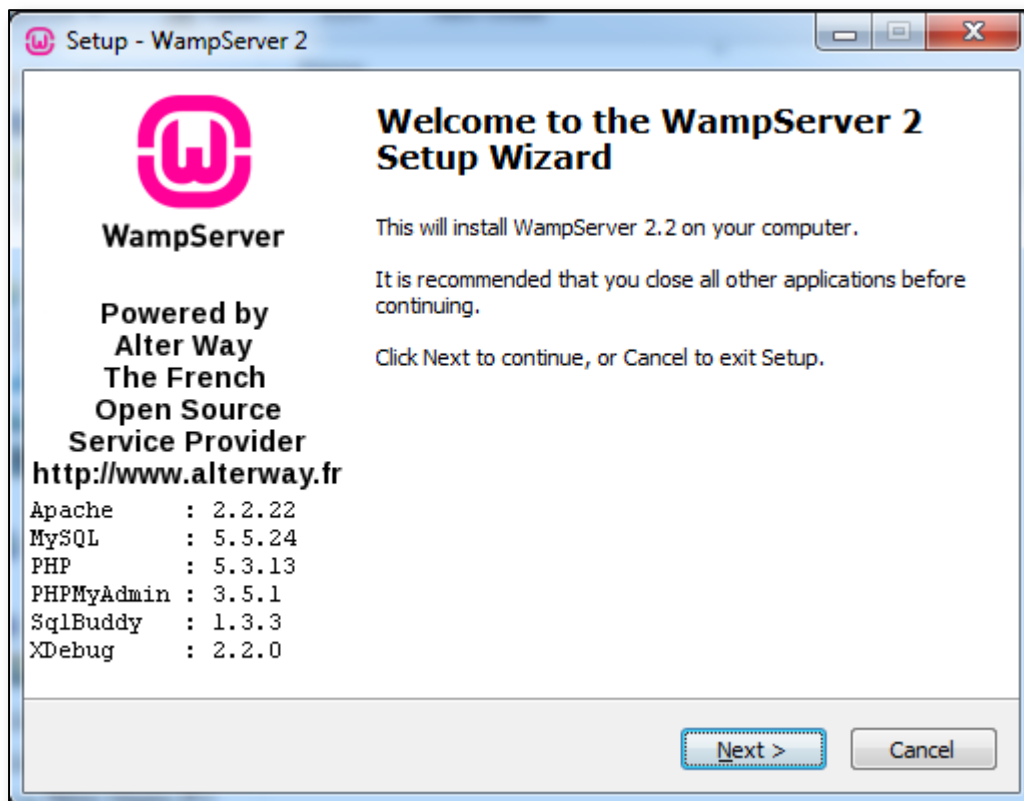
Για να βάλουμε τον WampServer 2.2 στον υπολογιστή μας, κατεβάζουμε την εγκατάσταση του προγράμματος από την ιστοσελίδα <http://www.wampserver.com/en/#download-wrapper>. Στην περίπτωση που έχει κυκλοφορήσει κάποια νέα έκδοση, μπορούμε να βρούμε την παλαιότερη έκδοση 2.2 στις εξής σελίδες:

- 32 bit
<http://sourceforge.net/projects/wampserver/files/WampServer%20/WampServer%202.2/wampserver2.2e/wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-32b.exe/download>
- 64 bit
<http://sourceforge.net/projects/wampserver/files/WampServer%20/WampServer%202.2/wampserver2.2e/wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-x64.exe/download>

Εγκατάσταση WampServer 2.2 σε Windows λειτουργικά:

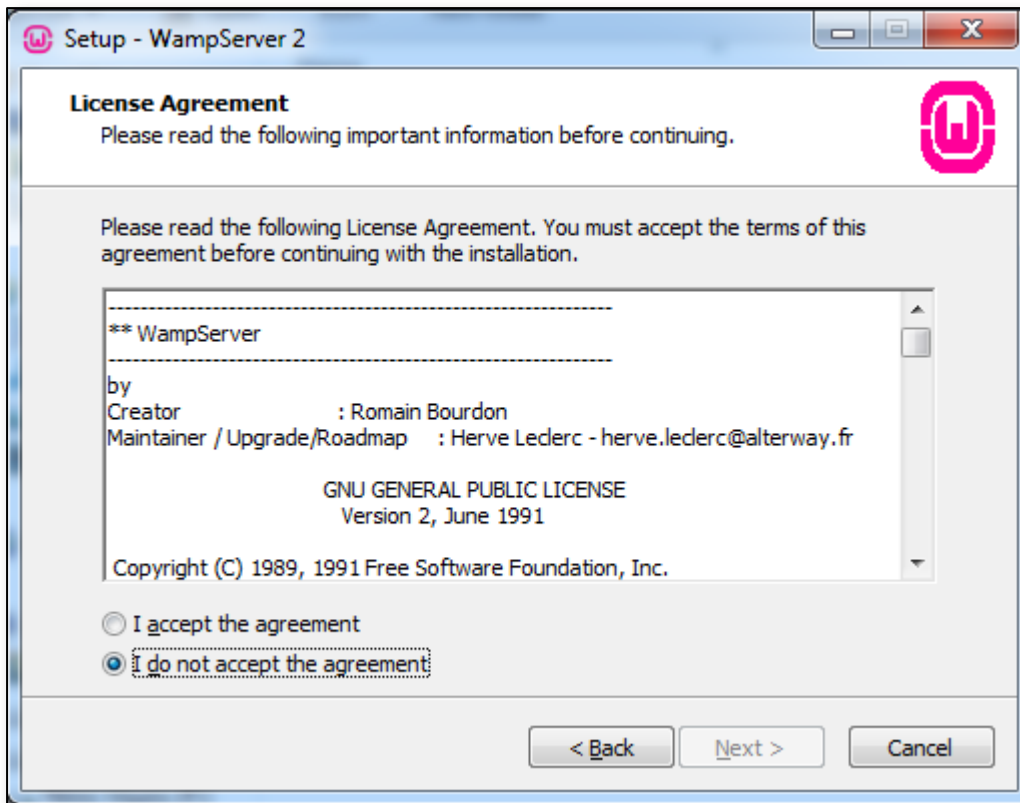
1^ο Βήμα:

Πατάμε διπλό κλικ στο εκτελέσιμο αρχείο του WampServer και στη συνέχεια πατάμε next στο παράθυρο διαλόγου που μας ανοίγει.



