

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

Τίτλος: Δικτυοκεντρική Βάση Δεδομένων

για μια Ασφαλιστική Εταιρεία

Βαρβάρα Φουκάκη (Α.Μ. 3560)

Χρυσάνθη Παναγιωτάκη (Α.Μ. 3546)

Επιβλέπων Καθηγητής: Παπαδάκης Νικόλαος

Ηράκλειο

Μάρτιος, 2018

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε πρώτα απ' όλα τις οικογένειες μας για την στήριξη τους σε όλο τον κύκλο των σπουδών μας και έπειτα τον καθηγητή μας κύριο Παπαδάκη Νικόλαο που μας βοήθησε στην ολοκλήρωση της πτυχιακής μας εργασίας.

Abstract

The purpose of this thesis is the design and administration of a Car Insurance Company's information system. This information system relates all the subject than can be found in a car insurance company such as employees, insurance departments, costumers, car information's, car insurances and accident information.

The main technologies we used to achieve this are XML language, Html5 and PHP.

Σύνοψη

Σκοπός αυτής της πτυχιακής εργασίας είναι η υλοποίηση μίας δικτυοκεντρικής βάσης δεδομένων μίας ασφαλιστικής εταιρείας. Αυτή η δικτυοκεντρική βάση δεδομένων περιέχει όλα τα στοιχεία που πρέπει να έχει μία ασφαλιστική εταιρεία όπως: τα στοιχεία των υπαλλήλων, των τμημάτων της, των πελατών της, τα στοιχεία των αυτοκινήτων, τα ασφαλιστικά της συμβόλαια και ατυχήματα.

Οι βασικές τεχνολογίες που χρησιμοποιήσαμε για την υλοποίηση της παραπάνω εργασίας είναι η XMLγλώσσα, η HTML5 και η PHP.

Περιεχόμενα

Ευχαριστίες	1
Abstract	2
Σύνοψη	3
Περιεχόμενα	4
Πίνακας Εικόνων.....	7
Κεφάλαιο 1	9
Εισαγωγή.....	9
1.1 Περίληψη.....	9
1.1 Κίνητρο για την Διεξαγωγή της Εργασίας	9
1.2 Σκοπός και Στόχοι Εργασίας	9
1.3 Δομή Εργασίας	10
Κεφάλαιο 2.....	11
Μεθοδολογία και Ανάπτυξη Εφαρμογής	11
2.1 Μέθοδος Ανάλυσης και Ανάπτυξης πτυχιακής	11
2.2 Θεωρίες	11
2.2.1 Η Ιστορία και η Εξέλιξη του Διαδικτύου.....	11
2.2.2 Ιστοσελίδες (Web Pages)	17
2.2.3 Γλώσσες σήμανσης	20
2.2.4 SGML (Standard Generalized Markup Language)	21
2.2.5 HTML (HyperText Markup Language)	21
2.2.6 CSS.....	23
2.2.7 JavaScript	25
2.2.8 AJAX.....	27
2.2.9 Βάσεις Δεδομένων	29
2.2.9.1 Δικτυοκεντρικά Συστήματα Βάσεων Δεδομένων	32
2.2.9.2 Συστήματα Διαχείρισης Βάσεων Δεδομένων	32
Κεφάλαιο 3.....	34
Σχέδιο Δράσης για την εκπόνηση της Πτυχιακής Εργασίας.....	34
3.1 Περίληψη.....	34
3.2 Η γλώσσα σήμανσης XML (Extensible Markup Language)	34
3.2.1 Μορφή εγγράφου XML	36
3.2.2 Τύποι αρχείων XML	37
3.2.3 Αναλυτές XML (XML Parsers)	45
3.2.4 Χρήση της XML για διαχείριση και μορφοποίηση κειμένων.....	46

3.2.5 Χρήση της XML ως Βάση Δεδομένων	54
3.2.6 Πλεονεκτήματα και Μειονεκτήματα της χρήσης XML.....	55
3.3 Η γλώσσα προγραμματισμού PHP	56
3.3.1 Περιγραφή εγγράφου PHP	57
3.3.2 Πλεονεκτήματα και Μειονεκτήματα της γλώσσα προγραμματισμού PHP	59
3.3.3 XML και PHP	60
3.3.4 Η επέκταση simpleXML	60
3.3.5 Η επέκταση DOM	62
Κεφάλαιο 4.....	64
Κύριο μέρος Πτυχιακής Εργασίας	64
4.1 Γενική Περιγραφή	64
4.2 Διαδικασία Ανάπτυξης.....	66
4.3 Εγκατάσταση Προγράμματος XAMPP.....	66
4.4 Διαδικασία Υλοποίηση της Βάσης Δεδομένων μας.....	72
4.4.1 Εισαγωγή νέας εγγραφής στην Βάση Δεδομένων.....	73
4.4.2 Επεξεργασία μίας εγγραφής από την Βάση Δεδομένων	77
4.4.3 Διαγραφή μίας εγγραφής από την Βάση Δεδομένων	81
4.5 Υλοποίηση Εφαρμογής	81
4.5.1 Είσοδος στο σύστημα Διαχείρισης.....	84
4.5.2 Αρχική Σελίδα Συστήματος Διαχείρισης	84
4.5.3 Υπάλληλοι.....	85
4.5.3.1 Αρχείο XML για τους Υπαλλήλους	85
4.5.3.2 Εισαγωγή ενός νέου Υπαλλήλου	86
4.5.3.3 Επεξεργασία ενός υπάρχον Υπαλλήλου	87
4.5.3.4 Διαγραφή κόμβου ενός υπάρχον Υπαλλήλου	88
4.5.4 Τμήματα	88
4.5.4.1 Αρχείο XML για τα Τμήματα	88
4.5.4.2 Εισαγωγή ενός νέου Τμήματος	89
4.5.4.3 Επεξεργασία ενός υπάρχον Τμήματος	89
4.5.4.4 Διαγραφή κόμβου ενός υπάρχον Υπαλλήλου	90
4.5.5 Πελάτες	90
4.5.5.1 Αρχείο XML για τους Πελάτες	91
4.5.5.2 Εισαγωγή ενός νέου Πελάτη	91
4.5.5.3 Επεξεργασία ενός υπάρχον Πελάτη	92
4.5.5.4 Διαγραφή κόμβου ενός υπάρχον Πελάτη.....	93

4.5.6 Αυτοκίνητα.....	94
4.5.6.1 Αρχείο XML για τα Αυτοκίνητα.....	94
4.5.6.2 Εισαγωγή ενός νέου Αυτοκινήτου	95
4.5.6.3 Επεξεργασία ενός υπάρχον Αυτοκινήτου	95
4.5.6.4 Διαγραφή κόμβου ενός υπάρχον Αυτοκινήτου	96
4.5.7 Κατηγορίες Αυτοκινήτων.....	96
4.5.7.1 Αρχείο XML για τις Κατηγορίες Αυτοκινήτων	97
4.5.7.2 Εισαγωγή μίας νέας Κατηγορίας Αυτοκινήτου	97
4.5.7.3 Επεξεργασία μίας Κατηγορίας Αυτοκινήτου	98
4.5.7.4 Διαγραφή κόμβου μίας Κατηγορίας Αυτοκινήτου.....	98
4.5.8 Ασφαλιστικά Συμβόλαια.....	99
4.5.8.1 Αρχείο XML για τα Ασφαλιστικά Συμβόλαια.....	99
4.5.8.2 Εισαγωγή ενός νέου Ασφαλιστικού Συμβολαίου	100
4.5.8.3 Επεξεργασία ενός Ασφαλιστηρίου Αυτοκινήτου.....	102
4.5.8.4 Διαγραφή κόμβου ενός Ασφαλιστηρίου Αυτοκινήτου	102
4.5.9 Ανανέωση Συμβολαίων.....	102
4.5.9.1 Αρχείο XML για τις Ανανεώσεις Συμβολαίων	103
4.5.9.2 Ανανέωση ενός Συμβολαίου	104
4.5.10 Ατυχήματα.....	105
4.5.10.1 Αρχείο XML για τα Ατυχήματα.....	106
4.5.10.2 Εισαγωγή ενός νέου Ατυχήματος.....	106
4.5.10.3 Επεξεργασία ενός Ατυχήματος	107
4.5.10.4 Διαγραφή κόμβου ενός Ατυχήματος.....	108
4.5.11 Συγκεντρωτικές Αναφορές.....	108
4.5.11.1 Υπόδειγμα μίας συγκεντρωτικής Αναφοράς.....	109
4.5.12 Μηνύματα Φόρμα Επικοινωνίας.....	110
4.5.12.1 Αρχείο XML για τα Μηνύματα Επικοινωνίας.....	110
4.5.12.2 Ανάγνωση ενός Μηνύματος Επικοινωνίας	111
Κεφάλαιο 5.....	112
Αποτελέσματα.....	112
5.1 Συμπεράσματα - Αποτελέσματα	112
5.2 Βελτιώσεις.....	112
Βιβλιογραφία.....	113
Διαδικτυακές Πηγές	113

Πίνακας Εικόνων

Εικόνα 1 - Παγκόσμια Συνδεσιμότητα 1991	15
Εικόνα 2 - Παγκόσμια Συνδεσιμότητα 1997	15
Εικόνα 3 - Παγκόσμια Συνδεσιμότητα 2012	16
Εικόνα 4 - Παγκόσμια Συνδεσιμότητα 2015	16
Εικόνα 5 - Πρώτη HTML σελίδα.....	18
Εικόνα 6 - Τρόπος Λειτουργίας ενός PHP αρχείου	58
Εικόνα 7 - Εγκατάσταση XAMPP – Εκκίνηση	67
Εικόνα 8 - Επιλογή των λειτουργιών που θέλουμε να εγκαταστήσουμε.....	67
Εικόνα 9 - Επιλογή φακέλου εγκατάστασης του Xampp	68
Εικόνα 10 - Bitnami for Xampp.....	68
Εικόνα 11 - Ένα κλικ πριν την εγκατάσταση του Xampp	69
Εικόνα 12 - Installation of Xampp / Allow Access on Windows Firewall	69
Εικόνα 13 - Ολοκλήρωση εγκατάστασης Xampp και τρέξιμο προγράμματος.....	70
Εικόνα 14 - Επιλογή γλώσσας για το Xampp	70
Εικόνα 15 - Εκκίνηση του Apache Server – Xampp	71
Εικόνα 16 - Εύρεση φακέλου htdocs στον φάκελο Xampp.....	71
Εικόνα 17 - Δημιουργία φακέλου εργασία asfaleies Xampp.....	72
Εικόνα 18 - Όλες οι εγγραφές για τα Τμήματα	73
Εικόνα 19 - Φόρμα Εισαγωγής νέας εγγραφής Τμήματος εταιρείας.....	73
Εικόνα 20 - Φόρμα επεξεργασίας μίας υπάρχον εγγραφής του Τμήματος εταιρείας.....	77
Εικόνα 21 - Αρχική Σελίδα Ιστοτόπου.....	83
Εικόνα 22 - Σελίδα Εισόδου για Διαχειριστές	84
Εικόνα 23 - Αρχική Σελίδα Συστήματος Διαχείρισης	84
Εικόνα 24 - Μενού Υπάλληλοι	85
Εικόνα 25- Φόρμα Εισαγωγής νέου Υπαλλήλου	87
Εικόνα 26 - Φόρμα Επεξεργασίας ενός Υπαλλήλου.....	87
Εικόνα 27 - Μενού Τμήματα	88
Εικόνα 28 - Εισαγωγή ενός Τμήματος.....	89
Εικόνα 29 - Επεξεργασία Τμήματος	90
Εικόνα 30 - Μενού Πελάτες.....	90
Εικόνα 31 - Φόρμα Εισαγωγής Νέου Πελάτη	92
Εικόνα 32 - Μενού Αυτοκίνητα	94
Εικόνα 33 - Εισαγωγή νέου Αυτοκινήτου.....	95
Εικόνα 34 - Επεξεργασία ενός Αυτοκινήτου	96
Εικόνα 35 - Μενού Κατηγορίες Αυτοκινήτων.....	96
Εικόνα 36 - Φόρμα Εισαγωγής μίας νέας Κατηγορίας Αυτοκινήτου	98
Εικόνα 37 - Φόρμα Επεξεργασίας μίας Κατηγορίας Αυτοκινήτων	98
Εικόνα 38 - Μενού Ασφαλιστικά Συμβόλαια	99
Εικόνα 39 - Φόρμα Εισαγωγής ενός νέου Ασφαλιστηρίου Αυτοκινήτου	100
Εικόνα 40 - Φόρμα Επεξεργασίας ενός Ασφαλιστηρίου Αυτοκινήτου	102
Εικόνα 41- Μενού Ανανέωσης Συμβολαίων	103
Εικόνα 42 - Φόρμα Ανανέωσης Συμβολαίου.....	104
Εικόνα 43 - Μενού Ατυχήματα.....	105

Εικόνα 44 - Φόρμα Εισαγωγής ενός νέου Ατυχήματος.....	107
Εικόνα 45 - Φόρμα Επεξεργασίας ενός Ατυχήματος.....	108
Εικόνα 46 - Μενού Συγκεντρωτικές Αναφορές.....	109
Εικόνα 47 - Συγκεντρωτική αναφορά για τα έσοδα/έξοδα κάθε μήνα	109
Εικόνα 48 - Μενού Μηνύματα Επικοινωνίας.....	110
Εικόνα 49 - Φόρμα Ανάγνωσης ενός Μηνύματος Επικοινωνίας.....	111

Κεφάλαιο 1

Εισαγωγή

1.1 Περίληψη

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η διαχείριση μίας δικτυοκεντρικής βάσης δεδομένων για μία ασφαλιστική εταιρεία. Η δικτυοκεντρική μας βάση δεδομένων συνδέει όλες τις πιθανές οντότητες που χρειάζονται για την λειτουργία μίας ασφαλιστικής εταιρείας όπως είναι όλα τα στοιχεία των υπαλλήλων, των πελατών, των ασφαλιστικών συμβολαίων, τα ατυχήματα που έχουν γίνει στα ασφαλισμένα αυτοκίνητα της εταιρείας κ.λ.π.

Οι οντότητες που χρειαστήκαμε για την υλοποίηση της πτυχιακής μας εργασίας είναι οι πληροφορίες για τα τμήματα της εταιρείας, τους πελάτες, τις κατηγορίες των αυτοκινήτων, τα ασφαλιστήρια συμβόλαια, πληροφορίες για τα αυτοκίνητα, όπως και για τα ατυχήματα που συμβαίνουν με ασφαλισμένα αυτοκίνητα και τέλος για τους υπαλλήλους της εταιρείας.

Για την διεξαγωγή της εργασίας υλοποιήσαμε την βάση δεδομένων μας με την χρήση της γλώσσας σήμανσης XML δημιουργώντας τα .xml αρχεία για κάθε μία από τις οντότητες. Και τέλος χρησιμοποιήσαμε την γλώσσα σήμανσης υπερκειμένου HTML5 και την γλώσσα προγραμματισμού PHP για την δημιουργία ενός δυναμικού περιεχομένου για την εμφάνιση των πληροφοριών και τέλος χρησιμοποιήσαμε τις επεκτάσεις της PHP τις SimpleXML και DOM για να μπορέσουμε να δημιουργήσουμε, να επεξεργαζόμαστε και να παίρνουμε τα δεδομένα από την βάση μας.

1.1 Κίνητρο για την Διεξαγωγή της Εργασίας

Με βάση τα σημερινά δεδομένα, η γλώσσα XML αποτελεί ένα από τα πιο ευρέως δεδομένα πρότυπα αναπαράστασης δεδομένων στον παγκόσμιο ιστό. Ένα μεγάλο ποσοστό εφαρμογών του διαδικτύου διαχειρίζεται πληροφορίες ημιδομημένης φύσεως και η πολυδιάστατη XML είναι κατάλληλη για την αναπαράστασή τους. Ένα κείμενο XML, σαν βάση δεδομένων, συγκεντρώνει ορισμένα πλεονεκτήματα και μειονεκτήματα έναντι των αρχείων των κλασσικών βάσεων δεδομένων. Με την γλώσσα XML να αυξάνεται σε ζήτηση στον τομέα ανάπτυξης και διαχείρισης ως βάση δεδομένων, μας οδήγησε στην επιλογή του συγκεκριμένου θέματος της πτυχιακής μας εργασίας.

1.2 Σκοπός και Στόχοι Εργασίας

Η παρούσα πτυχιακή εργασία έχει ως σκοπό την δημιουργία και την διαχείριση μίας δικτυοκεντρικής βάσης δεδομένων μίας ασφαλιστικής εταιρείας χρησιμοποιώντας σαν βάση δεδομένων την γλώσσα XML.

Οι βασικοί στόχοι για τους οποίους την αναλάβαμε είναι για να δείξουμε ότι η XML γλώσσα μπορεί να χρησιμοποιηθεί ως μία ικανοποιητική βάση δεδομένων, που δεν έχει να ζηλέψει τίποτα από την MySQL βάση δεδομένων.