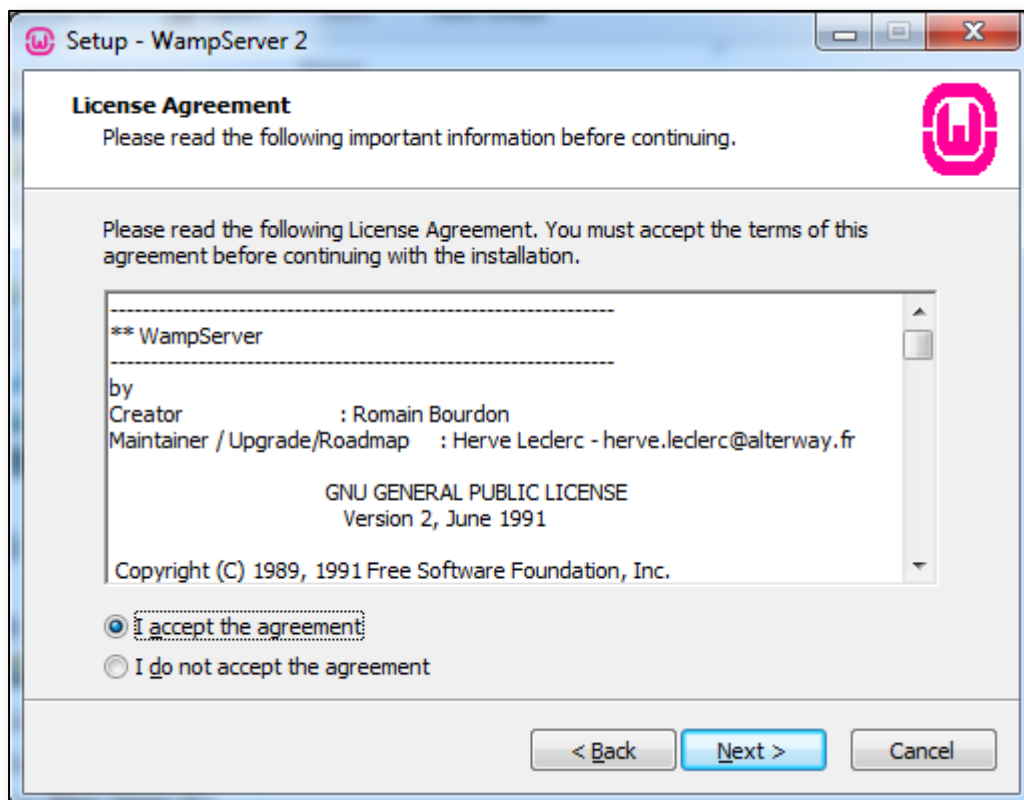
2° Βήμα:

Επιλέγουμε στο πεδίο επιλογής I accept the agreement.



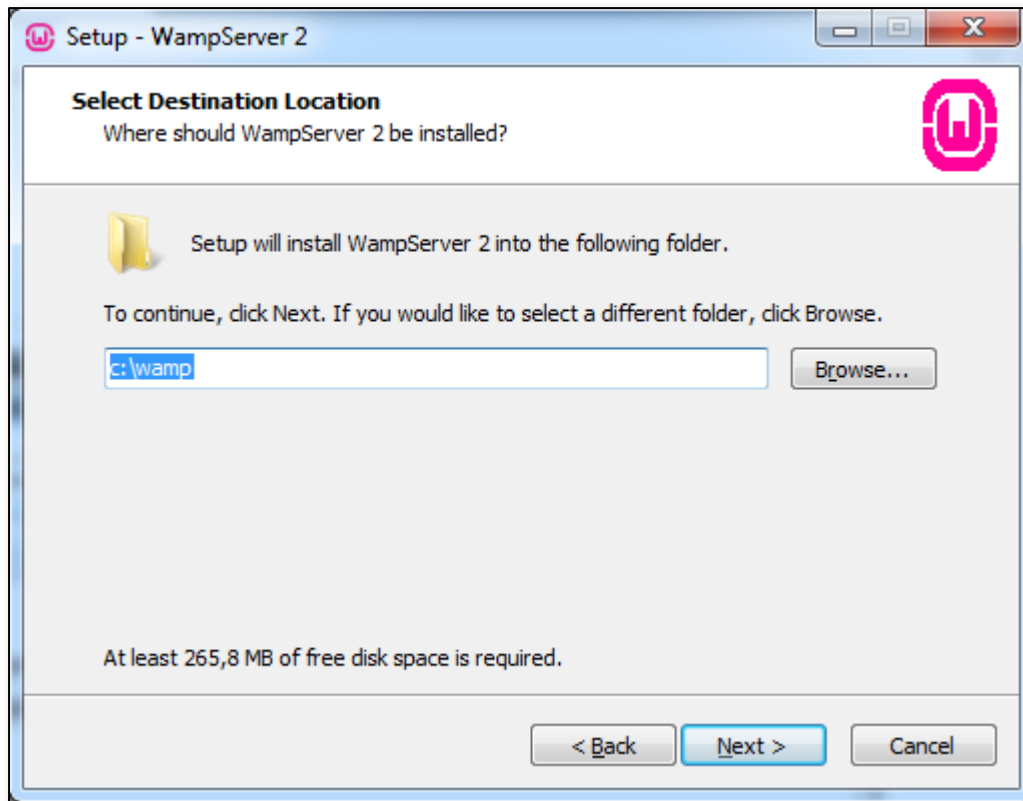
3° Βήμα:

Πατάμε next στο παράθυρο διαλόγου.



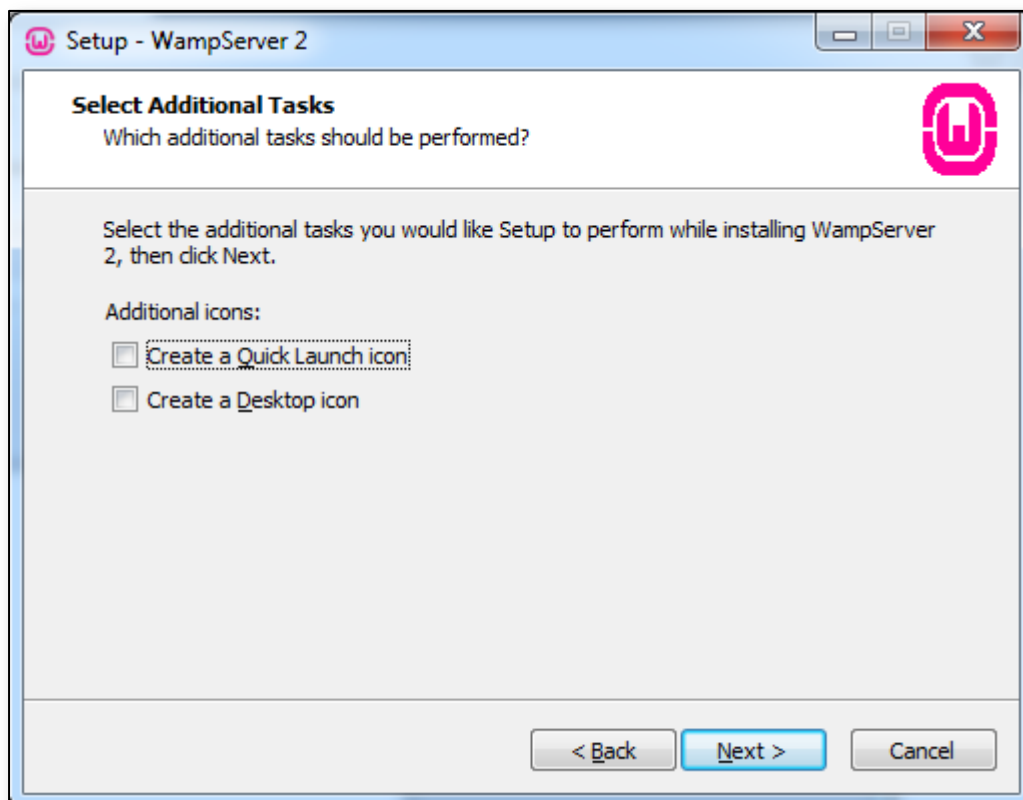
4^ο Βήμα:

Πατάμε next στο παράθυρο διαλόγου. Δεν αλλάζουμε την διαδρομή αποθηκεύσεις του προγράμματος.

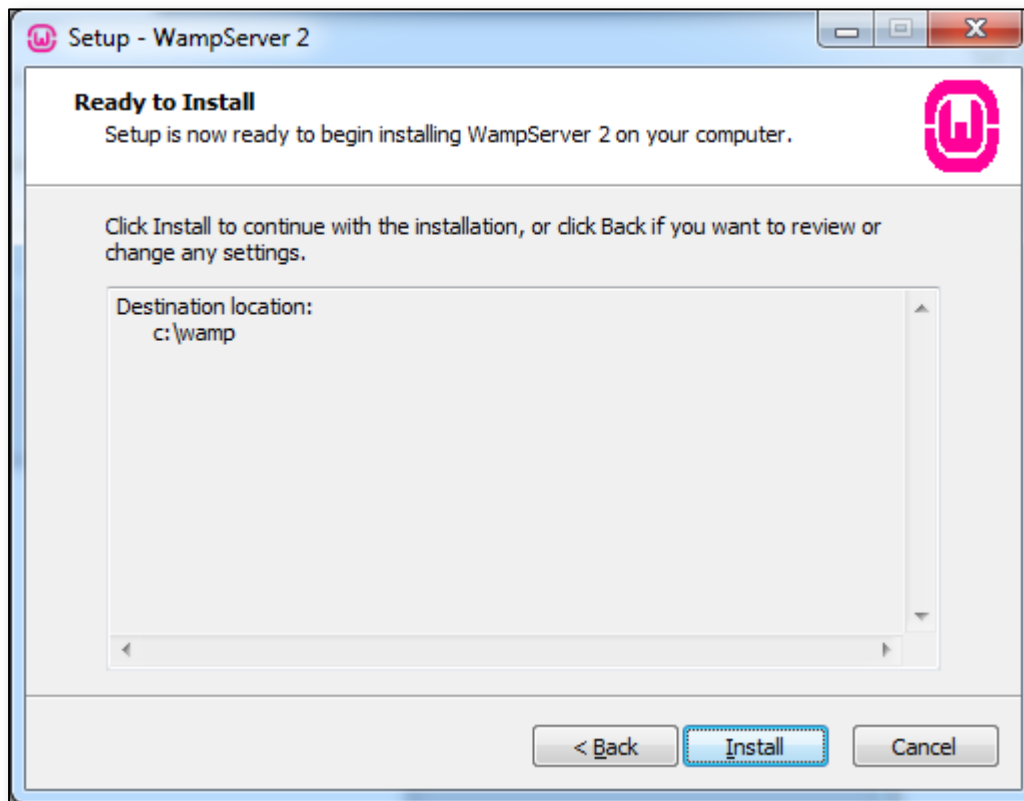


5^ο Βήμα:

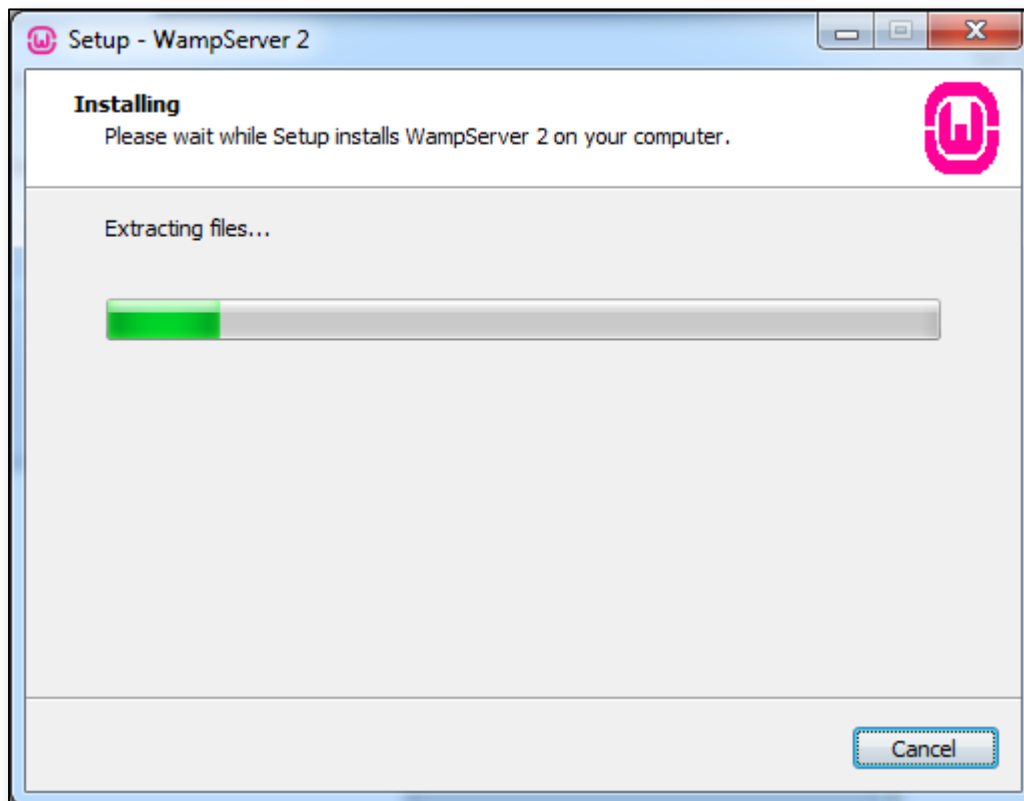
Πατάμε next στο παράθυρο διαλόγου.