1.3 Δομή Εργασίας

Στα επόμενα κεφάλαια θα αναφερθούμε:

- **Κεφάλαιο 2:** Θα παρουσιάσουμε αναλυτικά την μεθοδολογία υλοποίησης, δηλαδή θα αναφερθούμε στην θεωρία των τεχνολογιών που χρησιμοποιήθηκαν στην παρούσα εργασία για τον σχεδιασμό και την υλοποίηση της.
- **Κεφάλαιο3:** Ανάλυση των βασικών τεχνολογιών που τελικά χρησιμοποιήσαμε, δηλαδή την γλώσσα XML ως βάση δεδομένων καθώς και την PHP γλώσσα προγραμματισμού, που ήταν η βασική γλώσσα διαχείρισης του συστήματος μας.
- **Κεφάλαιο 4:** Εξηγούμε τα εργαλεία που χρησιμοποιήσαμε για την συγκεκριμένη εργασία. Καθώς επίσης, αναλύουμε και κάποιες από τις βασικές λειτουργίες της.
- **Κεφάλαιο 5:** Τέλος παραθέτουμε συμπεράσματα και μελλοντικά σχέδια, τα οποία μπορούν να χρησιμοποιηθούν ως βάση από την εργασίας μας.

Κεφάλαιο 2

Μεθοδολογία και Ανάπτυξη Εφαρμογής

2.1 Μέθοδος Ανάλυσης και Ανάπτυξης πτυχιακής

Για την ολοκλήρωση της παρούσας πτυχιακής εργασίας χρειάστηκε να ακολουθήσαμε τις παρακάτω ενέργειες:

1. Να ψάξουμε να βρούμε πληροφορίες για τις τεχνολογίες που θα χρησιμοποιήσουμε
2. Έπειτα να εγκαταστήσουμε το Xampp για να στήσουμε τον localhost μας
3. Να κάνουμε ανάλυση των οντοτήτων της ασφαλιστικής εταιρείας
4. Να σχεδιάσουμε τη βάση δεδομένων μας
5. Και τέλος, να υλοποιήσουμε όλα τα παραπάνω.

2.2 Θεωρίες

2.2.1 Η Ιστορία και η Εξέλιξη του Διαδικτύου

Εισαγωγή

Το Ίντερνετ έχει γίνει η αιτία μιας επανάστασης στον κόσμο των υπολογιστών και γενικά των επικοινωνιών όσο κανένα άλλο τεχνολογικό μέχρι σήμερα. Βρίσκεται ανάμεσα στα επιτεύγματα όπως είναι ο τηλεγράφος, το τηλέφωνο, το ραδιόφωνο και ο υπολογιστής. Το Internet θεωρείται τεχνολογία για παγκόσμια επικοινωνία, μηχανισμός μετάδοσης πληροφοριών και ένα μέσο συνεργασίας και αλληλεπίδρασης χρηστών και υπολογιστών χωρίς εμπόδιο την γεωγραφική τους τοποθεσία.

Πως ξεκίνησε το Ίντερνετ; Τα θεμέλια έθεσε το Υπουργείο Άμυνας των ΗΠΑ πριν σαράντα χρόνια περίπου, στις αρχές της δεκαετίας του '70, χρησιμοποιώντας ένα δίκτυο υπολογιστών που ονομαζόταν ARPANET¹. Όμως η χρήση του Ίντερνετ ξεκίνησε στις αρχές της δεκαετίας του '90 όταν άρχισε η ανάπτυξη του Παγκόσμιου Ιστού. Το ARPANET ωστόσο ήταν ένα μεγάλο δίκτυο ευρείας περιοχής, το οποίο δημιουργήθηκε από την Υπηρεσία Προηγμένων Ερευνητικών Εργασιών του Υπουργείου Άμυνας των ΗΠΑ, που ονομαζόταν ARPA.

Από το ARPANET στο Ίντερνετ

Τον Οκτώβριο 1972: Στην πρώτη Διεθνή Διάσκεψη για Θέματα Υπολογιστών και Επικοινωνιών, η οποία έγινε στην Ουάσιγκτον DC των ΗΠΑ, οι επιστήμονες της ARPA

¹ Διαδίκτυο: ARPAnet: Εργαστήριο Εφαρμογών πληροφορικής στα ΜΜΕ, ΑΠΘ
<http://pacific.jour.auth.gr/internet/page%201.2.htm>

έθεσαν το ARPANET δημόσιο, συνδέοντας υπολογιστές μεταξύ τους οι οποίοι βρίσκονταν σε 40 διαφορετικές τοποθεσίες, για να κάνουν μία επίδειξη της λειτουργίας του συστήματος. Η κίνηση αυτή ήταν το ερέθισμα για να ξεκινήσουν επιπλέον έρευνες στην επιστημονική κοινότητα και δεν άργησαν να εμφανιστούν κι άλλα δίκτυα. Ωστόσο, οι επιστήμονες της ARPA δεν σταμάτησαν να εργάζονται για την βελτίωση του δικού τους συστήματος και την επέκταση των δυνατοτήτων του.

Το 1972, δημιούργησαν ένα νέο πρόγραμμα που επέτρεπε την αποστολή μηνυμάτων μέσω του δικτύου, έχοντας έτσι την δυνατότητα μίας απευθείας επικοινωνίας πρόσωπο με πρόσωπο, δηλαδή το σημερινό e-mail.

Το 1974, είχαμε την δημιουργία του σημερινού TCP/IP η ανάπτυξη του οποίου ήταν ένα σημαντικό κατόρθωμα για την ανάπτυξη της δικτύωσης. Κατάφεραν δηλαδή να αναπτύξουν μια κοινή γλώσσα που επιτρέπει την επικοινωνία διαφορετικών δικτύων μεταξύ τους όπου δημιουργούνται προδιαγραφές για τη μεταφορά αρχείων μεταξύ υπολογιστών με την χρήση του FTP (File Transfer Protocol). Έπειτα με την πάροδο του χρόνου άρχισαν να συνδέονται με το ARPANET, ιδρύματα και από άλλες χώρες, με πρώτες την Αγγλία και την Νορβηγία.

Το 1983, ξεκινάει η εποχή ενός παγκόσμιου δικτύου της ακαδημαϊκής κοινότητας με το πρωτόκολλο TCP/IP να γίνεται πρότυπο από το Υπουργείο Άμυνας των ΗΠΑ. Επίσης έχουμε την δημιουργία του λειτουργικού συστήματος UNIX, το οποίο έχει ως αρχή το πρωτόκολλο TCP/IP, έχοντας έτσι την γρήγορη εξάπλωση της δικτύωσης των υπολογιστών. Με εκατοντάδες υπολογιστές Πανεπιστημίων να έχουν συνδεθεί στο ARPANET, λόγω της επιβάρυνσης του, το 1983 χωρίστηκε σε δύο τμήματα: MILNET για μόνο στρατιωτικές επικοινωνίες και σε νέο ARPANET για μόνο πανεπιστημιακή χρήση.

Το 1985, το Εθνικό Επιστημονικό Ίδρυμα, δημιούργησε το δικό του γρήγορο δίκτυο με όνομα NSFNET, το οποίο και αυτό βασίστηκε στο πρωτόκολλο TCP/IP, με σκοπό σύνδεσης πέντε υπεριολογιστών μεταξύ τους, με την υπόλοιπη επιστημονική κοινότητα. Έτσι, με όλο και περισσότερες χώρες να συνδέονται στο NSFNET, με όλο και χιλιάδες πανεπιστήμια και οργανισμούς να χρησιμοποιούν τα δικά τους δίκτυα και να συνδέονται πάνω σε αυτό το παγκόσμιο δίκτυο έχουμε την κατάργηση του ARPANET το 1990, και να γίνεται σε όλους γνωστό ως Ίντερνετ.

Από το Ίντερνετ στον Παγκόσμιο Ιστό

Το 1984, έχουμε την εισαγωγή των DNS (Domain Name Servers)². Μέχρι τότε ο κάθε κεντρικός υπολογιστής είχε ένα συγκεκριμένο όνομα, ανάμεσα από μία μοναδική λίστα με ονόματα και διευθύνσεις. Το νέο αυτό σύστημα ονομάτων χωρών στην ουσία είναι ένα ιεραρχικό σύστημα ονοματοδοσίας για τα δίκτυα υπολογιστών που χρησιμοποιούν το πρωτόκολλο IP. Το σύστημα DNS αντιστοιχεί τα ονόματα των κεντρικών υπολογιστών σε διευθύνσεις IP καθιστώντας τα πιο ευκολομνημόνευτα.

Με την δεκαετία του '90 να έρχεται, έχουμε την δημιουργία ενός παγκόσμιου δικτύου για όλους, με ολόένα και περισσότερες χώρες να συνδέονται στο NSFNET.

Το 1990, έχουμε την σύνδεση της Ελλάδας στο NSFNET.

² Domain Name System, Wikipedia, 2014, http://el.wikipedia.org/wiki/Domain_Name_System

Το 1990, έχουμε την πρώτη μηχανή αναζήτησης, το Archie, που αναπτύχθηκε στο Πανεπιστήμιο McGill του Montreal. Χρησιμοποιήθηκε για την εύρεση και την ανάκτηση διάφορων αρχείων σε υπολογιστές.

Το 1993, ο Tim Berners-Lee αναπτύσσει το World Wide Web³, δηλαδή τον Παγκόσμιο Ιστό, το οποίο παρουσιάστηκε στο εργαστήριο CERN στην Ελβετία. Ο Παγκόσμιος Ιστός είναι ένα ανοιχτό σύστημα διασυνδεδεμένων πληροφοριών και πολυμεσικού περιεχομένου, το οποίο δίνει την δυνατότητα στους χρήστες του να βρίσκουν πληροφορίες κάνοντας αναζήτηση από έγγραφο σε έγγραφο, τα οποία βρίσκονται αποθηκευμένα σε εκατοντάδες υπολογιστές που έχουν Ίντερνετ από όλον τον κόσμο και η εμφάνισή τους γίνεται σε ηλεκτρονικές σελίδες.

Παράλληλα με την εξέλιξη του Παγκόσμιου Ιστού έχουμε και την εμφάνιση διαφόρων εμπορικών δικτύων που ανήκουν σε εταιρείες παροχής υπηρεσιών Ίντερνετ (Internet Service Providers - ISP) και δίνουν πρόσβαση Ίντερνετ για όλους. Έχοντας έναν Ηλεκτρονικό Υπολογιστή και modem, μπορούσες πλέον να συνδεθείς στο Ίντερνετ, με το κόστος χρήσης ολοένα και να μειώνεται.

Το 1995, το NSFNET έχουμε την επίσημη κατάργηση του και το φορτίο του μεταφέρθηκε σε εμπορικά δίκτυα.

Παγκόσμιος Ιστός και Ίντερνετ, οι διαφορές τους

Όπως αναφέραμε και παραπάνω είναι ένα ανοιχτό σύστημα διασυνδεδεμένων πληροφοριών και πολυμεσικού περιεχομένου. Πολλοί θεωρούν ότι ο όρος Ίντερνετ και Παγκόσμιος Ιστός (World Wide Web) είναι το ίδιο πράγμα, όμως στην πραγματικότητα μιλάμε για δύο διαφορετικά πράγματα που σχετίζονται μεταξύ τους. Το Ίντερνετ είναι στην ουσία μία δικτυακή υποδομή, ένα τεράστιο δηλαδή δίκτυο που έχει άλλα δίκτυα και συνδέει εκατομμύρια υπολογιστές μεταξύ τους από όλον τον κόσμο. Οι υπολογιστές αυτοί για όσο είναι συνδεδεμένοι στο Ίντερνετ μπορούν να επικοινωνήσουν μεταξύ τους και να ανταλλάξουν πληροφορίες. Οι πληροφορίες αυτές μεταδίδονται μέσω του Ίντερνετ χρησιμοποιώντας ποικιλία από πρωτόκολλα όπως είναι το TCP/IP ή το UDP.

Ο Παγκόσμιος Ιστός από την άλλη, είναι απλά ένας τρόπος εμφάνισης και πρόσβασης των πληροφοριών μέσω του Ίντερνετ. Το πρωτόκολλο που χρησιμοποιεί ο Παγκόσμιος Ιστός είναι το Πρωτόκολλο Μεταφοράς Υπερκειμένου HTTP, το οποίο είναι το κύριο πρωτόκολλο επικοινωνίας που χρησιμοποιείται από τους φυλλομετρητές (Browsers) και χρησιμοποιείται για την μεταφορά των δεδομένων ανάμεσα σε έναν server και έναν client. Ο Παγκόσμιος Ιστός χρησιμοποιεί επίσης τους φυλλομετρητές (Browsers), όπως είναι οι Mozilla Firefox, Google Chrome ή Internet Explorer, για να αποκτάει πρόσβαση στα έγγραφα του διαδικτύου, δηλαδή τις γνωστές σε όλους μας Ιστοσελίδες (Web Pages), οι οποίες μπορούν να περιέχουν απλό κείμενο, εικόνες, γραφικά, ήχους ακόμα και βίντεο.

Και τέλος, ο Παγκόσμιος Ιστός αποτελεί σημαντικό κομμάτι για την διασπορά των πληροφοριών στο Ίντερνετ. Το Ίντερνετ από την άλλη, και όχι ο Παγκόσμιος Ιστός, χρησιμοποιείται για την ηλεκτρονική αλληλογραφία (email) με το πρωτόκολλο SMTP, στις ομάδες ειδήσεων (newsgroup), στα άμεσα μηνύματα και στο πρωτόκολλο μεταφοράς FTP.

³ World Wide Web, Wikipedia, 2018, https://en.wikipedia.org/wiki/World_Wide_Web & The birth of the web, <https://home.cern/topics/birth-web>

Έτσι καταλήγουμε ότι ο Παγκόσμιος Ιστός, είναι ένα πολύ μεγάλο και σημαντικό κομμάτι του Ίντερνετ, αλλά δεν θα πρέπει να τα μπερδεύουμε μεταξύ τους.

Η ιστορία του Παγκόσμιου Ιστού

Ο Παγκόσμιος Ιστός δημιουργήθηκε από τον Βρετανό Tim Berners-Lee⁴ το 1989, καθώς εκείνη την εποχή εργαζόταν στον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (CERN) στην Ελβετία. Ο ίδιος ο Lee ήταν αυτός που ονόμασε την εφεύρεση του World Wide Web, το γνωστό σε όλους μας “www”. Στόχος του ήταν να μπορεί ο καθένας μας να ανταλλάσσει πληροφορίες και ιδέες άμεσα και είναι προσβάσιμες από όλους. Ο Παγκόσμιος Ιστός, ως πληροφοριακό σύστημα χρησιμοποιεί ένα συγκεκριμένο μοντέλο δεδομένων, το οποίο βασίζεται σε κόσμους και υπερσυνδέσμους.

Το 1993, ήταν το έτος δημιουργίας του πρώτου γραφικού φυλλομετρητή (web browser), ο Mosaic, ο οποίος αναπτύχθηκε στο NCSA και στο πανεπιστήμιο του Illinois. Ο Mosaic προκάλεσε τεράστιο ενδιαφέρον σε όλο τον κόσμο για τον Παγκόσμιο Ιστό.

Με το ενδιαφέρον αυτό, όλο και να αυξάνεται, το 1994, ένας από τους δημιουργούς του Mosaic ο Marc Andreessen, αποχώρησε από το NCSA για να ιδρύσει μαζί με κάποιους άλλους την εταιρεία Netscape Communications Corp, όπου εκεί έχουμε την δημιουργία ενός νέου γραφικού φυλλομετρητή (web browser), τον Netscape Navigator. Η διανομή αυτού του φυλλομετρητή έγινε δωρεάν για τα λειτουργικά συστήματα UNIX, Windows και Macintosh OS. Έτσι, αποκτήθηκε ένα παγκόσμιο ενδιαφέρον για το Ίντερνετ και τον Παγκόσμιο Ιστό και έχουμε και το ξεκίνημα μίας επιχειρησιακής εποχής του Ίντερνετ. Στον χώρο των φυλλομετρητών μπαίνει και η Microsoft, το 1995, με την πρώτη έκδοση του γνωστού Internet Explorer. Από εκεί και μετά ξεκίνησε και το ηλεκτρονικό εμπόριο (e-commerce).

Χρήση του Ίντερνετ τότε και σήμερα

Σήμερα, το μεγαλύτερο μέρος του πληθυσμού της Γης ζει σε περιοχές που είναι συνδεδεμένες στο Ίντερνετ. Ζούμε σε μία εποχή που οι πληροφορίες είναι αμέτρητες και προσβάσιμες από όλους με επτά στους δέκα Έλληνες να χρησιμοποιούν το Ίντερνετ (69,7%), και έξι στους δέκα (60,4%) σε καθημερινή βάση⁵.