6^ο Βήμα:
Πατάμε install στο παράθυρο διαλόγου.



7^ο Βήμα:
Περιμένουμε να τελειώσει η εγκατάσταση.

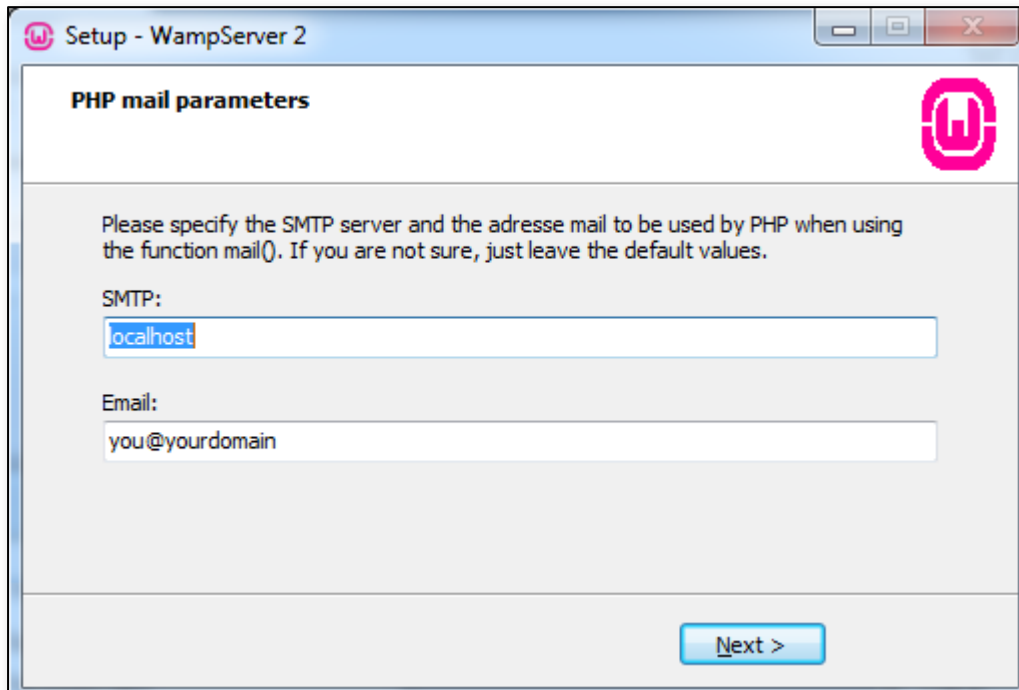


8° Βήμα:

Όταν μας ζητηθεί το προεπιλεγμένο πρόγραμμα περιήγησης, επιλέγουμε όποιο πρόγραμμα θέλουμε από τον υπολογιστή μας.

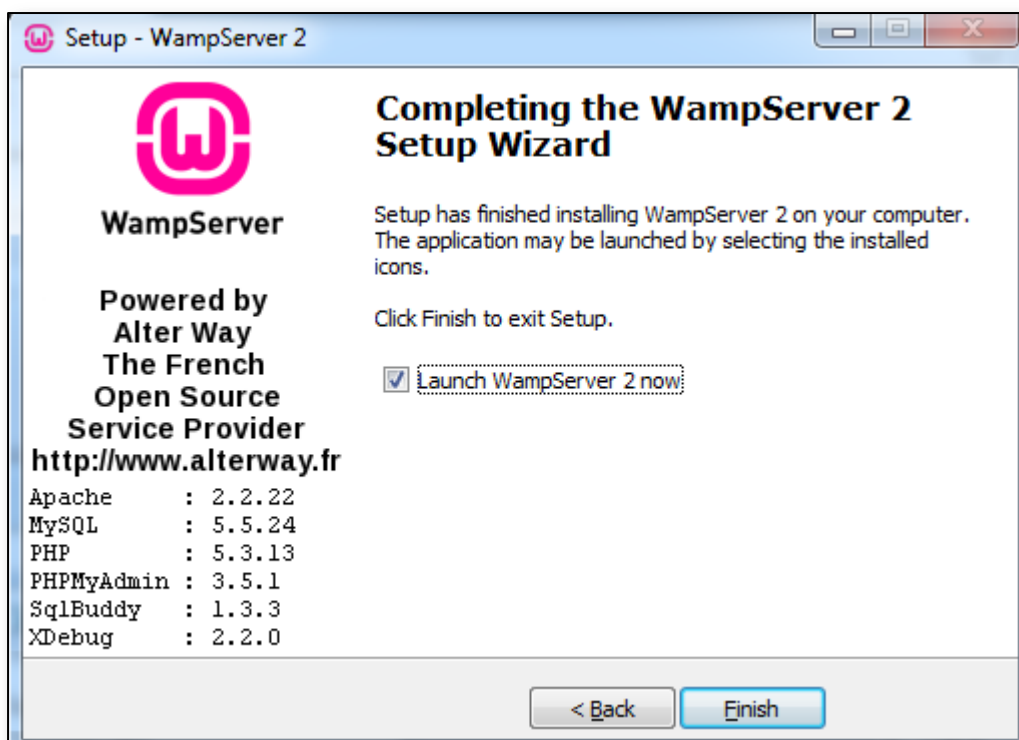
9° Βήμα:

Πατάμε next στο παράθυρο διαλόγου.



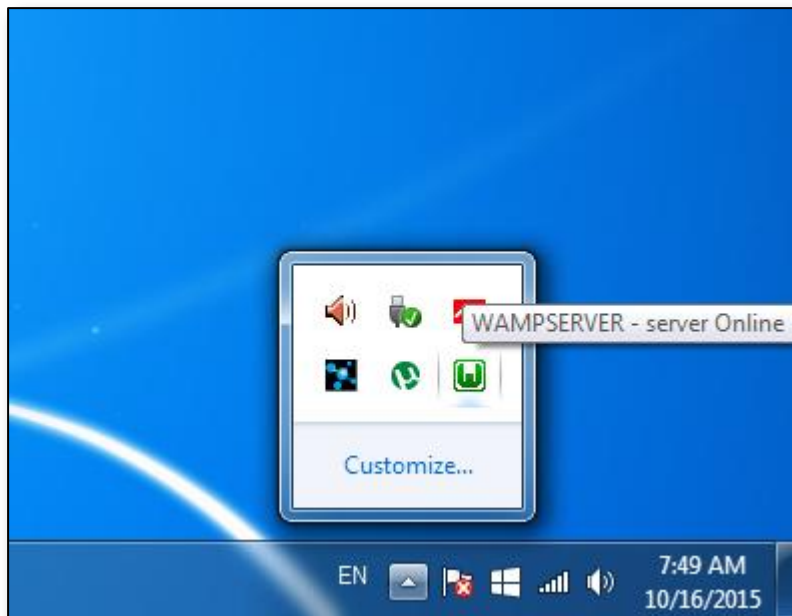
9° Βήμα:

Πατάμε finish στο παράθυρο διαλόγου και περιμένουμε να τρέξει ο Wamp Server στον υπολογιστή μας.



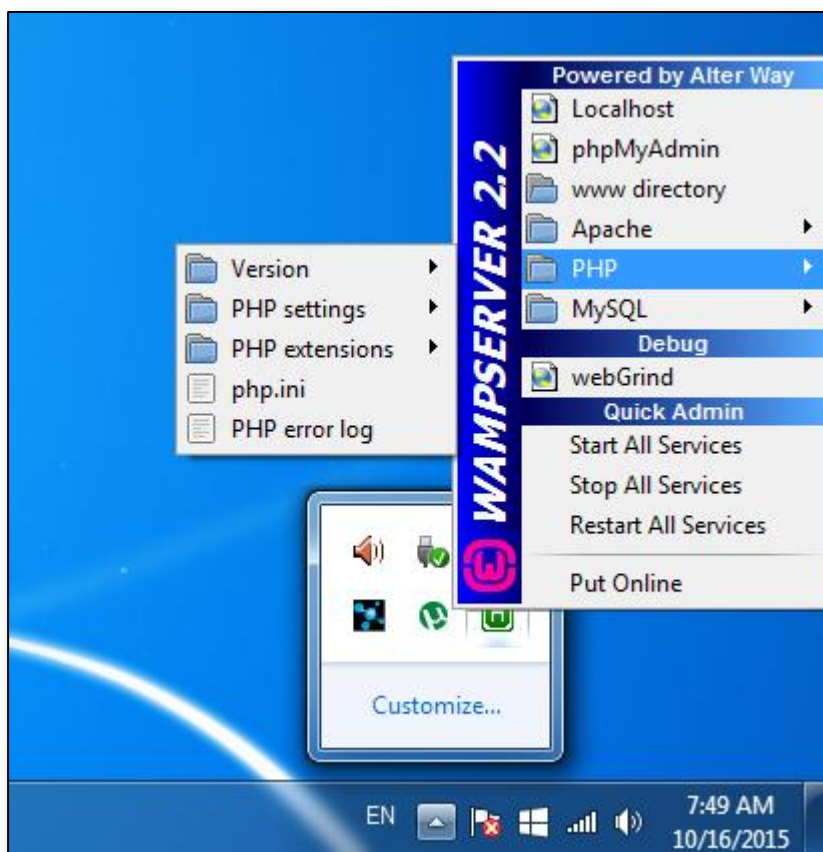
10^ο Βήμα:

Εφόσον ανοίξει ο Wamp Server, εμφανίζεται κάτω δεξιά στην μπάρα εργασιών των Windows το αντίστοιχο εικονίδιο του προγράμματος.



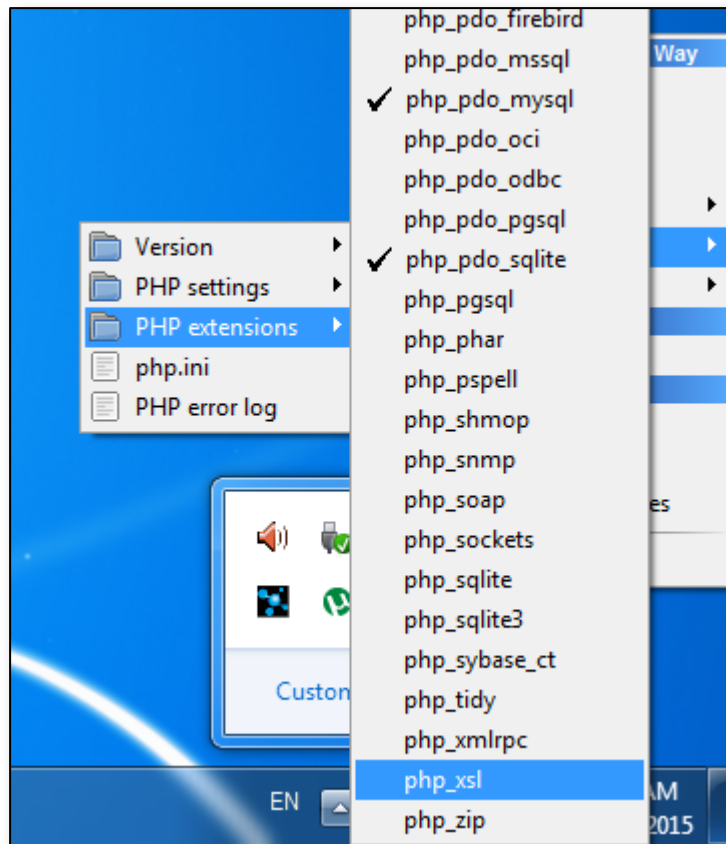
11^ο Βήμα:

Πατάμε αριστερό κλικ πάνω στο εικονίδιο του Wamp Server και πάμε με τον κέρσορα του ποντικιού πάνω στην επιλογή PHP.



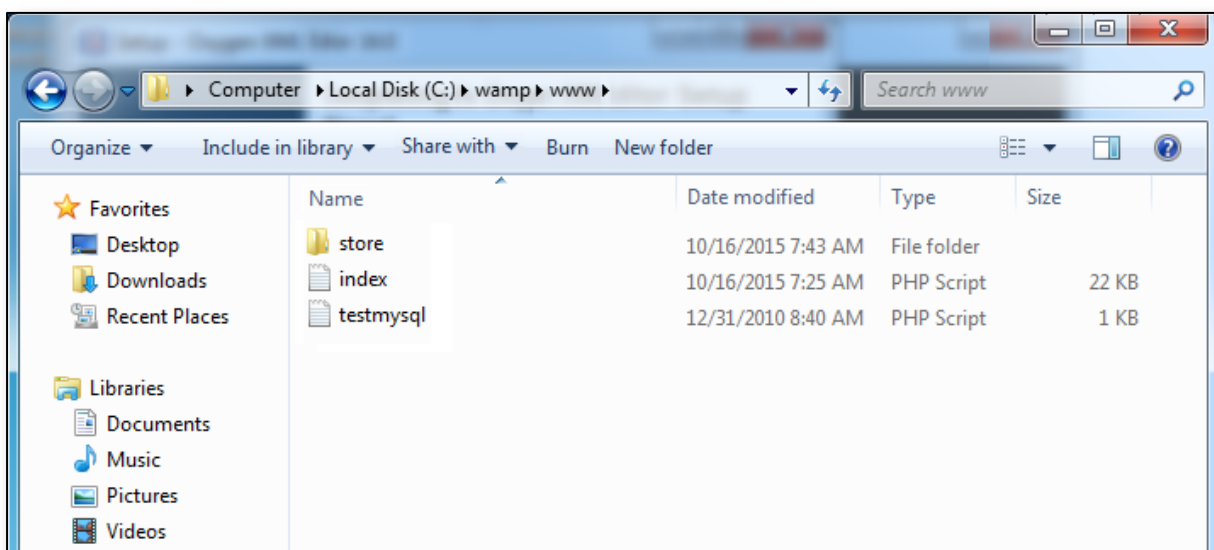
12^ο Βήμα:

Πάμε με τον κέρσορα του ποντικιού μας πάνω στην επιλογή PHP extensions, κατεβαίνουμε προς τα κάτω την λίστα και ενεργοποιούμε τις επιλογές php_xmlrpc και php_xsl.