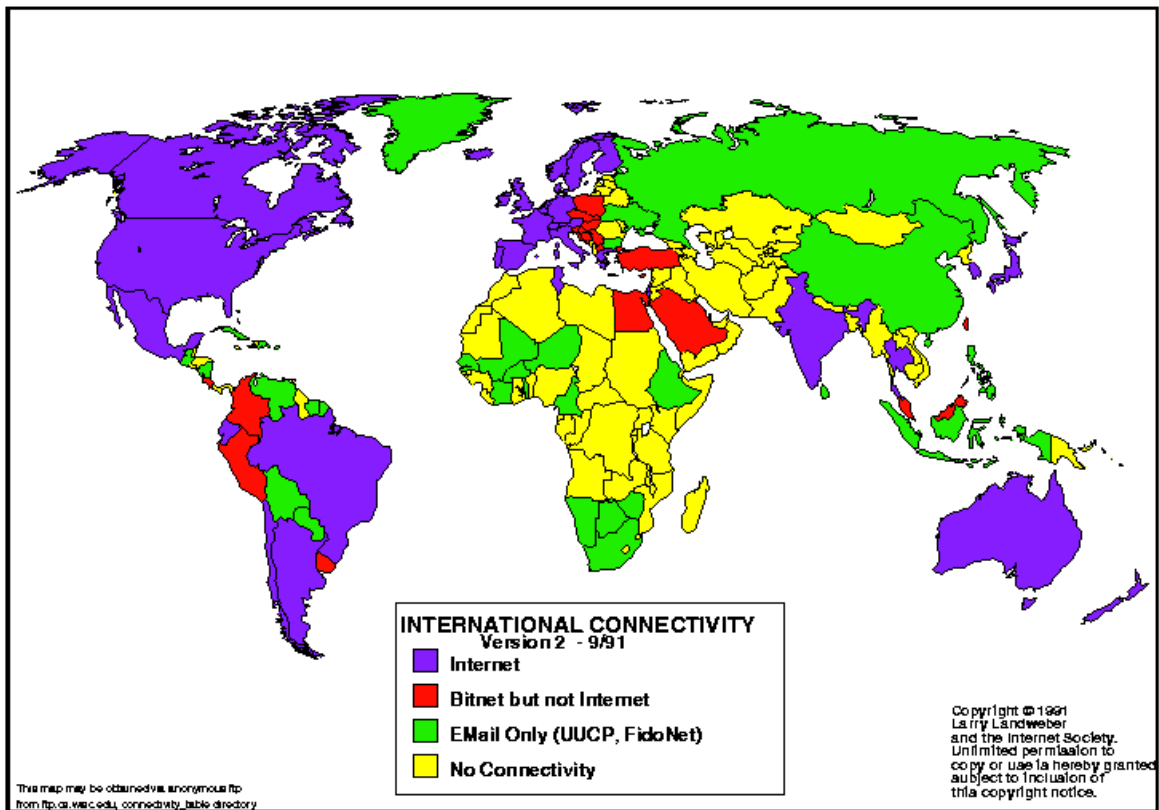
Είναι ξεκάθαρο πλέον, ότι το Ίντερνετ δεν είναι ένα δίκτυο μόνο για φοιτητές και ερευνητές, αλλά επεκτείνεται και αλλάζει την καθημερινότητα όλων μας.

Παρακάτω παραθέτουμε αναλυτικές εικόνες με την παγκόσμια προσβασιμότητα στο Ίντερνετ για το 1991 όπου η χρήση του Ίντερνετ είχε μόλις αρχίσει να ακμάζει και από το 1997 και έπειτα, υπάρχει μία ραγδαία αύξηση της χρήσης του Ίντερνετ για να φτάσουμε στο πιο κοντινό σήμερα το 2012 και το 2015 όπου το Ίντερνετ έχει γίνει παγκόσμιο φαινόμενο.

- Έτος 1991

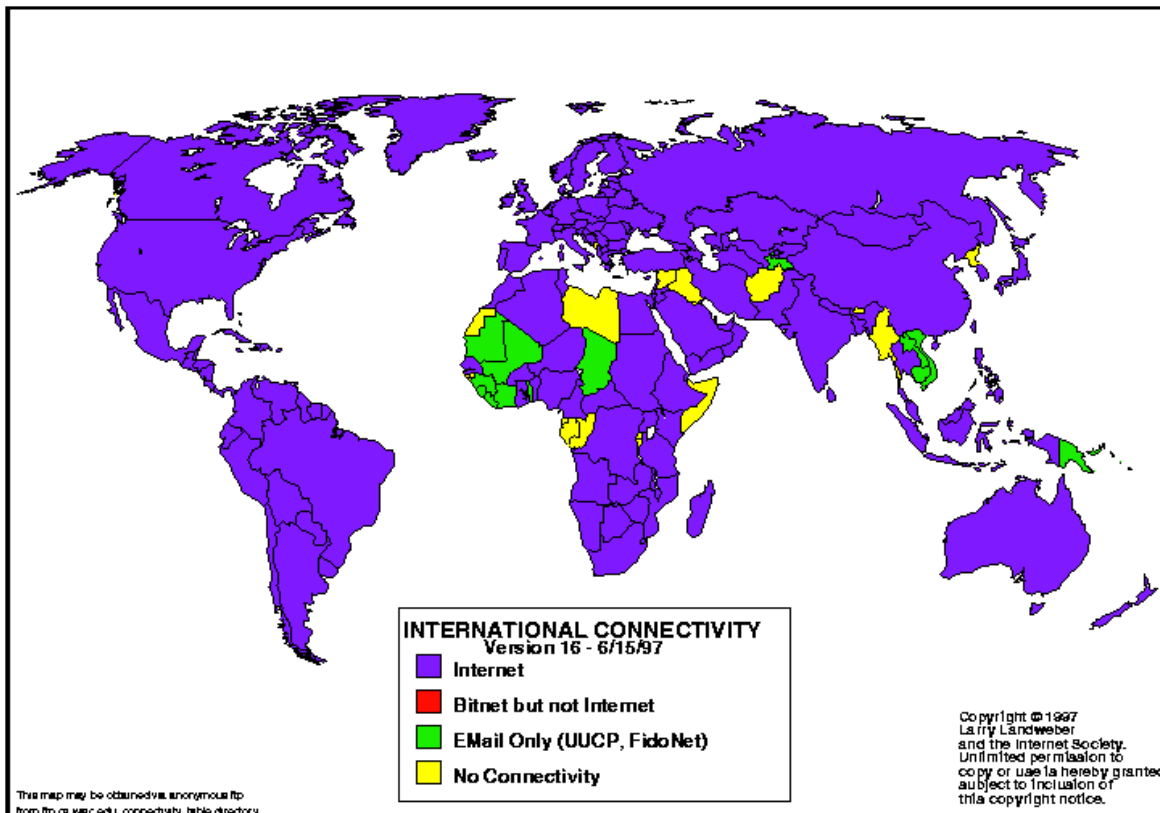
⁴ The World Wide Web: Past, Present and Future, 1996, <http://www.w3.org/People/Berners-Lee/1996/ppf.html>

⁵ «Οριμάζει» η χρήση του Ίντερνετ στην Ελλάδα, Naftemporiki.gr Online Εφημερίδα, 2015, <http://www.naftemporiki.gr/story/914535/orimazei-i-xrisi-tou-internet-stin-ellada>



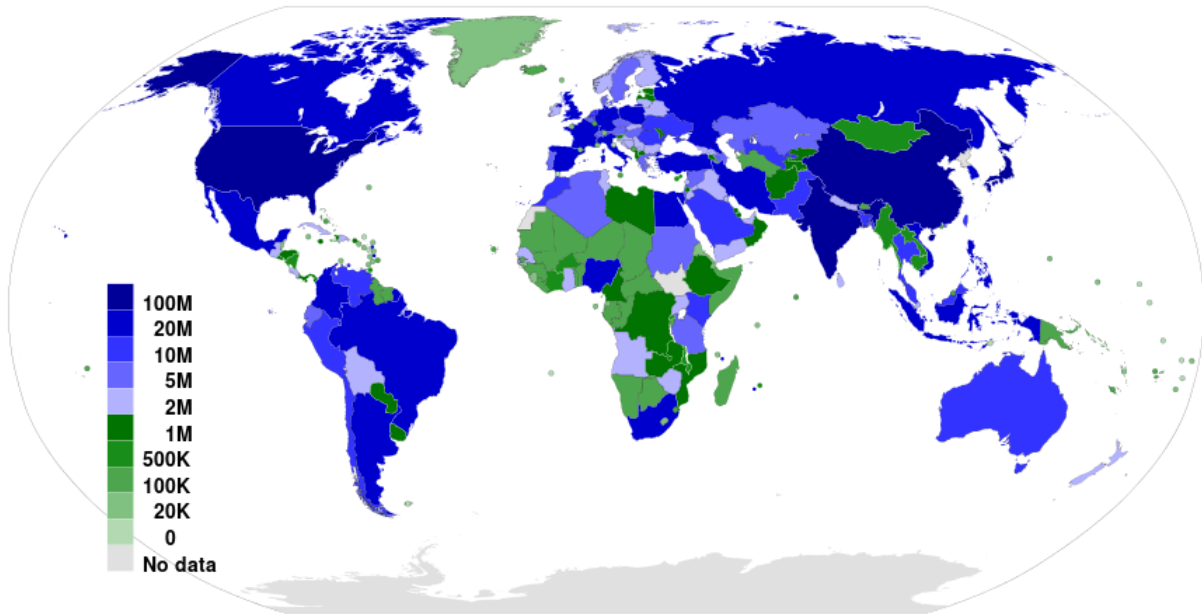
Εικόνα 1 - Παγκόσμια Συνδεσιμότητα 1991

- Έτος 1997



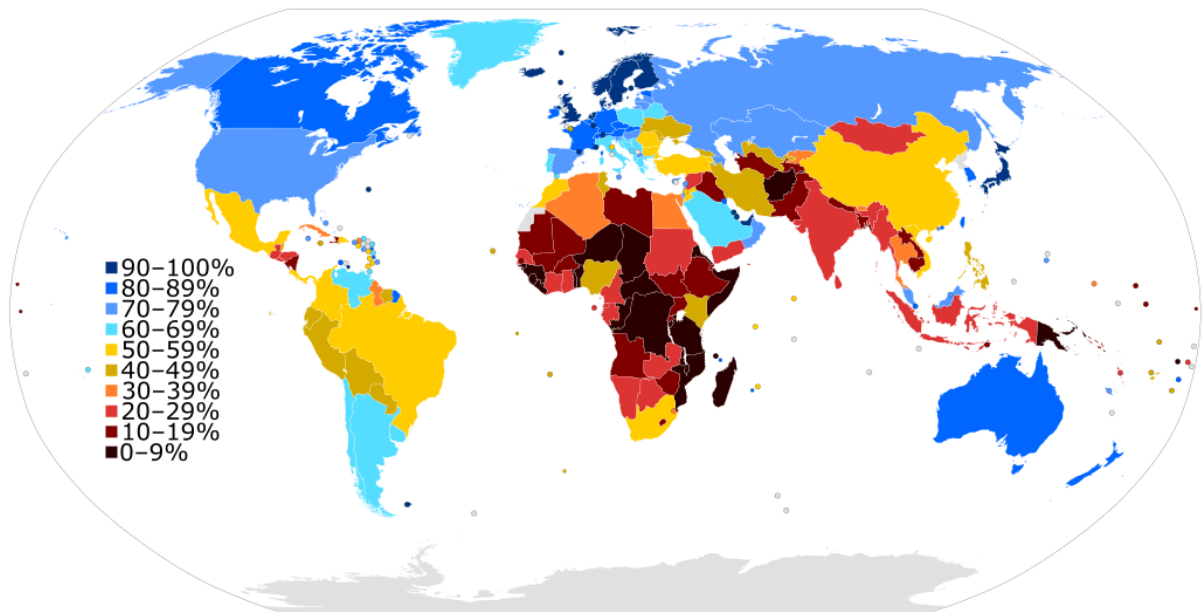
Εικόνα 2 - Παγκόσμια Συνδεσιμότητα 1997

- Έτος 2012



Εικόνα 3 - Παγκόσμια Συνδεσιμότητα 2012

- Έτος 2015



Εικόνα 4 - Παγκόσμια Συνδεσιμότητα 2015

2.2.2 Ιστοσελίδες (Web Pages)

2.2.2.1 Εισαγωγή

Η Ιστοσελίδα είναι ένα είδος εμπλουτισμένου εγγράφου το οποίο είναι ιδανικό για τον Παγκόσμιο Ιστό και για τους φυλλομετρητές (web browsers). Ιστοσελίδα, όπως το λέει και η λέξη, είναι μια σελίδα ιστού, είναι ένα έγγραφο που μπορεί να περιέχει απλό κείμενο, υπερκείμενο, εικόνες, βίντεο, ήχους, συνδέσμους (links) και άλλα διάφορα στοιχεία. Ένας φυλλομετρητής μπορεί να εμφανίσει μία ιστοσελίδα σε μία οθόνη ενός ηλεκτρονικού υπολογιστή ή σε μία κινητή συσκευή, χωρίς να υπάρχει περιορισμός του λειτουργικού τους συστήματος.

Η ιστοσελίδα συνήθως αναφέρεται σε αυτό που είναι ορατό, αλλά ο όρος επίσης μπορεί να αναφέρεται σε ένα έγγραφο υπολογιστή, το οποίο συνήθως είναι γραμμένο στην γλώσσα σήμανσης HTML ή σε οποιαδήποτε άλλη γλώσσα σήμανσης κειμένου. Για την εμφάνιση των ιστοσελίδων, οι φυλλομετρητές συντονίζουν τα διάφορα αρχεία που περιέχουν τα στοιχεία πόρων του ιστού, για την συγκεκριμένη ιστοσελίδα, όπως είναι τα φύλλα στυλ (stylesheets), scripts και εικόνες. Η μορφή μίας τυπικής ιστοσελίδας παρέχει υπερκείμενο, το οποίο περιλαμβάνει μία μπάρα πλοήγησης ή μία πλαϊνή μπάρα μενού που συνδέεται με άλλες ιστοσελίδες μέσω υπερσυνδέσμων, που συχνά αναφέρονται ως σύνδεσμοι (links). Με την βοήθεια του πρωτόκολλο HTTP (Hypertext Transfer Protocol) επιτυγχάνεται η πρόσβαση και η μεταφορά μίας ιστοσελίδας.

Η δημιουργία πολλών ιστοσελίδων μαζί, με κοινό θέμα και ένα μενού με υπερσυνδέσμους που οδηγούν στις υπόλοιπες ιστοσελίδες, οι οποίες όλες φιλοξενούνται στο ίδιο Domain Name και Server του Παγκόσμιου Ιστού, ονομάζεται Ιστότοπος ή Ιστοχώρος (Web Site). Μπορούμε να επισκεφτούμε έναν Ιστότοπο χρησιμοποιώντας την διεύθυνση της ιστοσελίδας, η οποία βασίζεται στην υπηρεσία WWW και χρησιμοποιεί το πρωτόκολλο http. Όταν θέλουμε λοιπόν να επισκεφτούμε μία ιστοσελίδα αρκεί να θυμόμαστε το domain name της και να χρησιμοποιήσουμε το url: `http://WWW.domainname.TLD`⁶, για παράδειγμα αν θέλουμε να επισκεφτούμε την ιστοσελίδα της Microsoft μπαίνουμε στο `http://www.microsoft.com`.

Οι ιστότοποι μπορούν να έχουν αρκετές λειτουργίες και μπορούν επίσης να χρησιμοποιηθούν σε διάφορες μορφές, όπως είναι ένας προσωπικός ιστότοπος ή ένας επαγγελματικός ιστότοπος μίας εταιρείας και πολλοί άλλοι. Συνήθως, όλοι οι ιστότοποι έχουν ένα συγκεκριμένο θέμα ή σκοπό, είτε είναι για ψυχαγωγία είτε για κοινωνική δικτύωση μέχρι και την παροχή ειδήσεων ή εκπαίδευσης. Επίσης, ανάλογα που απευθύνεται ο ιστότοπος διακρίνεται σε δημόσιο και ιδιωτικός ιστότοπος. Ο δημόσιος ιστότοπος είναι αυτός που είναι προσβάσιμος από όλους και συνιστούν την χρήση του WWW, ενώ ο ιδιωτικός ιστότοπος, μπορεί να είναι ένας ιστότοπος μίας εταιρείας που αφορά μόνο τους υπαλλήλους της και θέλει να έχουν πρόσβαση μόνο εκείνοι, οπότε είναι συνήθως μέρος ενός ενδοδικτύου (LAN).

Οι ιστοσελίδες χωρίζονται και αυτές σε κατηγορίες, οι δύο μεγαλύτερες και βασικότερες είναι οι στατικές ιστοσελίδες (static webpage) και οι δυναμικές ιστοσελίδες (dynamain

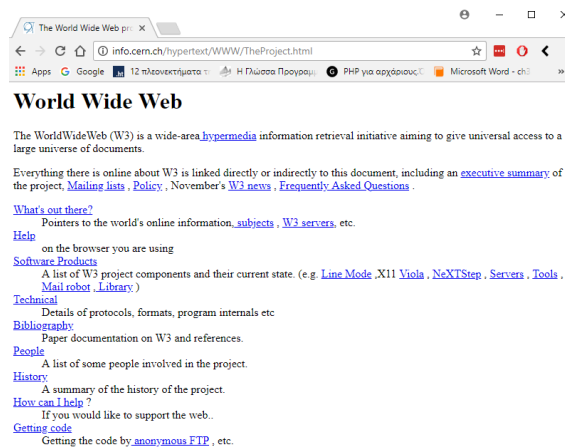
⁶ Top-level domain, Wikipedia, https://en.wikipedia.org/wiki/Top-level_domain

webpage)⁷. Οι διαφορές που έχουν δεν είναι στο θέμα εμφάνισης της ιστοσελίδας αλλά στον τρόπο λειτουργίας και στις δυνατότητες τους.