1.2 Εγκατάσταση των αρχείων συστήματος στο διακομιστή

Για να μπορέσουμε να εμφανίσουμε τις σελίδες του ηλεκτρονικού καταστήματος για online δημοπρασίες και πωλήσεις σε οποιοδήποτε πρόγραμμα περιήγησης στον υπολογιστή μας πρέπει πρώτα να βάλουμε τα αρχεία συστήματος στο διακομιστή. Εφόσον η εγκατάσταση του Wamp Server έγινε επιτυχώς, μπορούμε να πάμε και να αντιγράψουμε το φάκελο store(φάκελος που περιέχει τα αρχεία του συστήματος) στην διαδρομή C:\wamp\www του σκληρού δίσκου C.



Μετά την αντιγραφή των αρχείων του συστήματος στο σκληρό δίσκο C του υπολογιστή μας, μπορούμε να πληκτρολογήσουμε σε οποιοδήποτε πρόγραμμα περιήγησης(προτείνεται ο Mozilla Firefox) την διεύθυνση <http://localhost/store/index.php> για να συνδεθούμε στην σελίδα index του ηλεκτρονικού καταστήματος.

ΠΑΡΑΡΤΗΜΑ Β

1.1 Διαφάνειες Παρουσίασης



**Σύστημα για online δημοπρασίες
& πωλήσεις**

Λεβετσοβίτης Πέτρος Α.Μ. 2103

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος
Επιτροπή αξιολόγησης : Βαλσαμάκης Ιωάννης
Τσαγκαράκης Χαράλαμπος

Ανάπτυξη συστήματος για online δημοπρασίες & πωλήσεις

Το σύστημα για online δημοπρασίες και πωλήσεις έχει την μορφή ενός ηλεκτρονικού καταστήματος, υποστηρίζοντας τις εξής διαδικασίες και δυνατότητες:

- Απλούς χρήστες (users)
- Έναν χρήστη-διαχειριστή (administrator)
- Πώληση προϊόντων μέσω δημοπρασίας που εκτελείτε σε πραγματικό χρόνο
- Αναζήτηση δημοπρασιών
- Δημιουργία συγκεντρωτικών αναφορών (page reports) για τα αποτελέσματα των δημοπρασιών που έλαβαν τέλος

Υλοποίηση συστήματος για online δημοπρασίες & πωλήσεις

Η υλοποίηση του ηλεκτρονικού καταστήματος χωρίζεται σε **2 μέρη**:

1. Σύστημα αποθήκευσης δεδομένων:

Για την αποθήκευση της χρήσιμης πληροφορίας που συμβάλει στην ορθή διεξαγωγή και λειτουργία του συστήματος, γίνεται χρήση της τεχνολογίας XML.

Ο συνδυασμός της γλώσσας XML με τις τεχνολογίες XML Schema και XSL που υποστηρίζει, προσφέρει ένα αξιόπιστο περιβάλλον αποθήκευσης, ανάκτησης και παρουσίασης κάθε πληροφορίας με άρτια δομημένο τρόπο.

Υλοποίηση συστήματος για online δημοπρασίες & πωλήσεις

2. Περιβάλλον και δυνατότητες συστήματος:

Για την δημιουργία και το σχεδιασμό του ηλεκτρονικού καταστήματος χρησιμοποιήθηκαν συνδυαστικά οι γλώσσες PHP, HTML, JavaScript, CSS και η τεχνική Ajax.

Πιο αναλυτικά:

- Η γλώσσα **PHP** χρησιμοποιήθηκε για την δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο από την πλευρά του διακομιστή. Αποτελεί επίσης, το βασικό μηχανισμό για τη διαχείριση και την επεξεργασία των XML εγγράφων.
- Η γλώσσα **JavaScript** χρησιμοποιήθηκε για τη παραγωγή δυναμικού περιεχομένου και την προσθήκη διαδραστικότητας στις ιστοσελίδες από την πλευρά του πελάτη.

Υλοποίηση συστήματος για online δημοπρασίες & πωλήσεις

- Η τεχνική **AJAX** χρησιμοποιήθηκε για την ασυγχρόνιστη επικοινωνία του συστήματος με τον διακομιστή, επιτρέποντας την αποστολή και την ανάκτηση δεδομένων στο προσκήνιο, χωρίς να είναι απαραίτητη η ανανέωση ολόκληρης της ιστοσελίδας.
- Η γλώσσα **HTML** χρησιμοποιήθηκε για την δημιουργία των ιστοσελίδων, των φορμών εισαγωγής δεδομένων, και των πινάκων παρουσίασης δεδομένων.
- Η γλώσσα φύλλων στυλ **CSS** χρησιμοποιήθηκε για την μορφοποίηση και τον καθορισμό της εμφάνισης κάθε ιστοσελίδας.

Δυνατότητες χρήστη

Το ηλεκτρονικό κατάστημα για online δημοπρασίες και πωλήσεις υποστηρίζει τις εξής δυνατότητες για ένα χρήστη:

- ✓ Εγγραφή χρήστη
- ✓ Συμμετοχή σε μια πώληση προϊόντος μέσω δημοπρασίας σε πραγματικό χρόνο
- ✓ Δυνατότητα αναζήτησης δημοπρασιών με βάση το όνομα του προϊόντος, την κατηγορία του προϊόντος και την εταιρία στην οποία ανήκει το προϊόν

Φόρμα εγγραφής χρήστη

The screenshot shows the 'Super Auctions' website interface. At the top, there is a navigation menu with 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock displays '10:12:00pm' and the date 'October 06, 2015'. The main content area is divided into two sections: 'Registration' and 'Login/Register'. The 'Registration' section contains input fields for 'Username*', 'Password*', 'Password verification*', 'Name*', 'Address', 'Phone', and 'Credit card number*', along with a 'Register' button. The 'Login/Register' section contains input fields for 'username:' and 'password:', a 'Log in' button, and a 'Register' link. Below this, a 'Users' section displays the text 'We have 5 registerd user.'

Φόρμα εγγραφής χρήστη

Το κάθε πεδίο της φόρμας αντιστοιχεί σε ένα συγκεκριμένο στοιχείο του αρχείου XML, που είναι υπεύθυνο για την αποθήκευση των προσωπικών δεδομένων ενός χρήστη.

Τα δεδομένα περνούν από δύο ελέγχους πριν αποθηκευτούν στο αντίστοιχο αρχείο XML:

1. PHP έλεγχος:

Η πρώτη διαδικασία αφορά τον PHP έλεγχο των δεδομένων. Όλα τα πεδία με αστερίσκο πρέπει να είναι συμπληρωμένα, το username πρέπει να είναι μοναδικό κ.τ.λ.

2. XML Schema validation:

Εφόσον τα δεδομένα περάσουν επιτυχώς από τον PHP έλεγχο, αποθηκεύονται σε ένα προσωρινό XML αρχείο για να γίνει η διαδικασία εγκυρότητας με το αντίστοιχο XML Schema. Εάν τα δεδομένα που βρίσκονται στο προσωρινό XML αρχείο είναι έγκυρα τότε αποθηκεύονται στο XML αρχείο χρηστών.

Επιλογή δημοπρασίας

Για να μπορέσει ένας εγγεγραμμένος χρήστης να συμμετάσχει σε μία δημοπρασία, θα πρέπει πρώτα να επιλέξει από τον πίνακα δημοπρασιών την αντίστοιχη δημοπρασία που τον ενδιαφέρει:

The screenshot shows the 'Super Auctions' website interface. At the top, there is a navigation bar with 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock displays '8:54:40pm' and the date 'October 07, 2015'. Below the navigation bar, the main content area is titled 'Auctions' and contains a table with 10 rows of auction listings. To the right of the table is a sidebar with a user profile for 'Takis!', including a 'Logout' button and a message stating 'We have 5 registered user.'.

A/A	Product Name	Price(€)	Start Time	Corpration	Category
1	A1	3063	T17:23 27-09-2015	Audi	Cars
2	L200	10700	T15:01 20-10-2015	Mitsubishi	Cars
3	L400	15700	T20:00 01-02-2013	Mitsubishi	Cars
4	Compaq 6530g	345	T18:43 23-07-2015	Hewlett Packard	Computers
5	230g	340	T23:34 26-09-2015	Dell	Computers
6	N300g	700	T21:15 13-11-2015	Dell	Computers
7	N400g	498.50	T20:43 10-10-2015	Dell	Computers
8	G-Shock XL	130.60	T20:24 27-12-2015	Casio	Watches
9	New Melenioum	100	T20:00 18-10-2015	Casio	Watches
10	Vogue City	350.60	T16:00 19-09-2015	Casio	Watches

Επιλογή δημοπρασίας

Ο πίνακας δημοπρασιών δημιουργείται από τον XSLT μετασχηματισμό του εγγράφου auction_products.xml σε HTML μορφή. Η διαδικασία του XSLT μετασχηματισμού παρέχει τις εξής δυνατότητες στην έξοδο του νέου εγγράφου:

- ▶ Παρουσίαση των στοιχείων του XML εγγράφου σε HTML μορφή
- ▶ Δημιουργία HTML πίνακα για την παρουσίαση του περιεχομένου των στοιχείων που συνθέτουν το έγγραφο auction_products.xml
- ▶ Επιλογή των στοιχείων που θα εμφανίζονται στον πίνακα
- ▶ Ταξινόμηση των στοιχείων που εμφανίζονται στον πίνακα
- ▶ Δημιουργία κατάλληλων συνδέσμων (link) για την μετάβαση πληροφοριών, μέσω της URL διεύθυνσης, στην ιστοσελίδα διεξαγωγής μίας δημοπρασίας
- ▶ Γραφική εμφάνιση του περιεχομένου των στοιχείων, μέσω CSS μορφοποίησης

Συμμετοχή χρήστη σε μία δημοπρασία

The screenshot displays the 'Super Auctions' website interface. At the top, there is a navigation bar with links for 'Home', 'Auctions', 'Search', and 'Contact us'. A digital clock shows '12:00:03am' on 'October 11, 2015'. The main content area features an 'Auction for G-Shock XL'. It indicates the auction has started with a countdown timer showing '0 M. 0 D. 0 H. 9 M. 56 S'. The start price is listed as '131 euros' and the corporation is 'Casio'. A description box contains the text: 'Product description. We can load here the product decription for current auction'. Below this is a text input field for 'Give your offer please:' and a 'Submit Offer' button. On the right side, a user profile box shows 'Hello, angell', a 'Logout' link, and a 'Users' section stating 'We have 5 registerd user.'

Συμμετοχή χρήστη σε μία δημοπρασία

Η ιστοσελίδα αποτελείται από τέσσερα βασικά πεδία:

1. Το πεδίο που εμφανίζει το χρονοδιακόπτη μίας δημοπρασίας. Ο χρονοδιακόπτης εκτελεί τρεις λειτουργίες χρονομέτρου:
 - *Εμφάνιση υπολειπόμενου χρόνου έναρξης μίας δημοπρασίας*
 - *Χρονόμετρο με αντίστροφη μέτρηση διάρκειας δέκα λεπτών στην εκκίνηση μίας δημοπρασίας*
 - *Χρονόμετρο διάρκειας πέντε λεπτών για κάθε παράταση μίας δημοπρασίας*
2. Το πεδίο που εμφανίζει την τιμή του προϊόντος σε μία δημοπρασία. Μετά την έναρξη μίας δημοπρασίας η τιμή του προϊόντος ανανεώνεται κάθε 20 δευτερόλεπτα.
3. Το πεδίο εισαγωγής μίας προσφοράς
4. Το κουμπί της φόρμας για την υποβολή μίας προσφοράς

Συμμετοχή χρήστη σε μία δημοπρασία

Συνοπτικά για μία δημοπρασία:

- Ένας χρήστης έχει την δυνατότητα να κάνει μία προσφορά πριν και μετά την έναρξη της δημοπρασίας.
- Προσφορές πριν την έναρξη της δημοπρασίας δεν φαίνονται στους χρήστες.
- Η τιμή του προϊόντος ανανεώνεται αυτόματα με την τιμή της μεγαλύτερης προσφοράς χρήστη, με την έναρξη μιας δημοπρασίας.
- Μια δημοπρασία λήγει άγονη εφόσον δεν έχει γίνει καμία προσφορά με μεγαλύτερη τιμή από την αρχική, για το χρονικό διάστημα πριν και μετά την έναρξή της δημοπρασίας, για τουλάχιστον δέκα λεπτά.
- Ένας χρήστης κερδίζει σε μία δημοπρασία, εφόσον δεν έχει γίνει μεγαλύτερη πρόσφορα από αυτή που έχει κάνει ο ίδιος, για τουλάχιστον δέκα λεπτά μετά την έναρξη μίας δημοπρασίας ή για τουλάχιστον πέντε λεπτά που διαρκεί μία παράταση δημοπρασίας.