Οι στατικές ιστοσελίδες είναι οι ιστοσελίδες που έχουν μόνιμο περιεχόμενο, αλλά αυτό δεν σημαίνει ότι και το περιεχόμενό τους πρέπει να είναι στατικό, μπορούν για παράδειγμα να περιέχουν κινούμενα γραφικά.

Οι δυναμικές ιστοσελίδες από την άλλη είναι οι ιστοσελίδες όπου ο ίδιος ο χρήστης μπορεί να αλληλοεπιδρά με την ίδια την σελίδα. Η στατική ιστοσελίδα έχει τα δικά της πλεονεκτήματα που δεν υπάρχουν στις δυναμικές και το αντίθετο, γι' αυτό δεν είναι σωστό να λέμε ότι η μία είναι καλύτερη από την άλλη.

Η πρώτη ιστοσελίδα⁸ δημιουργήθηκε από τον δημιουργό του Παγκόσμιου Ιστού (World Wide Web) Tim Berners-Lee, στις 6 Αυγούστου του 1991 στο CERN. Η οποία «ανέβηκε» ξανά online για να γιορτάσουν τα 20 χρόνια του Παγκόσμιου Ιστού. Ήταν η πρώτη στατική ιστοσελίδα με απλό κείμενο και υπερσυνδέσμους που οδηγούσαν στις άλλες σελίδες του ιστοτόπου. Μπορείτε να επισκεφτείτε την πρώτη ιστοσελίδα πηγαίνοντας στον παρακάτω σύνδεσμο: <http://info.cern.ch/hypertext/WWW/TheProject.html>.



Εικόνα 5 - Πρώτη HTML σελίδα

2.2.2.2 Στατικές Ιστοσελίδες

Οι στατικές ιστοσελίδες διακρίνονται από την μονιμότητα του περιεχομένου τους, αλλά και της εμφάνισής τους (Layout). Οι ιστοσελίδες αυτές είναι απλά HTML έγγραφα που περιέχουν απλό κείμενο, εικόνες και συνδέσμους. Λόγω του ότι το περιεχόμενό τους είναι σταθερό, είναι κατάλληλες για ιστοσελίδες που δεν υπάρχει η ανάγκη για συχνές αλλαγές ή τροποποίηση του περιεχομένου τους.

Η τροποποίηση των δεδομένων (κείμενα, φωτογραφίες κλπ) σε μία στατική ιστοσελίδα γίνεται μόνο με την παρέμβαση του χειριστή χειροκίνητα, με την βοήθεια κάποιου ειδικού προγράμματος κατασκευής ιστοσελίδων.

Τα θετικά μίας στατικής ιστοσελίδας:

⁷ World Wide Web Consortium, 2015, <http://www.w3.org/standards/webdesign/>

⁸ Αυτή είναι η πρώτη ιστοσελίδα που κατασκευάστηκε στην Ιστορία, [iefimerida.gr](http://www.iefimerida.gr) Online Εφημερίδα, 2013, <http://www.iefimerida.gr/news/> αυτή-είναι-η-πρώτη-ιστοσελίδα-που-κατασκευάστηκε-στην-ιστορία-εικόνα

- Χαμηλό κόστος κατασκευής και συντήρησης της
- Δεν απαιτεί μεγάλος χώρος σε φιλοξενία σε κάποιον Server
- Αναπτύσσεται και σχεδιάζεται γρηγορότερα
- Η σελίδα «φορτώνει» πιο γρήγορα

Τα αρνητικά της:

- Χρειάζονται προγραμματιστικές γνώσεις για την ανανέωση του περιεχομένου της
- Περιορισμός των σελίδων που μπορεί να έχει ένας ιστότοπος
- Δεν είναι ιδανική για συχνή αλλαγή του περιεχομένου της

Μεγάλο πλεονέκτημα των στατικών ιστοσελίδων είναι επίσης ότι είναι πιο φιλικές στο SEO, έναντι των δυναμικών ιστοσελίδων. Το SEO (Search Engine Optimization)⁹ που στα Ελληνικά σημαίνει Βελτιστοποίηση Σελίδας για τις μηχανές αναζήτησης, είναι πιο φιλικές για τις μηχανές αναζήτησης και αυτό συμβαίνει γιατί στις στατικές υπάρχει η δυνατότητα εισαγωγής στον κώδικα HTML, των meta-tags τα οποία αναγνωρίζονται από τις μηχανές αναζήτησης, και τα οποία βοηθούν τις μηχανές αναζήτησης να κατατάσσουν την στατική σελίδα πιο πάνω από μία δυναμική που δεν έχει ακριβώς αυτήν την δυνατότητα. Αυτό είναι πολύ σημαντικό διότι αυξάνει την επισκεψιμότητα ενός ιστοχώρου, άρα και το κέρδος της. Όσο καλύτερη θέση κατάταξης έχει ένας ιστότοπος στις μηχανές αναζήτησης τόσο μεγαλύτερο πλεονέκτημα έχει από τους ανταγωνιστές της.

Στο πλεονέκτημα των στατικών ιστοσελίδων είναι ότι για την επεξεργασία των ιστοσελίδων ο χρήστης πρέπει να έχει γνώσεις προγραμματισμού, για να μπορεί να ανανεώσει το περιεχόμενο της, γιατί πρέπει να γίνεται επέμβαση μέσω κώδικα.

Η επιλογή μίας στατικής ιστοσελίδας γίνεται όταν δεν υπάρχει ανάγκη αλλαγής του περιεχομένου της τακτικά. Είναι χρήσιμες για μικρές επιχειρήσεις που θέλουν να έχουν έναν ιστότοπο για να κάνουν μία απλή παρουσίαση των προϊόντων τους στο Διαδίκτυο.

2.2.2.3 Δυναμικές Ιστοσελίδες

Οι δυναμικές ιστοσελίδες έναντι των στατικών, προσαρμόζουν το περιεχόμενο τους και την εμφάνισή τους όποια στιγμή θέλουμε. Ο διαχειριστής μπορεί εκείνη την στιγμή να κάνει οποιαδήποτε προσθήκη, αφαίρεση, αλλαγή και διόρθωση χρειάζεται, διότι οι δυναμικές ιστοσελίδες είναι εφαρμογές-προγράμματα και όχι απλά HTML έγγραφα όπως είναι οι στατικές.

Για να διαβαστούν τα περιεχόμενα μία δυναμικής ιστοσελίδας χρειάζεται κάτι παραπάνω από έναν φυλλομετρητή όπως είναι στις στατικές. Τα περιεχόμενα είναι γραμμένα σε γλώσσες προγραμματισμού, με τις πιο συχνές όπως είναι η PHP, ASP .NET και άλλες. Για να μεταφραστούν και να εμφανιστούν στις οθόνες μας μέσω των φυλλομετρητών, τα περιεχόμενα τους μεταφράζονται στους Servers και από εκεί αποστέλλονται μέσω του διακομιστή Apache¹⁰. Βασικό όμως είναι, ότι πίσω από όλες τις δυναμικές ιστοσελίδες

⁹ Search Engine Optimization, Wikipedia, https://en.wikipedia.org/wiki/Search_engine_optimization & Intro to Search Engine Optimization, Searchenginewatch.com, 2017, <https://searchenginewatch.com/sew/news/2048891/intro-search-engine-optimization>

¹⁰ Apache HTTP Server, 2015, <http://httpd.apache.org/>

κρύβονται Βάσεις Δεδομένων (Databases), όπου εκεί αποθηκεύονται όλα τα περιεχόμενα και οι πληροφορίες των ιστοτόπων. Άρα για να μπορέσει ένας χρήστης να δει κάποιο περιεχόμενο, πρέπει πρώτα ο ιστότοπος να επικοινωνήσει με την βάση δεδομένων και να αντλήσει από εκεί τις πληροφορίες που χρειάζονται.

Εξαιτίας των βάσεων δεδομένων που χρησιμοποιούνται, είναι πολύ εύκολο για κάποιον που δεν γνωρίζει από γλώσσες προγραμματισμού να μπορέσει να προσθέσει ή να αφαιρέσει κάποια περιεχόμενα από την ιστοσελίδα του, γιατί το μόνο που χρειάζεται είναι να έχει τη γνώση να διαχειρίζεται το περιεχόμενα μίας βάσης δεδομένων. Όλα τα υπόλοιπα γίνονται αυτοματοποιημένα από το πρόγραμμα.

Για την δημιουργία ενός δυναμικού ιστοτόπου χρειάζονται πολλές γραμμές κώδικα προγραμματισμού κάτι που το κάνει δύσκολο στην κατασκευή, όμως είναι εύκολο αργότερα στην διαχείριση και στην ανανέωση των περιεχομένων του. Εδώ έχουμε και το μεγάλο τους πλεονέκτημα που είναι η χρήση των CMS¹¹. Για να μπορεί ένας απλός χρήστης να διαχειρίζεται τα περιεχόμενα μίας δυναμικής ιστοσελίδας εύκολα και χωρίς να χρειάζονται γνώσεις προγραμματισμού υπάρχουν τα Content Management System ή αλλιώς CMS τα οποία είναι μηχανισμοί διαχείρισης περιεχομένων. Τα CMS έχουν μία περιοχή διαχείρισης της σελίδας, όπου ο διαχειριστής μπορεί να διαχειρίζεται όλα τα περιεχόμενα του ιστοτόπου του.

Τα θετικά μίας δυναμικής ιστοσελίδας είναι τα παρακάτω:

- Δυναμική παρουσίαση στο Διαδίκτυο
- Συχνή ανανέωση περιεχομένου και καθόλου περιορισμός στις σελίδες που μπορεί να έχει κάποιος
- Δεν χρειάζονται προγραμματιστικές ικανότητες
- Εύκολη διαχείριση χρηστών, διαχειριστών και μελών μίας ιστοσελίδας

Τα αρνητικά είναι:

- Υψηλό κόστος κατασκευής και συντήρησης της
- Απαιτείται μεγάλος χώρος σε φιλοξενία σε κάποιον Web Server
- Αναπτύσσεται και σχεδιάζεται πιο αργά από μία στατική ιστοσελίδα
- Η σελίδα «φορτώνει» πιο αργά λόγω του περιεχομένου της

Οι δυναμικές ιστοσελίδες είναι χρήσιμες για επιχειρήσεις που θέλουν να κατασκευάσουν ένα ηλεκτρονικό κατάστημα, όπου χρειάζεται η συχνή ανανέωση των περιεχομένων της. Όπως επίσης και για ξενοδοχεία που θέλουν οι πελάτες να μπορούν να κάνουν ηλεκτρονικές κρατήσεις. Και επίσης, όλα οι ειδησεογραφικοί ιστότοποι όπου χρειάζονται συχνά να ανανεώνουν το περιεχόμενό τους.

2.2.3 Γλώσσες σήμανσης

¹¹ Content Management System, Wikipedia, https://en.wikipedia.org/wiki/Content_management_system

Οι γλώσσες σήμανσης ή αλλιώς markup languages¹², έχουν σχεδιαστεί για την επεξεργασία, τον ορισμό και την παρουσίαση ενός κειμένου. Η γλώσσα αυτή καθορίζει τον κώδικα μορφοποίησης, τόσο για την διάταξη όσο και για την εμφάνισή του, μέσα σε ένα αρχείο κειμένου. Ο κώδικας που χρησιμοποιείται για τον ορισμό της μορφοποίησης του κειμένου λέγονται «ετικέτες». Η γλώσσα HTML είναι ένα παράδειγμα μιας ευρέως γνωστής και χρησιμοποιούμενης γλώσσας σήμανσης.

Η σήμανση, δηλαδή η ετικέτα που υπάρχει στο κείμενο, δίνει εντολή στο πρόγραμμα που εμφανίζει το κείμενο να κάνει τις ανάλογες ενέργειες βάση αυτών, με την διαφορά ότι αυτές οι ετικέτες δεν εμφανίζονται στον απλό χρήστη. Οι γλώσσες σήμανσης δεν έχουν αυστηρή δομή όπως έχουν άλλες γλώσσες προγραμματισμού, άρα για να εκτελεστούν δεν χρειάζονται κάποιον compiler και χρησιμοποιούνται μόνο για να περιγράψουν πως θα γίνει μια εργασία, παρά πως να γίνει η εκτέλεσή της.

Επίσης χωρίζονται σε τρεις κατηγορίες:

- Παρουσιαστική σήμανση (Presentational Markup)
- Διαδικαστική σήμανση (Procedural Markup)
- Περιγραφική σήμανση (Descriptive Markup).

2.2.4 SGML (Standard Generalized Markup Language)

Η SGML (Standard Generalized Markup Language)¹³, αποτελεί πλέον το διεθνές πρότυπο της γενικευμένης γλώσσας σήμανσης για τα έγγραφα. Η SGML έγινε πρότυπο ISO 8879 το 1986 ως τρόπος έκφρασης δεδομένων σε εφαρμογές επεξεργασίας κειμένων. Η SGML στην ουσία είναι ένα σύνολο χαρακτήρων που επιτρέπει την χρήση οντοτήτων (αντικείμενα). Η SGML ως γλώσσα σήμανσης δίνει έμφαση και περιγράφει τη δομή του κειμένου και όχι την μορφοποίησή του. Οι σημάνσεις που χρησιμοποιούνται στο κείμενο επίσης πρέπει να συμφωνούν με ένα μοντέλο σημάνσεων το DTD, που ορίζεται στην αρχή του SGML κειμένου. Η SGML ως μέτα-γλώσσα, αποτελεί μία γενικευμένη γλώσσα σήμανσης μέσα από την οποία μπορούν να προκύψουν άλλες ειδικές γλώσσες (όπως για παράδειγμα η HTML).

2.2.5 HTML (HyperText Markup Language)

Τι είναι η HTML

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language, είναι μία γλώσσα σήμανσης μορφοποίησης υπερκειμένου και είναι η βασική γλώσσα δόμησης ιστοσελίδων στον Παγκόσμιο Ιστό. Τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων που σημαίνει ότι γράφεται πρώτα το κείμενο και έπειτα γύρω από τις λέξεις ή

¹² Markup language, Wikipedia, https://en.wikipedia.org/wiki/Markup_language

¹³ The SGML Declaration in SGML and HTML Explained, Martin Bryan, 1997, <https://web.archive.org/web/20160306121935/http://www.is-thought.co.uk/book/sgml-4.htm>

από ολόκληρες προτάσεις, προσθέτονται ειδικά σύμβολα ώστε να καθορίζεται η εμφάνισή τους στην οθόνη μας.

Ιστορία

Δημιουργός της HTML γλώσσας είναι ο Tim Berners-Lee, ο ίδιος που δημιούργησε και τον Παγκόσμιο Ιστό (WWW). Το 1980, όταν ο Lee εργαζόταν ακόμα στο CERN, για τις ανάγκες των ερευνητών να χρησιμοποιούν και να διαμοιράζονται έγγραφα, επινόησε το σύστημα ENQUIRE¹⁴. Έτσι κατασκεύασε ένα πρωτότυπό της. Αργότερα, το 1989, πρότεινε ένα νέο σύστημα το οποίο βασίστηκε στο διαδίκτυο και πλέον θα χρησιμοποιούσε υπερκείμενο, φτιάχνοντας έτσι την προδιαγραφή της HTML και αργότερα στα τέλη του 1990, έγραψε τον φυλλομετρητή και το λογισμικό του server. Η δημιουργία της HTML βασίστηκε στην SGML.

Η HTML ορίστηκε και επίσημα από τον οργανισμό IETF¹⁵ ως μια υλοποίηση της SGML. Το 1993, με μία δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, η οποία προδιαγραφή περιελάμβανε και έναν Ορισμό τύπου εγγράφου (DTD) της SGML, ορίζει την σωστή γραμματική του εγγράφου. Ο Ορισμός τύπου εγγράφου (DTD) είναι στην ουσία ένα σύνολο δηλώσεων σήμανσης που ορίζουν έναν τύπο εγγράφου για μία γλώσσα σήμανσης που ανήκει στην οικογένεια της SGML, ορίζει δηλαδή τη δομή του εγγράφου με μια λίστα αποδεκτών στοιχείων και χαρακτηριστικών.

Δομή ενός αρχείου HTML

Τα στοιχεία HTML είναι και τα δομικά στοιχεία των HTML σελίδων. Τα ειδικά σύμβολα στην HTML λέγονται ετικέτες ή tags. Οι ετικέτες γράφονται ανάμεσα στα σύμβολα < ... > και σχεδόν πάντα όλες οι ετικέτες αποτελούνται από μια ετικέτα με αρχή και τέλος και ανάμεσα σε αυτές υπάρχει το κείμενο που θέλουμε να χαρακτηρίζεται από αυτές. Την ετικέτα τέλους την ξεχωρίζουμε από τον χαρακτήρα / πριν το όνομα της ετικέτας.

Για έναν φυλλομετρητή, για να μπορέσει να καταλάβει ότι ένα αρχείο είναι HTML, πρέπει να υπάρχουν μέσα στο έγγραφο κάποιες βασικές ετικέτες. Αυτές είναι οι εξής:

- Ετικέτα html
- Ετικέτα head
- Ετικέτα body

Παράδειγμα:

```
<html>
  <head>
    <title>Αυτό είναι ένα παράδειγμα!</title>
```

¹⁴ Tim Berner-Lee, Wikipedia, https://el.wikipedia.org/wiki/Τιμ_Μπέρνερς_Λι

¹⁵ Internet Engineering Task Force, Τακτική Δύναμη Μηχανικών Internet, <https://el.wikipedia.org/wiki/IETF>

```
</head>
<body>
  <p>Παράδειγμα σωστής δομής ενός html αρχείου.</p>
</body>
</html>
```

Εξήγηση παραδείγματος:

- Με τις ετικέτες `<html>... </html>` ο φυλλομετρητής καταλαβαίνει ότι το έγγραφο μας είναι ένα html έγγραφο. Το κείμενο ανάμεσα στις ετικέτες περιγράφει την ιστοσελίδα.
- Το κείμενο που βρίσκεται ανάμεσα στις ετικέτες `<head>.....</head>` περιέχει τις πληροφορίες για την σελίδα που βρισκόμαστε, όπως γλώσσα συγγραφής, τίτλο σελίδας (αυτός που εμφανίζεται στην καρτέλα ενός φυλλομετρητή), δήλωση κώδικα CSS, JavaScript και άλλα.
- Το κείμενο που βρίσκεται ανάμεσα στις ετικέτες `<body>.....</body>` είναι το ορατό κομμάτι μίας HTML σελίδας. Ότι θέλουμε να εμφανίζεται στην ιστοσελίδα μας το περιλαμβάνουμε ανάμεσα σε αυτές ετικέτες.
- Το κείμενο που βρίσκεται ανάμεσα στις ετικέτες `<title>.....</title>` είναι ο τίτλος που αναφερθήκαμε παραπάνω, δηλαδή είναι ο τίτλος που εμφανίζεται στην ανοιχτή καρτέλα ενός φυλλομετρητή.

2.2.6 CSS

Εισαγωγή

Το CSS σημαίνει Cascading Style Sheets¹⁶ ή αλλιώς Διαδοχικά Φύλλα Στυλ είναι μία γλώσσα stylesheet και χρησιμοποιείται για έλεγχο του τρόπου εμφάνισης ενός εγγράφου που είναι γραμμένο συνήθως στις γλώσσες σήμανσης HTML και XHTML. Γενικότερα δηλαδή, για τον τρόπο εμφάνισης μίας ιστοσελίδας ή ενός ολόκληρου ιστοτόπου. Με πιο απλά λόγια το CSS είναι από τις πιο απλές και εύκολες γλώσσες και βοηθάνε να ορίσουμε με ακρίβεια και ιδιαίτερη ευελιξία τους τρόπους εμφάνισης των διάφορων στοιχείων μίας ιστοσελίδας.

Το CSS είναι μία γλώσσα στυλ και μπορεί να αναφέρεται είτε σε μία παράγραφο, είτε σε έναν χαρακτήρα, σε ένα πίνακα και γενικά σε ένα οποιοδήποτε στοιχείο ενός HTML. Μπορούμε να ορίσουμε μέγεθος γραμματοσειρών, χρώματα, περιθώρια, εικόνες φόντου και πολλά άλλα. Το CSS έρχεται για να καλύψει την αδυναμία της HTML για την μορφοποίηση και την διάταξή του. Το CSS υποστηρίζονται από όλους τους φυλλομετρητές.

Διαφορές CSS και HTML

Η διαφορά ανάμεσα στο CSS και την HTML είναι ότι η HTML χρησιμοποιείται για την δόμηση του περιεχομένου ενώ το CSS χρησιμοποιείται για τη διαμόρφωση και μορφοποίηση του δομημένου περιεχομένου. Η HTML χρησιμοποιείται μόνο για την δομή

¹⁶ CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ), Wikipedia, <https://el.wikipedia.org/wiki/CSS>

του κειμένου και πιο συγκεκριμένα για να επισημαίνουμε ποια είναι η κεφαλίδα, η παράγραφος, χρησιμοποιώντας τις γνωστές ετικέτες της HTML `<h1>...</h1>` ή `<p>.....</p>`. Ενώ με το CSS μπορούμε να καθορίσουμε ότι το h1 θέλουμε να έχουν 20pt στην γραμματοσειρά και να είναι εντονότερα από την κανονική γραμματοσειρά ή ότι η παράγραφος θέλουμε να εμφανίζεται με ένα γκρι περίγραμμα.

Σύνταξη CSS

Μπορούμε να ορίσουμε «στυλ» για:

- Συγκεκριμένες ετικέτες ενός αρχείου
- Ετικέτες με την ιδιότητα Class, δηλαδή τα στυλ θα εφαρμοστούν μόνο στις ετικέτες με το συγκεκριμένο class.
- Ετικέτες με την ιδιότητα id, δηλαδή τα στυλ θα εφαρμοστούν μόνο στις ετικέτες με το συγκεκριμένο id.

Η γενική σύνταξη ενός στυλ ενός CSS στοιχείου αποτελείται από τρία στοιχεία:

1. Τον Επιλογέα ή αλλιώς Selector
2. Τις ιδιότητες του επιλογέα ή αλλιώς Attributes
3. Και τις τιμές των ιδιοτήτων του επιλογέα ή αλλιώς values

Παράδειγμα σύνταξης μίας επικεφαλίδας:

```
h1 (selector) {  
    font (attribute): 15pt (value);  
    color (attribute): blue (value);  
}
```

Πλεονεκτήματα

Τα πλεονεκτήματα του CSS είναι πολλά και μερικά από αυτά είναι τα εξής:

- Μεγαλύτερη ευελιξία μορφοποίησης. Το CSS μπορεί να εφαρμόσει μορφοποιήσεις οι οποίες είναι αδύνατες ή πολύ δύσκολες να γίνουν με την HTML.
- Εύκολη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου ιστοτόπου μπορεί να βασίζεται σε ένα μόνο αρχείο CSS. Αυτό σημαίνει ότι όταν θέλουμε να αλλάξουμε για παράδειγμα την γραμματοσειρά σε όλες τις κεφαλίδες του ιστοτόπου, αρκεί να ανοίξουμε και επεξεργαστούμε μόνο το συγκεκριμένο κομμάτι στο CSS και όχι ένα σημείο όπως θα χρειαζόταν στην HTML.
- Πολύ μικρότερο μέγεθος αρχείου, διότι ο κάθε κανόνας γράφεται μόνο μία φορά και όχι σε κάθε σημείο που θα χρειαζόταν στην HTML.
- Καλύτερο SEO (Search Engine Optimization). Όπως αναφερθήκαμε και παραπάνω το καλύτερο SEO σημαίνει μεγαλύτερη επισκεψιμότητα, άρα και μεγαλύτερο κέρδος για έναν ιστότοπο. Καταφέρνουμε καλύτερο SEO διότι οι μηχανές αναζήτησης δεν μπερδεύονται ανάμεσα σε περιεχόμενο και την μορφοποίηση του, βρίσκουν ένα

καθαρό κείμενο, οπότε είναι πιο εύκολο να το καταγράψουν και να το αρχειοθετήσουν.

- Γρήγορο φόρτωμα σελίδας. Με την χρήση ενός εξωτερικού αρχείου CSS, ο φυλλομετρητής την πρώτη φορά που θα ανοίξει μία ιστοσελίδα, αποθηκεύει το αρχείο αυτό στην προσωρινή του μνήμη (cache), οπότε δεν χρειάζεται να το κατεβάξει κάθε φορά που ο χρήστης μπαίνει σε άλλη σελίδα στον ίδιο ιστότοπο.¹⁷

2.2.7 JavaScript

Εισαγωγή

Η JavaScript ή JS¹⁸ σε συντομογραφία, είναι μία ελαφριά αντικειμενοστραφής γλώσσας προγραμματισμού, που δεν χρειάζεται μεταγλώττιση, δηλαδή compiler όπως και η PHP. Αρχικά δημιουργήθηκε για τον Παγκόσμιο Ιστό, ώστε τα σενάρια από την μεριά του πελάτη να μπορούν να επικοινωνούν με τον χρήστη. Να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που βρίσκονται εκείνη την στιγμή.

Είναι κυρίως γνωστή ως γλώσσα σεναρίου (Scripting Language) για ιστοσελίδες. Είναι δυναμική και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Ως γλώσσα προγραμματισμού είναι επηρεασμένη και αυτή από την γλώσσα προγραμματισμού C. Οι βασικές αρχές σχεδιασμού της JS προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme και είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm)

Ιστορία της JS

Δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία αρχικά Mocha, έπειτα μετονομάστηκε σε Live Script όπου αργότερα πήρε την τελική της ονομασία JavaScript, επειδή η ανάπτυξη της προήλθε κυρίως από την γλώσσα προγραμματισμού Java. Ως Live Script κυκλοφόρησε τον Σεπτέμβριο του 1995, μαζί με τον Netscape Navigator 2.0 και ως JavaScript μετονομάστηκε στις 4 Δεκεμβρίου 1995 όταν επεκτάθηκε και ο Netscape Navigator σε έκδοση 2.0B3.

Η JavaScript ως γλώσσα που εκτελούνταν από την μεριά του πελάτη σε ιστοσελίδες, απέκτησε πολύ μεγάλη επιτυχία και αυτός ήταν και ο λόγος που την συμπεριέλαβαν σε διάφορους φυλλομετρητές. Μέχρι και η Microsoft ονόμασε την εφαρμογή της σε JScript, για να αποφύγει διενέξεις λόγω εμπορικών σημάτων και πρόσθεσε νέες μεθόδους για να διορθώσει τα Y2K-προβλήματα που είχε τότε η JavaScript. Η Microsoft έπειτα συμπεριέλαβε την Jscript στον φυλλομετρητή Internet Explorer 3.0, ο οποίος κυκλοφόρησε τον Αύγουστο του 1996.

Όμως τον Νοέμβριο του 1996, η δημιουργός του JavaScript ανακοίνωσε ότι είχε υποβάλει την γλώσσα JavaScript στο Ecma International, μία οργάνωση που ασχολούταν με την τυποποίηση των γλωσσών προγραμματισμού, για να την εξετάσει ως βιομηχανικό πρότυπο, πράγμα που κατάφερε και είχε ως αποτέλεσμα να συμπεριληφθεί ως ECMAScript.

¹⁷ Εισαγωγή στο CSS, Μιχαήλια Κομβούτη- Βέρου, http://pages.cs.aueb.gr/courses/ep1131/files/CSS_notes.pdf

¹⁸ JavaScript, Wikipedia, <https://el.wikipedia.org/wiki/JavaScript>

Τον Ιανουάριου του 2009, η δημιουργία του έργου Common JS είχε ως στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης για να μπορέσει η JavaScript να αναπτυχθεί και έξω από τους φυλλομετρητές, μέσα σε άλλες τεχνολογίες.

Χρήση της JavaScript σε HTML σελίδα

Με το ότι η JavaScript είναι και η αυτή μία scripting γλώσσα η οποία έχει σχεδιαστεί και χρησιμοποιείται κυρίως για να εισάγουμε διαδραστικότητα στις HTML σελίδες, μπορούμε να κάνουμε πλέον τα εξής:

- Μας δίνεται η δυνατότητα εκτέλεσης κάποιων πραγμάτων όταν συμβαίνει ένα γεγονός, για παράδειγμα όταν ο χρήστης πατήσει πάνω σε ένα HTML στοιχείο, όπως είναι ένα κουμπί, αυτόματα, εκτελείται κάποιο script το οποίο εμφανίζει κάποιο αποτέλεσμα εκείνη την στιγμή.
- Με την χρήση της JavaScript μπορούμε να διαβάσουμε και να αλλάξουμε τα περιεχόμενα ενός συγκεκριμένου HTML στοιχείου.
- Μπορεί να χρησιμοποιηθεί για να επικυρωθούν τα δεδομένα μίας HTML φόρμας, πριν αυτά αποσταλούν στον Server.
- Με την JavaScript μπορούμε να εντοπίσουμε τον φυλλομετρητή που χρησιμοποιεί ο χρήστης για να περιηγηθεί στην ιστοσελίδα και ανάλογα με τον φυλλομετρητή, να φορτώνει την αντίστοιχη ιστοσελίδα που είναι φτιαγμένη γι' αυτόν.
- Και τέλος, ένα ακόμα σημαντικό που μας δίνει την δυνατότητα να κάνουμε η γλώσσα JavaScript, είναι να δημιουργήσουμε Cookies, που βοηθάνε να λαμβάνουμε πληροφορίες από τον υπολογιστή του επισκέπτη της ιστοσελίδας.

Σύνταξη

Για να εισάγουμε ένα JavaScript script σε ένα HTML αρχείο, όπως και στην PHP, χρησιμοποιήσουμε την ετικέτα `<script>.... </script>` και μέσα σε αυτή την ετικέτα χρησιμοποιήσουμε το όρισμα `type` για να δηλώσουμε την scripting γλώσσα που θα χρησιμοποιήσουμε.

Με την ετικέτα `<script>` δηλώνουμε που αρχίζει η JavaScript και με την ετικέτα `</script>` που τελειώνει.

Παράδειγμα JavaScript μέσα σε HTML:

```
<html>
  <head>
    <title>Αυτό είναι ένα παράδειγμα!</title>
  </head>
  <body>
    <p>Παράδειγμα σωστής δομής ενός html αρχείου.</p>

    <script type="text/javascript">
```

```
        alert('Είμαι ένα JavaScript!');
    </script>
</body>
</html>
```

Στο συγκεκριμένο παράδειγμα στην σελίδα με την εκτέλεση του JavaScript θα εμφανίσει αυτό εδώ στην οθόνη μας:

*Παράδειγμα σωστής δομής ενός html αρχείου.
Είμαι ένα JavaScript!*

2.2.8 AJAX

Εισαγωγή

Τα αρχικά AJAX σημαίνουν «Asynchronous JavaScript και XML»¹⁹. Από τα αρχικά του ονόματός του καταλαβαίνουμε ότι η AJAX προέρχεται από διάφορες τεχνολογίες οι οποίες χρησιμοποιούνται κυρίως για την ανάπτυξη ιστοσελίδων. Στην ουσία η AJAX είναι ένα σύνολο τεχνικών ανάπτυξης ασύγχρονων εφαρμογών, καθώς με την AJAX, οι διαδικτυακές εφαρμογές μπορούν να στέλνουν και να λαμβάνουν δεδομένα από ένα server, ασυγχρόνιστα. Δηλαδή πίσω από το παρασκήνιο χωρίς να γίνεται αντιληπτή, χωρίς επίσης να παρεμβαίνει στον τρόπο εμφάνισης της ιστοσελίδας ή στον τρόπο λειτουργίας της.

Ιστορία της AJAX

Στις αρχές τις δεκαετίας του '90, οι περισσότεροι ιστότοποι ήταν βασισμένοι εξολοκλήρου στις HTML σελίδες. Κάθε ενέργεια που έκανε ο χρήστης στον ιστότοπο, απαιτούσε την φόρτωση μίας νέας σελίδας από τον server από την αρχή.

Η αλλαγή έγινε το 1996 με την εισαγωγή της iframe ετικέτας από τον Internet Explorer. Όπως το <object>στοιχείο, το οποίο χρησιμοποιείται για την ενσωμάτωση πολυμέσων ή μίας άλλης ολόκληρης HTML ιστοσελίδας, μπορεί να φορτώσει ή να προσεγγίσει το περιεχόμενο ασύγχρονα. Το 1998, πάλι η ομάδα της Microsoft και συγκεκριμένα του Microsoft Outlook, ανέπτυξε την ιδέα πίσω από το XMLHttpRequest αντικείμενο script. Εμφανίστηκε ως XMLHttpRequest στην δεύτερη έκδοση της βιβλιοθήκης MSXML η οποία όμως απεστάλη από τον Internet Explorer 5.0 τον Μάρτιο του 1999.

Η λειτουργικότητα του στοιχείου ελέγχου XMLHttpRequest στον Internet Explorer 5.0 υλοποιήθηκε αργότερα από τους φυλλομετρητές όπως τον Mozilla, Safari, Opera και άλλους φυλλομετρητές ως το XMLHttpRequest αντικείμενο της JavaScript. Η Microsoft ξανά υιοθέτησε το γνήσιο πρότυπο του XMLHttpRequest στον Internet Explorer 7.

Αργότερα, η Google, πραγματοποίησε ευρεία ανάπτυξη της τεχνολογίας AJAX, αφού την χρησιμοποίησε στο Gmail το 2004 και στους Χάρτες της Google το 2005. Τον Οκτώβριο του 2004, είχαμε δημόσια beta έκδοση του Kayak.com, το οποίο ήταν από τις πρώτες μεγάλες χρήσεις του ηλεκτρονικού εμπορίου για ότι οι προγραμματιστές τους εκείνη την εποχή

¹⁹ Ajax (Programming), Wikipedia, [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))

αποκαλούσαν «το xml html πράγμα». Αυτό εκτόξευσε το ενδιαφέρον των προγραμματιστών για την AJAX.

Ο όρος AJAX χρησιμοποιήθηκε δημοσίως στις 18 Φεβρουαρίου 2005 από το Jesse James Garrett σε ένα άρθρο με τίτλο «AJAX: Μία νέα προσέγγιση στις εφαρμογές Διαδικτύου, με βάση τις τεχνικές που χρησιμοποιούνται στις σελίδες Google».