Αναζήτηση δημοπρασιών

Ένας χρήστης έχει την δυνατότητα να διεξάγει αναζήτηση δημοπρασιών βάσει τριών κριτηρίων:

- Το όνομα του προϊόντος
- Την κατηγορία προϊόντος
- Την εταιρία προϊόντος

The screenshot shows the 'Super Auctions' website interface. At the top, there is a navigation bar with 'Home', 'Auctions', 'Search', and 'Contact us' links. A digital clock displays '10:06:55pm' and the date 'October 11, 2015'. Below the navigation bar is a 'Search Auctions' section. It features a dropdown menu for 'Category' with options: '--- Search Auctions by ---', 'Product', 'Corporation', and 'Category' (which is selected). Below the dropdown is a text input field labeled 'Type Category' containing the word 'Computers', and a 'search' button. To the right of the search section is a user profile box for 'Hello, angell!' with a 'Logout' link and a 'Users' section stating 'We have 5 registerd user.'

Δυνατότητες διαχειριστή συστήματος

Το ηλεκτρονικό κατάστημα για online δημοπρασίες και πωλήσεις υποστηρίζει ένα χρήστη - διαχειριστή με τις εξής δυνατότητες:

- Εισαγωγή νέων προϊόντων
- Εισαγωγή νέων δημοπρασιών
- Επίβλεψη συγκεντρωτικών αναφορών (page reports)

Η διαφοροποίηση ενός απλού χρήστη από έναν χρήστη με δικαιώματα διαχειριστή, μέσα στο XML αρχείο που είναι υπεύθυνο για την αποθήκευση των προσωπικών δεδομένων ενός χρήστη, υλοποιείται μέσω του στοιχείου type.

Το στοιχείο μπορεί να πάρει μόνο δυο τιμές για τον προσδιορισμό των δικαιωμάτων ενός χρήστη και ενός διαχειριστή στο σύστημα.

Εισαγωγή νέων προϊόντων

Ο διαχειριστής του συστήματος έχει την δυνατότητα να εισάγει νέα προϊόντα στο σύστημα. Τα δεδομένα που περιγράφουν το νέο προϊόν αποθηκεύονται στο αντίστοιχο XML αρχείο.

The screenshot shows the 'Super Auctions' web application interface. At the top, there is a navigation bar with 'Home', 'Auctions', 'Search', and 'Contact us'. A clock displays '6:12:45pm' and the date 'October 12, 2015'. Below the navigation bar, there are two main sections. On the left, under 'Administrator Privileges', is the 'Insert Product' form. It includes input fields for 'Product Name', 'Corporation', and 'Price', and a dropdown menu for 'Category' with the text '--- Select Category ---'. An 'Insert Product' button is at the bottom of the form. On the right, there is a user greeting 'Hello, petros!' with a 'Logout' link, and a 'Users' section stating 'We have 4 registerd users'.

Εισαγωγή νέων δημοπρασιών

Ο διαχειριστής έχει την δυνατότητα να εισάγει μία νέα δημοπρασία στο σύστημα, εφόσον υπάρχει κάποιο προϊόν που δεν αντιστοιχεί σε μία δημοπρασία. Το πεδίο Choose product της φόρμας παρουσιάζει τα προϊόντα τα οποία δεν αντιστοιχούν σε μία δημοπρασία.

The screenshot shows the 'Super Auctions' web application interface. At the top, there is a navigation bar with 'Home', 'Auctions', 'Search', and 'Contact us'. A clock displays '7:59:55pm' and the date 'October 12, 2015'. Below the navigation bar, there are two main sections. On the left, under 'Administrator Privileges', is the 'Insert Auctions' form. It includes input fields for 'Day', 'Month', and 'Year' with an example 'ex : 29/08/2015', a 'Time start' field with an example 'ex : 19:30', and a 'Choose product' dropdown menu. The dropdown menu is open, showing '--- Select product ---' and 'Acer Aspire E-71'. An 'Insert Auction' button is at the bottom of the form. On the right, there is a user greeting 'Hello, petros!' with a 'Logout' link, and a 'Users' section stating 'We have 4 registerd users'.

Εισαγωγή νέων προϊόντων & δημοπρασιών

Με την υποβολή της φόρμας εισαγωγής ενός νέου προϊόντος ή μίας νέας δημοπρασίας στο σύστημα, ακολουθούν οι εξής προγραμματισμένες ενέργειες από την πλευρά του διακομιστή:

1. PHP έλεγχος δεδομένων
2. Τα δεδομένα αποθηκεύονται σε ένα προσωρινό αρχείο XML, εφόσον περάσουν με επιτυχία τον PHP έλεγχο.
3. Γίνεται έλεγχος εγκυρότητας του προσωρινού αρχείου XML, σύμφωνα με τους κανόνες που ορίζει το αντίστοιχο αρχείο XML Schema.
4. Τα δεδομένα αποθηκεύονται στο αντίστοιχο XML αρχείο προϊόντων/δημοπρασιών, εφόσον το προσωρινό αρχείο κριθεί έγκυρο σύμφωνα με το XML Schema που το περιγράφει.
5. Ο διαχειριστής οδηγείται στην αντίστοιχη ιστοσελίδα μετά την επιτυχή εισαγωγή του προϊόντος ή της δημοπρασίας.

Συγκεντρωτικές αναφορές

Ο διαχειριστής του συστήματος έχει την δυνατότητα επίβλεψης των συγκεντρωτικών αναφορών, που εκδίδει το σύστημα, για την περιγραφή των δημοπρασιών που έχουν λήξει.

Το σύστημα υλοποιεί και παρουσιάζει τις κάτωθι συγκεντρωτικές αναφορές:

1. Πόσες δημοπρασίες έληξαν άγονες,
2. Πόσο είναι ο μέσος χρόνος όλων των δημοπρασιών που έχουν λήξει,
3. Ποια προϊόντα δεν πουλήθηκαν λόγω άγονων δημοπρασιών, ποια ήταν η μεγαλύτερη προσφορά για το κάθε ένα από αυτά και ποια είναι η εναρκτήρια τιμή της αντίστοιχης δημοπρασίας.

**Τέλος
Παρουσίασης**

Ευχαριστώ για τη προσοχή σας!

Λεβετσοβίτης Πέτρος Α.Μ. 2103