Στις 5 Απριλίου 2006, η W3C, κυκλοφόρησε το πρώτο σχέδιο προδιαγραφής για το αντικείμενο XMLHttpRequest σε μία προσπάθεια της να δημιουργήσει ένα επίσημο πρότυπο Διαδικτύου. Το τελευταίο προσχέδιο του XMLHttpRequest αντικείμενο δημοσιεύθηκε στις 30 Ιανουαρίου του 2014.

Οι τεχνολογίες από τις οποίες προέρχεται η AJAX

Οι τεχνολογίες από τις οποίες προέρχεται η AJAX είναι οι εξής:

- JavaScript
- XML και XSLT
- (X)HTML και CSS
- Το XMLHttpRequest αντικείμενο
- Document Object Model (DOM)

Όλα αυτά συνδυάζονται με την τεχνολογία AJAX καθώς με το (object) XMLHttpRequest, πραγματοποιείται η ασύγχρονη επικοινωνία με τον server. Έπειτα με την JavaScript και DOM, έχουμε την αλληλεπίδραση με τα δεδομένα και την εμφάνιση τους στην οθόνη, την CSS η οποία μορφοποιεί τα δεδομένα που είναι για εμφάνιση και την XML που χρησιμοποιείται συχνά πλέον για την μεταφορά των δεδομένων.

Πλεονεκτήματα της AJAX

Υπάρχουν 4 βασικά πλεονεκτήματα της χρήσης της τεχνολογίας AJAX στις διαδικτυακές εφαρμογές τα οποία είναι:

1. *Callback (Επανάκληση)*: Το AJAX χρησιμοποιείται για να πραγματοποιήσει μια επανάκληση, κάνοντας έναν γρήγορο κύκλο προς και από τον server για να ανακτήσει και να αποθηκεύσει δεδομένα χωρίς να χρειάζεται να στείλει ολόκληρη ξανά την σελίδα πίσω στον server. Με το να μην χρειάζεται να επιστρέψει όλη την σελίδα και να κάνει αποστολή όλων των δεδομένων μιας φόρμας στον server, η χρήση του διαδικτύου ελαχιστοποιείται και έτσι έχουμε ταχύτερες λειτουργίες. Σε ιστοτόπους και σε τοποθεσίες με περιορισμένο εύρος ζώνης, αυτό μπορεί να βελτιώσει σημαντικά την απόδοση του δικτύου. Τις περισσότερες φορές, τα δεδομένα που αποστέλλονται προς και από τον server είναι ελάχιστα και με την χρήση της επανάκλησης, ο server δεν χρειάζεται να επεξεργαστεί όλα τα στοιχεία μιας φόρμας, οπότε με την αποστολή μόνο των απαραίτητων δεδομένων, υπάρχει περιορισμένη επεξεργασία στον server.
2. *Πραγματοποίηση ασύγχρονων κλήσεων*: Η AJAX μας επιτρέπει να πραγματοποιήσουμε ασύγχρονες κλήσεις σε έναν διαδικτυακό server. Αυτό σημαίνει

ότι επιτρέπει στον φυλλομετρητή του χρήστη να αποφύγει την αναμονή μέχρι να φτάσουν όλα τα δεδομένα, με τον χρήστη να μπορεί να κάνει και κάτι άλλο.

3. *User-Friendly (Φιλικό προς τον χρήστη)*: Επειδή δεν χρειάζεται να γίνει πλέον ανανέωση της σελίδας, οι διαδικτυακές εφαρμογές που δημιουργούνται με την AJAX θα είναι πάντα πιο ανταποκριτικές σε όλες τις συσκευές, πιο γρήγορες και πιο φιλικές προς τον χρήστη.
4. *Αύξηση της ταχύτητας*: Ο κύριος σκοπός της AJAX είναι η αύξηση της ταχύτητας, της απόδοσης και της χρηστικότητας μιας εφαρμογής στο διαδίκτυο. Ένα εξαιρετικό παράδειγμα της AJAX είναι η βαθμολόγηση ταινιών στο imdb όπου θα αποθηκευτεί στην βάση δεδομένων τους χωρίς να χρειάζεται να γίνει η ανανέωση της σελίδας²⁰.

Χρήσεις της AJAX

Το AJAX θα έπρεπε να χρησιμοποιείται σε οποιαδήποτε διαδικτυακή εφαρμογή όπου μικρές ποσότητες πληροφοριών μπορούν να αποθηκευτούν ή να ανακτηθούν από τον server χωρίς να φορτώνονται οι σελίδες από την αρχή. Ένα καλό παράδειγμα αυτού είναι η επικύρωση των δεδομένων σε φόρμες που έχουν κουμπιά αποθήκευσης. Άλλο ένα παράδειγμα θα ήταν η αλλαγή των δεδομένων ενός πτυσσόμενου μενού λίστας ανάλογα με άλλα στοιχεία που έχουν εισαχθεί, όπως το πτυσσόμενο μενού για τις περιοχές ενός νομού, όταν ο χρήστης εισάγει ότι κατοικεί στο Ηράκλειο Κρήτης στο μενού να του εμφανίζει τις αντίστοιχες περιοχές.

Επίσης, άλλη μία χρήση της AJAX είναι όταν ο χρήστης χρειάζεται να αποθηκεύσει ή να ανακτήσει session values, δηλαδή τιμές συνεδρίας, από τον server, ανάλογα με τις προτιμήσεις του χρήστη όπως είναι το ύψος, πλάτος ή την θέση ενός αντικειμένου. Η προσαρμογή του πλάτους θα μπορούσε να κάνει μία επανάκληση στον server για να ορίσει την τιμή της συνεδρίας για το νέο πλάτος. Με αυτό τον τρόπο, κάθε φορά που η σελίδα θα ανανεώνεται, ο server θα μπορεί να προσαρμόσει το πλάτος του αντικειμένου βάση της μεταβλητής της συνεδρίας. Διαφορετικά, το αντικείμενο θα επανέλθει στο προκαθορισμένο του πλάτος.

Άλλα χαρακτηριστικά επίσης είναι οι υποδείξεις κειμένων και τα πλαίσια κειμένου αυτόματης συμπλήρωσης. Ο χρήστης κατά την πληκτρολόγηση πάνω από δύο γραμμάτων, τότε μία λίστα εμφανίζεται με λέξεις που ξεκινάνε με αυτά τα γράμματα. Μία επανάκληση γίνεται στον server διαδικτύου με σκοπό την ανάκτηση όλων των τιμών που ξεκινάνε με αυτούς τους χαρακτήρες.

2.2.9 Βάσεις Δεδομένων

Εισαγωγή

²⁰ What is Ajax and Where is it Used in Technology?, Seque Technologies, <https://www.seguetech.com/ajax-technology/>

Μία βάση δεδομένων είναι μια οργανωμένη συλλογή δεδομένων²¹. Μία σχεσιακή βάση δεδομένων, πιο περιοριστικά, είναι μια συλλογή σχημάτων, πινάκων, ερωτημάτων, αναφορών, προβολών και άλλων στοιχείων. Όπως και να έχει όμως, μία βάση δεδομένων λειτουργεί έτσι ώστε να μπορεί να εξυπηρετεί αποτελεσματικά πολλές διαδικτυακές εφαρμογές, μειώνοντας τις άσκοπες επαναλήψεις των δεδομένων.

Με μία βάση δεδομένων όλα τα δεδομένα καταχωρούνται με τέτοια μέθοδο, ώστε να παρουσιάζονται με ενιαίο τρόπο στον χρήστη. Κάθε εφαρμογή μπορεί να κάνει χρήση των δεδομένων που της είναι απαραίτητα για την αντίστοιχη επεξεργασία ή εμφάνιση των πληροφοριών που χρειάζεται.

Ιστορία

Η ιστορία των βάσεων δεδομένων ξεκινάει την εποχή που πρωτοεμφανίστηκαν και οι υπολογιστές.

Στα τέλη της δεκαετίας του '60, τότε λεγόταν «τράπεζα δεδομένων» ή αλλιώς «data banks». Εκείνη την εποχή έχουμε και την δημιουργία των διαδικτυακών βάσεων δεδομένων (network databases) οι οποίες τότε ονομαζόντουσαν Βάσεις Δεδομένων CODASYL, από το όνομα της επιτροπής που την δημιούργησαν και του IMS, του πιο γνωστού συστήματος διαχείρισης ιεραρχικών βάσεων δεδομένων (Hierarchical DBMS). Το IMS, ανήκει στην IBM, που είναι μέχρι και σήμερα ένα από τα συστήματα με τις περισσότερες εγκαταστάσεις.

Στην δεκαετία του 1970, έρχεται επίσης ακόμα μια αλλαγή. Τα τότε υπάρχοντα συστήματα δεν ήταν εύκολο να χρησιμοποιηθούν. Έτσι, ο Edgar Codd, σκέφτηκε έναν τρόπο να κάνει τα πράγματα πιο εύκολα. Σκέφτηκε το σχεσιακό μοντέλο, το οποίο δεν είχε καμία σχέση με την παραδοσιακή βάση δεδομένων, καθώς επέμενε ότι οι εφαρμογές θα έπρεπε να αναζητούν δεδομένα κατά το περιεχόμενο, παρά από την ακολουθία συνδέσμων. Το σχεσιακό μοντέλο χρησιμοποιεί το καθολικό στυλ πινάκων, που ο καθένας χρησιμοποιείται για διαφορετικό τύπο οντότητας.

Στα μέσα της δεκαετίας του 1990, ωστόσο, τα σχεσιακά συστήματα κυριάρχησαν σε όλες τις εφαρμογές με μεγάλη κλίμακα επεξεργασίας δεδομένων. Και μέχρι σήμερα, το 2018, παραμένουν κυρίαρχες, όπως είναι IBM DB2, Oracle, MySQL και Microsoft SQL Server, ως τα κορυφαία συστήματα διαχείρισης βάσεων δεδομένων.

Η κυρίαρχη γλώσσα της βάσης δεδομένων, η τυποποιημένη SQL για το σχεσιακό μοντέλο, έχει επηρεάσει τις γλώσσες των βάσεων δεδομένων για άλλα μοντέλα δεδομένων²².

Πλεονεκτήματα Βάσεων Δεδομένων

- Ελαχιστοποίηση των δεδομένων που δεν χρειάζονται.
- Καλύτερη οργάνωση δεδομένων, ώστε οι απαντήσεις σε αναζητήσεις που γίνονται στις βάσεις δεδομένων να ανταποκρίνονται γρηγορότερα

²¹ Βάση Δεδομένων, Wikipedia, [https://el.wikipedia.org/wiki/ Βάση_δεδομένων & Database](https://el.wikipedia.org/wiki/Βάση_δεδομένων_&_Database), Wikipedia, <https://en.wikipedia.org/wiki/Database>

²² History of Database Management Systems, Bright Hub, 2017, <http://www.brighthub.com/internet/web-development/articles/109527.aspx>

- Λόγω του κεντρικού ελέγχου, έχουμε ασφάλεια των δεδομένων
- Σωστή καταχώρηση των δεδομένων χωρίς λάθη
- Κοινή χρήση δεδομένων από πολλές εφαρμογές

Μειονεκτήματα Βάσεων Δεδομένων

- Πολυέξοδη προμήθεια, οργάνωση και συντήρηση, λόγω της πολυπλοκότητας του λογισμικού
- Απαιτήσεις σε υλικό. Με κάθε αναβάθμιση των συστημάτων διαχείρισης βάσεων δεδομένων, οι απαιτήσεις τους αυξάνονται και μπορεί να χρειάζεται τακτικά οι αναβαθμίσεις των Ηλεκτρονικών Υπολογιστών και των συσκευών του.
- Λόγω της συγκέντρωσης όλων των δεδομένων σε ένα σημείο, υπάρχει ο κίνδυνος της υποκλοπής και γενικά της πρόσβαση στις πληροφορίες από μη εξουσιοδοτημένα άτομα.

2.2.9.1 Δικτυοκεντρικά Συστήματα Βάσεων Δεδομένων

Την σημερινή εποχή, διαπιστώνουμε ότι υπάρχει ολοένα και περισσότερο ενδιαφέρον για προσθήκη περισσότερης πληροφορίας σε ψηφιακή μορφή. Μεγάλο ρόλο σε αυτό, συμβάλει η ραγδαία αύξηση του ποσοστού πληροφορίας. Τεχνολογίες όπως η XML, χρησιμεύουν όχι μόνο στο να συγκεντρωθούν και να μαζευτούν πληροφορίες, αλλά και στο να δοθούν λύσεις στην οργάνωση και τη διαχείριση της πληροφορίας. Επίσης, τεχνολογίες όπως είναι οι DSL, εξασφαλίζουν γρηγορότερη πρόσβαση στο διαδίκτυο, ικανοποιώντας εκατομμύρια χρήστες ανά τον κόσμο. Στο πρόσφατο παρελθόν μας, μπήκαν στην καθημερινότητα μας, καινοτόμες τεχνολογίες όπως είναι το Bluetooth, το ασύρματο internet Wi-Fi με αποτέλεσμα οι χρήστες ολοένα να αυξάνονται. Κατά συνέπεια, στις μέρες μας, σημαντικό κομμάτι της τεχνολογίας, είναι οι συνεχής έρευνες για την αντιμετώπιση προβλημάτων σε θέματα ασφαλείας λόγω κινδύνων που βρίσκονται πίσω από τα ασύρματα δίκτυα που έχουν εισχωρήσει πλέον στην πρόγραμμα της καθημερινότητα μας.

2.2.9.2 Συστήματα Διαχείρισης Βάσεων Δεδομένων

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων ή DBMS (Data Base Management Systems²³), είναι προγράμματα μέσω των οποίων γίνεται η δημιουργία, η διαχείριση, η συντήρηση και η χρήση μιας ηλεκτρονικής βάσης δεδομένων, τα οποία προγράμματα όμως ακολουθούν μία αρκετά πολύπλοκη προσέγγιση για την αποθήκευση και την ανάκτηση των δεδομένων σε σχέση με τα απλά συστήματα επεξεργασίας αρχείων.

Ένα από τα πολλά πλεονεκτήματα των DBMS, είναι ότι μπορεί να προσπελάσει τα δεδομένα χωρίς τη μεσολάβηση του χρήστη, δηλαδή δεν χρειάζεται να γνωρίζει που είναι αποθηκευμένο και τη μορφή των δεδομένων, καθώς μέσω του προγράμματος ζητά να βρεθούν τα δεδομένα και το σύστημα αναλαμβάνει να τα ανακτήσει για λογαριασμό του.

²³ Database, Wikipedia, <https://en.wikipedia.org/wiki/Database>

Βασικά μέρη ενός DBMS

Τα βασικά μέρη ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων είναι τέσσερα, τα οποία είναι τα εξής:

1. Γλώσσα Ορισμού Δεδομένων: Ορίζει τη δομή και το περιεχόμενο της βάσης δεδομένων.
2. Γλώσσα Χειρισμού Δεδομένων: Χειρίζεται τα δεδομένα για τις εφαρμογές που χρησιμοποιούν τη βάση δεδομένων.
3. Λεξικό Δεδομένων: Είναι ένα ηλεκτρονικό κείμενο που παρέχει αναλυτικές πληροφορίες για κάθε δεδομένο που βρίσκεται στη βάση δεδομένων (για παράδειγμα τα δικαιώματα χρηστών, υπεύθυνος ενημέρωσης της βάσης δεδομένων).
4. Υπηρεσίες και Αναφορές: Λογισμικό που παράγει αναφορές και καθιστά την βάση δεδομένων πιο φιλική για τον χρήστη.

Χαρακτηριστικά και Ιδιότητες των DBMS

- Μοίρασμα των δεδομένων και χρήση τους από πολλούς χρήστες
- Συστήματα βάσεων δεδομένων με κλήση στην περιγραφή των δυνατοτήτων τους ως προς την ομοιόμορφη δημιουργία, την πρόσβαση και τον έλεγχο στα δεδομένα
- Υποστήριξη κανόνων ακεραιότητας, ασφάλειας και δικαιοδοσίας στα δεδομένα
- Κεντρικός έλεγχος λειτουργίας
- Πολλαπλοί τρόποι πρόσβασης
- Έλεγχος συνδρομικότητας και μηχανισμοί για ανάκαμψη
- Τυποποιήσεις
- Αύξησης της παραγωγικότητας στην ανάπτυξη εφαρμογών
- Απομόνωση προγραμμάτων και δεδομένων, αφαίρεση δεδομένων
- Απόδοση

Κεφάλαιο 3

Σχέδιο Δράσης για την εκπόνηση της Πτυχιακής Εργασίας

3.1 Περίληψη

Σε αυτό το κεφάλαιο θα κάνουμε ανάλυση των βασικών τεχνολογιών που τελικά χρησιμοποιήσαμε για την υλοποίηση της εργασίας μας, δηλαδή την γλώσσα XML καθώς και την PHP γλώσσα προγραμματισμού, που ήταν η βασική γλώσσα διαχείρισης του συστήματος μας.

Η δικτυοκεντρική βάση δεδομένων μας για μία ασφαλιστική εταιρείας έχοντας αρχικά την μορφή ενός απλού ιστοτόπου που μπορεί να έχουν πρόσβαση οι πελάτες του ώστε να ενημερώνονται για τα νέα ασφαλιστικά πακέτα της εταιρείας και στην συνέχεια, ο υπεύθυνος υπάλληλος, με χρήση των μυστικών κωδικών μπορεί να εισέλθει στην σελίδα Διαχείρισης και να έχει πρόσβαση σε όλες τις λειτουργίες που χρειάζεται για την ασφαλιστική εταιρεία.

3.2 Η γλώσσα σήμανσης XML (Extensible Markup Language)

Εισαγωγή

Με την τεράστια διάδοση και χρήση του Παγκόσμιου Ιστού, ολοένα και περισσότερος όγκος πληροφοριών ανταλλάσσεται καθημερινά μέσω του Διαδικτύου. Εξαιτίας της τεράστιας αυτής ζήτησης πληροφοριών και επειδή οι πηγές πληροφορίας πρέπει να είναι εύκολα προσβάσιμες και ευέλικτες ως προς την μεταφορά των πληροφοριών και επίσης ανεξάρτητες ως προς το λειτουργικό σύστημα και το περιεχόμενο, γι' αυτό το λόγο αναπτύχθηκαν οι γλώσσες σήμανσης.

Στην ουσία, μία γλώσσα σήμανσης δεν είναι τίποτα παραπάνω από ένα σύνολο εντολών που ονομάζονται ετικέτες (tags), οι οποίες επιτρέπουν την προσθήκη πληροφοριών στο περιεχόμενο μιας πηγής δεδομένων. Οι πληροφορίες αυτές καταφέρνουν να ξεχωρίζουν το περιεχόμενο από το υπόλοιπο κείμενο. Αυτή η γλώσσα που το καταφέρνει αυτό είναι η XML²⁴.

Δημιουργήθηκε από το XML Working Group του οργανισμού World Wide Web Consortium το 1996. Είναι μία γλώσσα που στην ουσία αποτελεί το υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language), η οποία είναι πρότυπο κατά ISO 8879:1986²⁵.

Πιο συγκεκριμένα, η γλώσσα XML δημιουργήθηκε για να μπορέσει να καλύψει τα κενά της HTML, εφόσον ήταν δύσκολο και πολύπλοκο όπως αποδείχτηκε να το καταφέρει η SGML σε μία προσπάθεια της. Για την HTML και την γλώσσα σήμανσης SGML έχουμε αναφερθεί στο Κεφάλαιο 2. Μερικά από τα κενά που έπρεπε να καλύψει η XML είναι η

²⁴ XML, Wikipedia, <https://el.wikipedia.org/wiki/XML>

²⁵ SGML, Wikipedia, https://en.wikipedia.org/wiki/Standard_Generalized_Markup_Language

δυνατότητα δημιουργίας νέων ετικετών, για να μην χρειάζεται η υπερβολική χρήση ετικετών εφόσον ένας τύπος εγγράφου χρησιμοποιείται για όλους τους σκοπούς και επίσης οι περισσότερες από αυτές τις ετικέτες χρησιμοποιούνται για τον τρόπο εμφάνισης του συγκεκριμένου εγγράφου και όχι για να ξεχωρίζουν το περιεχόμενο από το υπόλοιπο κείμενο.

Η Κοινοπραξία του Παγκόσμιου Ιστού (W3C), τα πρότυπα που όρισε για την σωστή χρήση της γλώσσα XML είναι τα εξής:

- Να είναι εύχρηστη στο Ίντερνετ
- Να υποστηρίζει πλήθος εφαρμογών
- Να είναι συμβατή με την SGML
- Ευκολία ανάπτυξης προγραμμάτων που επεξεργάζονται XML έγγραφα
- Ο αριθμός των προαιρετικών χαρακτηριστικών στην XML να είναι όσο το δυνατόν πιο μικρός, ιδανικό επίπεδο το μηδέν
- Τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα
- Ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα
- Ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός
- Τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα
- Η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας

Κύρια Χαρακτηριστικά

Τα κύρια χαρακτηριστικά της XML είναι:

- Παρέχει ακριβή καθορισμό του περιεχομένου, ώστε τα αποτελέσματα μιας αναζήτησης να είναι ικανοποιητικά
- Παρέχει δυνατότητα καθορισμού των στοιχείων του εγγράφου (για παράδειγμα Τίτλος, Συγγραφέας, Εκδότης, Ημερομηνία Έκδοσης)
- Επιτρέπει τη μεταφορά μέσω διαδικτύου, δομημένης πληροφορίας ανεξάρτητα από πλατφόρμες και συστήματα.

Πλεονεκτήματα

Τα πλεονεκτήματα της XML είναι τα εξής:

- Πιο ικανοποιητικές αναζητήσεις
- Ανάπτυξη πιο ευέλικτων εφαρμογών
- Ενσωμάτωση δεδομένων από ανόμοιες πηγές πληροφορίας
- Δεδομένα από πολλαπλές εφαρμογές
- Υπολογισμός και διαχείριση δεδομένων τοπικά
- Πολλαπλοί τρόποι εμφάνισης των δεδομένων
- Ενημέρωση πεδίων
- Διαθέσιμα Πρότυπα
- Μορφοποίηση εμφάνισης για το διαδίκτυο²⁶

²⁶ Η γλώσσα σήμανσης XML, Δημήτριος Σάμψων, Πανεπιστημιακές Σημειώσεις Δεκεμβρίου 2003,

Διαφορές XML με HTML

Η γλώσσα σήμανσης XML σε καμία περίπτωση δεν αντικαθιστά την γλώσσα HTML. Η HTML είναι υπεύθυνη για την σωστή διατύπωση και εμφάνιση των δεδομένων και η XML είναι υπεύθυνη ώστε να περιγράψει και να δείξει πια είναι η πληροφορία.

Στην HTML έχουν έτοιμες ετικέτες στις οποίες δεν μπορούμε να παρέμβουμε, ενώ με την XML μπορούμε να δημιουργήσουμε από την αρχή ετικέτες και να καθορίσουμε τι κάνουν αυτές οι ετικέτες.

3.2.1 Μορφή εγγράφου XML

Στην XML γλώσσα, όπως και στην HTML έχουμε τις ετικέτες (tags), οι οποίες έχουν την μορφή <ετικέτα>...</ετικέτα>. Στην HTML οι ετικέτες είναι προκαθορισμένες όμως στην XML έχουμε την δυνατότητα να δημιουργήσουμε εμείς ονόματα ετικετών που χρειαζόμαστε. Οι ετικέτες αυτές οριοθετούνται ανάμεσα σε < > και χρησιμοποιούνται σε ζεύγη.

Τα XML αρχεία συνήθως αρχίζουν με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως είναι το εξής:

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

Παράδειγμα XML εγγράφου για την περιγραφή της μορφής ενός XML αρχείου

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Αυτό είναι ένα σχόλιο -->
<κατάλογος>
  <cd id="01">
    <τίτλος>Everything is Christmas</τίτλος>
    <καλλιτέχνης>Sia</καλλιτέχνης>
    <κομμάτια>10</κομμάτια>
    <έτος >2017</έτος>
    <τιμή νόμισμα="euro" >18</τιμή>
  </cd>
  <cd id="02">
    <τίτλος>This is acting</τίτλος>
    <καλλιτέχνης>Sia</καλλιτέχνης>
    <κομμάτια>12</κομμάτια>
    <έτος >2016</έτος>
    <τιμή νόμισμα="euro" >17</τιμή>
  </cd>
</κατάλογος>
```

Η μορφή ενός εγγράφου XML πρέπει να ξεκινάει με το στοιχείο ρίζας, το οποίο περιλαμβάνει κάθε άλλο στοιχείο και είναι ακριβώς το πρώτο μετά την δήλωση XML. Το στοιχείο ρίζας είναι στην κορυφή της ιεραρχίας και στο παράδειγμα μας είναι το <κατάλογος></κατάλογος>. Με την ιδιότητα της ενθυλάκωσης (nesting) μπορούν να δημιουργηθούν στοιχεία με πολύπλοκη εσωτερική δομή.

Κάθε στοιχείο element έχει την δυνατότητα να παρέχει κάποιες ιδιότητες attributes, όπως είναι το στοιχείο <τιμή> το οποίο περιέχει το γνώρισμα νόμισμα=euro. Τα γνωρίσματα, σε σχέση με τα στοιχεία γράφονται πάντα ανάμεσα από διπλά εισαγωγικά. Τα γνωρίσματα θεωρούνται ένας εναλλακτικός τρόπος αναπαράστασης δεδομένων σε ένα XML αρχείο.

Επίσης, υπάρχει η δυνατότητα προσθήκης μοναδικών ταυτοτήτων σε κάποιο συγκεκριμένο στοιχείο, το οποίο ονομάζεται id. Αυτό το γνώρισμα, id, όπως γνωρίζουμε και από την HTML μπορεί να αναφερθεί σαν αναφορά του στοιχείου από άλλα του στοιχεία του κειμένου και πρέπει να είναι μοναδικό.

Πιο αναλυτικά, σε σχέση με το παράδειγμα μας, έχουμε, την δήλωση η οποία δηλώνει την έκδοση (version) της XML γλώσσας που χρησιμοποιήσαμε στο έγγραφο, στην δικιά μας περίπτωση την έκδοση 1.0 και το encoding="UTF-8"²⁷ είναι για μπορούν να χρησιμοποιηθούν και οι ελληνικοί χαρακτήρες. Έπειτα έχουμε το στοιχείο ρίζας το οποίο είναι ο <κατάλογος>, το οποίο περιέχει κάποια στοιχεία. Τα στοιχεία <τίτλος>, <καλλιτέχνης>, <κομμάτια>, <έτος> και <τιμή> έχουν περιεχόμενο κειμένου επειδή περιέχουν κείμενο. Τα στοιχεία <κατάλογος> και <cd> έχουν περιεχόμενο στοιχεία, επειδή περιέχουν στοιχεία. Το στοιχείο <τιμή> έχει ιδιότητα attribute νόμισμα="euro" και το στοιχείο <cd> έχει ένα μοναδικό id.

Και τέλος, όπως σε όλες τις γλώσσες σήμανσης ή προγραμματισμού, έχουμε την δυνατότητα πρόσθεσης σχολίων σε όποιο σημείο του εγγράφου θέλουμε. Στην XML για να μπορέσουμε να προσθέσουμε κάποιο σχόλιο τότε εισάγουμε <!-- γράφουμε το σχόλιο μας και έπειτα κλείνουμε το σχόλιο με -->.²⁸

3.2.2 Τύποι αρχείων XML

Τα XML αρχεία χωρίζονται σε δύο τύπους εγγράφων:

- Τα ορθά μορφοποιημένα ή καλά ορισμένα έγγραφα (Well-Formed)
- Και τα έγκυρα έγγραφα (valid)

Well-formed Έγγραφα

²⁷ UTF-8, Wikipedia, <https://el.wikipedia.org/wiki/UTF-8>

²⁸ XML, Wikipedia, <https://en.wikipedia.org/wiki/XML> &

XML Tutorial, W3schools, <https://www.w3schools.com/xml/default.asp> &

XML Family of Specifications: A Practical Guide, Kenneth B. Sall, 2002 &

Η γλώσσα σήμανσης XML, Δημήτριος Σάμπσον, Πανεπιστημιακές Σημειώσεις Δεκεμβρίου 2003, www.fme.aegean.gr/sites/default/files/dsampsom_xml_lectures-notes-dec2003.pdf

Τα ορθά μορφοποιημένα ή καλά ορισμένα έγγραφα σημαίνει ότι ακολουθεί τους γενικούς κανόνες σύνταξης της XML, οι οποίοι σε σχέση με της HTML και της SGML είναι πιο απαιτητικοί, καθώς δεν αφήνει περιθώρια παρέκκλισης.

Μερικοί από τους κανόνες αυτούς είναι:

- Οι ετικέτες δεδομένων της XML πρέπει πάντα να έχουν μία ετικέτα τέλους. Είτε αυτές είναι σε ζεύγη όπως <ετικέτα>...</ετικέτα>, είτε ακόμα και αν έχουμε να κάνουμε με ετικέτα κενού στοιχείου όπως </ετικέτα>.
- Η σήμανση της XML ξεκινάει πάντοτε με το σύμβολο < ή &.
- Οι τύποι των στοιχείων και τα ονόματα των εισαγωγικών αναγνωρίζουν πεζά και κεφαλαία.
- Οι τιμές των ιδιοτήτων απαιτούν εισαγωγικά²⁹

Valid Έγγραφο

Τα έγκυρα XML έγγραφα είναι εκείνα που ακολουθούν ένα συγκεκριμένο τύπο εγγράφου (Document Type Definition – DTD³⁰), ένα σετ κανόνων δηλαδή που ακολουθεί ένα έγγραφο, τους οποίους κανόνες το λογισμικό έχει την ικανότητα να διαβάσει πριν την επεξεργασία και την εμφάνιση του εγγράφου. Αυτοί οι κανόνες καθορίζουν στην ουσία το όνομα και το περιεχόμενο κάθε στοιχείου element.

Σε σχέση με τα ορθά μορφοποιημένα, τα έγκυρα XML προσφέρουν περισσότερα στην επεξεργασία του εγγράφου. Όλες οι διεργασίες που μπορούν να γίνουν σε ένα έγγραφο XML, συγγραφή, επεξεργασία, αποθήκευση και εμφάνιση του εγγράφου, έγιναν ευκολότερες γιατί οι εγγραφές των στοιχείων γίνονται δομημένα.

Διαφορές ανάμεσα σε Well-formed και Valid έγγραφα

Η πιο σημαντική διαφορά μεταξύ των δύο αυτών εγγράφων είναι ότι στα ορθά μορφοποιημένα έγγραφα τον έλεγχο για την ορθή μορφοποίηση πρέπει να την κάνει το λογισμικό πλοήγησης XML, ενώ τα έγκυρα πρέπει να τα γράφονται και να ελέγχονται από τους ίδιους τους δημιουργούς τους.

Αν ένα αντικείμενο δεδομένων είναι ορθά μορφοποιημένο τότε είναι ένα XML έγγραφο. Ένα ορθά μορφοποιημένο έγγραφο μπορεί να είναι έγκυρο, μόνο εάν πλήρη κάποιους περιορισμούς.

Ένα ορθά μορφοποιημένο έγγραφο XML είναι σχεδιασμένο για χρήση χωρίς DTD, σε σχέση με τα έγκυρα XML έγγραφα που ρητά απαιτούν την χρήση DTD³¹.

3.2.2.1 Περιγραφή Δεδομένων – Ορισμός Τύπου Εγγράφου (DTD)

Ο Ορισμός Τύπου Εγγράφου (DTD) είναι ένα σύνολο δηλώσεων σήμανσης που ορίζουν έναν τύπο εγγράφου για μια γλώσσα σήμανσης οικογένειας SGML (SGML, XML,

²⁹ Well-formed XML έγγραφα, <http://www.it.uom.gr/project/xml/wellformed.htm>

³⁰ Document Type Definition, Wikipedia, https://en.wikipedia.org/wiki/Document_type_definition

³¹ Valid XML έγγραφα, <http://www.it.uom.gr/project/xml/valid.htm>

HTML). Ένα DTD ορίζει τη δομή και τα νομικά στοιχεία και χαρακτηριστικά ενός εγγράφου XML και μπορεί να δηλωθεί μέσα σε ένα έγγραφο XML ή ως εξωτερική αναφορά.

Η XML χρησιμοποιεί τον ορισμό τύπου εγγράφου DTD για να περιγράψει τα δεδομένα, όπως είναι τα στοιχεία και τα χαρακτηριστικά. Αποτελεί ένα σύνολο κανόνων ή δηλώσεων για τις ετικέτες και το ποια είναι τα επιτρεπτά ονόματα στοιχείων και τα επιτρεπτά υποστοιχεία subelements και χαρακτηριστικά attributes για κάθε συγκεκριμένο στοιχείο.

Παράδειγμα ενός εσωτερικού DTD

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

```
<!DOCTYPE note [  
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  

```

```
<note>  
<to>Νικόλαος</to>  
<from>Χρυσάνθη</from>  
<heading>Εργασία</heading>  
<body>Αυτό είναι ένα παράδειγμα!</body>  

```

Στο συγκεκριμένο παράδειγμα βλέπουμε τις δηλώσεις του DTD μέσα στο XML έγγραφο. Εάν το DTD δηλωθεί μέσα στο αρχείο XML, πρέπει να δηλωθεί μέσα <!DOCTYPE> ορισμό. Πιο αναλυτικά για το παράδειγμα μας:

- !DOCTYPE note: ορίζει ότι το στοιχείο ρίζας αυτού του εγγράφου είναι το note
- !ELEMENT note: ορίζει ότι το στοιχείο note πρέπει να περιέχει τέσσερα στοιχεία: "προς, από, τίτλο, σώμα"
- !ELEMENT to: ορίζει ότι το στοιχείο to είναι τύπου "#PCDATA"
- !ELEMENT from: ορίζει ότι το στοιχείο from είναι τύπου "#PCDATA"
- !ELEMENT heading: ορίζει ότι το στοιχείο heading είναι τύπου "#PCDATA"
- !ELEMENT body: ορίζει ότι το στοιχείο body είναι τύπου "#PCDATA"

Και ακριβώς από κάτω έχουμε τα δεδομένα του XML εγγράφου³².

³² DTD Tutorial, W3Schools, https://www.w3schools.com/xml/xml_dtd_intro.asp

Παράδειγμα ενός εξωτερικού DTD

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

```
<!DOCTYPE note SYSTEM "note.dtd">
```

```
<note>
```

```
<to>Νικόλαος</to>
```

```
<from>Χρυσάνθη</from>
```

```
<heading>Εργασία</heading>
```

```
<body>Αυτό είναι ένα παράδειγμα!</body>
```

```
</note>
```

Εάν το DTD δηλωθεί σε ένα εξωτερικό αρχείο, ο ορισμός <! DOCTYPE> πρέπει να περιέχει μια αναφορά στο αρχείο DTD. Και η μορφή του εξωτερικού αρχείου note.dtd θα έχει την εξής μορφή:

```
<!ELEMENT note (to,from,heading,body)>
```

```
<!ELEMENT to (#PCDATA)>
```

```
<!ELEMENT from (#PCDATA)>
```

```
<!ELEMENT heading (#PCDATA)>
```

```
<!ELEMENT body (#PCDATA)>
```

Η ανάλυση εδώ είναι ίδια, με την ανάλυση του παραδείγματος ενός εσωτερικού DTD.

Κανόνες δομής και σύνταξης ενός DTD

- Όλα τα ονόματα των αντικειμένων ξεκινούν με γράμμα και δεν επιτρέπεται ο χαρακτήρας κενού διαστήματος, εάν πρέπει να χρησιμοποιηθεί το κενό, τότε στην θέση του μπορεί να μπει ο ενωτικός χαρακτήρας.
- Με την χρήση του πρότυπου κωδικοποίησης χαρακτήρων Unicode: UTF-8, μας δίνεται η δυνατότητα χρήσης των Ελληνικών χαρακτήρων.
- Στον τύπο περιεχομένου που περιέχει μόνο στοιχεία είναι δυνατό να εφαρμοστούν στα στοιχεία, ένα σύνολο ειδικών χαρακτήρων που τους προσδιορίζουν ορισμένα χαρακτηριστικά. Οι χαρακτήρες *, ?, και + ονομάζονται χαρακτήρες ύπαρξης. Υποδεικνύουν αν και πώς τα στοιχεία στην λίστα επαναλαμβάνονται, αν δεν υπάρχει δείκτης ύπαρξης σε ένα στοιχείο αυτό σημαίνει ότι το συγκεκριμένο στοιχείο εμφανίζεται μια φορά. Στοιχείο με τον χαρακτήρα + πρέπει να εμφανίζεται μια τουλάχιστον φορά. Με τον χαρακτήρα * καμία ή περισσότερες φορές. Με τον χαρακτήρα ? σημαίνει ότι το στοιχείο μπορεί να εμφανιστεί το πολύ μια φορά. Οι χαρακτήρες ,(κόμμα) και | ονομάζονται συζευκτικές. Βοηθούν στον διαχωρισμό των παιδιών ενός στοιχείου μέσα στο μοντέλο περιεχομένου. Ο χαρακτήρας ,(κόμμα) σημαίνει ότι τα δύο στοιχεία αριστερά και δεξιά του κόμματος πρέπει να

εμφανίζονται με την ίδια σειρά μέσα στο έγγραφο. Ενώ ο χαρακτήρας | σημαίνει ότι μόνο το ένα από τα στοιχεία πρέπει να εμφανίζεται στο έγγραφο³³.

Έλεγχος εγκυρότητας ενός XML εγγράφου

Η εγκυρότητα ενός XML εγγράφου μπορεί να επαληθευτεί με την χρήση οποιουδήποτε προγράμματος validator. Τα προγράμματα αυτά ελέγχουν το έγγραφο αν συμμορφώνεται με τους κανόνες του DTD.

Τα γνωστότερα προγράμματα validators χωρίζονται σε τρεις κατηγορίες όπως είναι:

- Γραμμής εντολών: XSV
- Προγράμματα με API για τους προγραμματιστές: Apache Xerces, IBM Schema Quality Checker, Microsoft MSXML 4.0
- Με γραφικό περιβάλλον: XML Spy, Turbo XML

3.2.2.2 XSD – Ορισμός Σχήματος XML (XML Schema)

Το XSD (XML Schema Definition), είναι μια σύσταση της Κοινοπραξίας World Wide Web (W3C), καθορίζει τον τρόπο επίσημης περιγραφής των στοιχείων σε ένα έγγραφο XML. Μπορεί να χρησιμοποιηθεί από προγραμματιστές για να επαληθεύσει κάθε κομμάτι περιεχομένου ενός εγγράφου. Μπορούν να ελέγξουν αν τηρεί την περιγραφή του στοιχείου στο οποίο είναι τοποθετημένο.

Όπως όλες οι γλώσσες σχήματος XML, έτσι και η XSD μπορεί να χρησιμοποιηθεί για να εκφράσει ένα σύνολο κανόνων στους οποίους ένα έγγραφο XML πρέπει να συμμορφώνεται, προκειμένου να θεωρηθεί "έγκυρο" σύμφωνα με αυτό το σχήμα. Ωστόσο, σε αντίθεση με τις περισσότερες άλλες γλώσσες σχημάτων, η XSD σχεδιάστηκε επίσης με την πρόθεση ότι ο προσδιορισμός της εγκυρότητας ενός εγγράφου θα παρήγαγε μια συλλογή πληροφοριών που θα τηρούν συγκεκριμένους τύπους δεδομένων³⁴.

Σε σχέση με το DTD, η XSD, δίνει την δυνατότητα ενός πιο λεπτομερούς τρόπου για το τι μπορούν ή δεν μπορούν να περιέχουν τα δεδομένα ενός XML εγγράφου. Πιο συγκεκριμένα οι επιπλέον λειτουργίες οι οποίες παρέχει είναι οι εξής:

- Προσδιορίζει τα στοιχεία και τις ιδιότητες που μπορούν να εμφανιστούν σε ένα έγγραφο
- Για τα στοιχεία-παιδιά, προσδιορίζει ποια στοιχεία είναι, τη διάταξη τους και τον αριθμό τους
- Προσδιορίζει εάν ένα στοιχείο είναι κενό ή αν μπορεί να περιέχει κείμενο
- Προσδιορίζει τους τύπου δεδομένων data types για τα στοιχεία και τις ιδιότητες

³³ Η γλώσσα σήμανσης XML, Δημήτριος Σάμψων, Πανεπιστημιακές Σημειώσεις Δεκεμβρίου 2003, www.fme.aegean.gr/sites/default/files/dsampsom_xml_lectures-notes-dec2003.pdf

³⁴ XML Schema Definition, Wikipedia, [https://en.wikipedia.org/wiki/XML_Schema_\(W3C\)](https://en.wikipedia.org/wiki/XML_Schema_(W3C)) & XML Schema Tutorial, W3Schools, https://www.w3schools.com/xml/schema_intro.asp

- Προσδιορίζει τις προεπιλεγμένες default και τις σταθερές τιμές fixed values για τα στοιχεία και τις ιδιότητες

Παράδειγμα ενός XSD

Απλό έγγραφο XML note.xml

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

<note xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="note.xsd">
<to>Νικόλαος</to>
<from>Χρυσάνθη</from>
<heading>Εργασία</heading>
<body>Αυτό είναι ένα παράδειγμα!</body>
</note>
```

XML Schema note.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Στο παράδειγμα μας έχουμε το απλό αρχείο XML, με τις ίδιες ετικέτες και τα δεδομένα από τα προηγούμενα παραδείγματα του DTD, για να μπορέσουμε να δούμε τις επιπλέον πληροφορίες που μας παρέχει η XSD.

Στην παραπάνω γραμμή: xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" λέμε στον XML parser ότι το έγγραφο αυτό πρέπει να επικυρωθεί έναντι ενός σχήματος. Η γραμμή: xsi: noNamespaceSchemaLocation = "note.xsd" καθορίζει ΠΟΥ βρίσκεται το σχήμα αυτό, στον ίδιο φάκελο με το όνομα note.xsd.

Μέσα στο έγγραφο note.xsd έχουμε, εκτός από την κλασική δήλωση όπως ενός XML αρχείου έχουμε και την δήλωση ενός xml schema, το οποίο είναι της μορφής: <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">.

Στη συνέχεια, πρέπει να ορίσουμε το στοιχείο "note". Αυτό το στοιχείο έχει ένα χαρακτηριστικό γνώρισμα και περιέχει άλλα στοιχεία, επομένως το θεωρούμε πολύπλοκο. Τα παιδικά στοιχεία του στοιχείου "note" περιβάλλεται από ένα στοιχείο αλληλουχίας xs: το οποίο ορίζει μια διατεταγμένη ακολουθία από επιμέρους στοιχεία:

```
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Στη συνέχεια, πρέπει να ορίσουμε το στοιχείο "to" ως απλό τύπο (επειδή δεν περιέχει άλλα χαρακτηριστικά ή άλλα στοιχεία). Ο τύπος (xs: string) έχει πρόθεμα με το πρόθεμα χώρου ονομάτων που σχετίζεται με το XML Schema που υποδηλώνει έναν προκαθορισμένο τύπο δεδομένων σχήματος:

```
<xs:element name="to" type="xs:string"/>
```

Με αυτόν τον τρόπο δηλώνουμε και τα υπόλοιπα στοιχεία του XML εγγράφου.

Πλεονεκτήματα ενός XML Schema

Υπάρχουν αρκετοί λόγοι για τους οποίους το XSD είναι πολύ καλύτερο από το DTD. Μερικοί από τους λόγους αυτούς είναι οι εξής:

- *Υποστηρίζει τύπους δεδομένων.* Αυτό σημαίνει ότι πολύ ευκολότερο να περιγραφεί η επιτρεπόμενη περιεκτικότητα σε έγγραφα, να επικυρωθεί η ακρίβεια των στοιχείων, η συνεργασία με τα στοιχεία από μια βάση δεδομένων, να καθοριστούν οι μορφές δεδομένων, να μετατραπούν τα στοιχεία μεταξύ των διαφορετικών τύπων δεδομένων
- *Χρησιμοποιούν την σύνταξη XML.* Αυτό τους δίνει τα εξής πλεονεκτήματα: Ο χρήστης δεν είναι απαραίτητο να μάθει μια άλλη γλώσσα, μπορεί να χρησιμοποιήσει το πρόγραμμα σύνταξης XML για να διαμορφώσει τα αρχεία με τα XML Schemas, μπορεί να χρησιμοποιήσει επίσης τον αναλυτή parser της XML για να αναλύσει τα XML Schemas και τέλος μπορεί να μετασχηματίσει το XML Schema με το XSLT.
- *Εξασφαλίζουν τη μετάδοση των δεδομένων.* Όταν τα δεδομένα στέλνονται από έναν αποστολέα σε έναν δέκτη πρέπει και οι δύο μεριές να περιμένουν να διαβάσουν τα ίδια για το περιεχόμενο. Με τα XML Schemas, ο αποστολέας μπορεί να περιγράψει τα στοιχεία με έναν τρόπο που ο δέκτης μπορεί να καταλάβει.
- *Είναι επεκτάσιμα.* Ακριβώς όπως και η XML. Μπορεί να επαναχρησιμοποιηθεί ένα σχήμα σε άλλα σχήματα. Μπορούν να δημιουργηθούν καινούργιοι τύποι στοιχείων που θα προέρχονται από τύπους που ήδη υπάρχουν. Μπορεί να γίνει αναφορά σε πολλαπλά σχήματα από το ίδιο έγγραφο.

- *Τηρούν τους κανόνες σύνταξης της XML.* Αυτό σημαίνει ότι το XML Schema πρέπει να ξεκινήσει με την δήλωση XML και να έχει ένα μοναδικό στοιχείο ρίζας root. Όλες οι ετικέτες έναρξης να έχουν και ετικέτες τέλους. Όλες οι τιμές των οντοτήτων πρέπει να αναφέρονται επακριβώς.³⁵

3.2.3 Αναλυτές XML (XML Parsers)

Η ανάλυση parsing είναι η πράξη της διάσπασης των πληροφοριών στα συστατικά μέρη της. Ένα απλό παράδειγμα για να μπορέσουμε να καταλάβουμε τι εννοούμε με αυτό είναι, όπως στα σχολεία είχαμε να μάθουμε να κάνουμε ανάλυση προτάσεων για να βρούμε το Υποκείμενο – Ρήμα – Αντικείμενο. Ακόμα πιο συγκεκριμένα, έχουμε την παρακάτω πρόταση:

Η Βαρβάρα διαβάζει ένα βιβλίο

Η ανάλυση της παραπάνω πρότασης θα είναι:

- Υποκείμενο: *Βαρβάρα*, ονομαστικό ουσιαστικό
- Ρήμα: *διαβάζει*, τρίτο ενικό πρόσωπο, ενεργητική φωνή, παροντικός χρόνος
- Αντικείμενο: *βιβλίο*, ουσιαστικό

Στην πληροφορική, ένας αναλυτής parser είναι ένα πρόγραμμα (ή ένα κομμάτι κώδικα ή API που μπορείτε να αναφέρετε μέσα στα δικά σας προγράμματα), το οποίο αναλύει τα αρχεία για να ταυτοποιήσει τα συστατικά μέρη. Όλες οι εφαρμογές που διαβάζουν δεδομένα έχουν έναν αναλυτή κάποιου είδους, διαφορετικά δεν θα μπορούσαν ποτέ να καταλάβουν τι σημαίνουν οι πληροφορίες αυτές.

Παραδείγματα εφαρμογών είναι για παράδειγμα το Microsoft Word περιέχει έναν αναλυτή που εκτελείται όταν ανοίγετε ένα αρχείο .doc και ελέγχει ότι μπορεί να αναγνωρίσει όλους τους κρυφούς κώδικες. Το iCal και το Ημερολόγιο Google περιέχουν έναν αναλυτή ο οποίος διαβάζει ένα συνημμένο ical ραντεβού στο ηλεκτρονικό σας ταχυδρομείο και επεξεργάζεται τις πληροφορίες που περιέχονται σε αυτό και εάν τους δώσετε ένα κατεστραμμένο αρχείο θα λάβετε ένα μήνυμα σφάλματος.

Οι εφαρμογές XML είναι ακριβώς οι ίδιες: περιέχουν έναν αναλυτή parser που διαβάζει XML και προσδιορίζει τη λειτουργία κάθε δεδομένου του εγγράφου και στη συνέχεια καθιστά αυτές τις πληροφορίες διαθέσιμες στη μνήμη για το υπόλοιπο πρόγραμμα.

Κατά την ανάγνωση ενός αρχείου XML, ένας αναλυτής parser ελέγχει τη σύνταξη (παρενθέσεις, ταιριαστά εισαγωγικά και άλλα) για ορθά μορφοποιημένα well-formed έγγραφα και αναφέρει τυχόν παραβιάσεις, πετώντας κάποιο μήνυμα. Για να ελέγξει εάν είναι ορθά μορφοποιημένα well-formed, ελέγχει την λίστα με τις προδιαγραφές XML³⁶.

³⁵ Η γλώσσα σήμανσης XML, Δημήτριος Σάμψων, Πανεπιστημιακές Σημειώσεις Δεκεμβρίου 2003, www.fme.aegean.gr/sites/default/files/dsamps_on_xml_lectures-notes-dec2003.pdf

³⁶ Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C 2008, <https://www.w3.org/TR/2008/REC-xml-20081126/>

Η εγκυρότητα XML εγγράφων (valid XML) είναι ένα άλλο στάδιο πέρα από την ανάλυση parser για τον έλεγχο ορθών μορφοποιημένων εγγράφων. Καθώς εντοπίζονται τα συστατικά μέρη του προγράμματος, ένας αναλυτής εγκυρότητας (valid parser) μπορεί να τα συγκρίνει με το πρότυπο που καθορίζεται από το DTD ή το Schema, για να ελέγξει ότι συμμορφώνονται. Κατά τη διαδικασία, οι προεπιλεγμένες τιμές και οι τύποι δεδομένων (εφόσον έχουν οριστεί) μπορούν να προστεθούν στο αποτέλεσμα εντός της μνήμης της εγκυρότητας που δίνει ο αναλυτής εγκυρότητας (valid parser) στην εφαρμογή.

Παράδειγμα χρήσης του αναλυτή (parser)

```
<άτομο xml: id = "01" γέννηση = "1990" φύλλο = "θηλυκό">
  <στοιχεία>
    <όνομα> Χρυσάνθη </όνομα>
    <επώνυμο> Παναγιωτάκη </επώνυμο>
  </στοιχεία>
</άτομο>
```

Το παραπάνω παράδειγμα αναλύετε (*parsing*) ως εξής:

--Το στοιχείο <άτομο> προσδιορισμένο με τρία χαρακτηριστικά attributes, xml: id = 01, γέννηση = 1990, φύλλο = θηλυκό. Το οποίο περιέχει ... :

--Το στοιχείο <στοιχεία> το οποίο με την σειρά του περιέχει ... :

---Το στοιχείο <όνομα>, το οποίο περιέχει το κείμενο «Χρυσάνθη» το οποίο ακολουθείται από ...

---Το στοιχείο <επίθετο>, το οποίο περιέχει το κείμενο «Παναγιωτάκη».

Αυτό καταλήγει ως ένα είδος δομής οικογενειακού δέντρου στη μνήμη της εφαρμογής (οι δομές των δέντρων είναι ένας κοινός τρόπος για τα προγράμματα αποθήκευσης σχετικών δεδομένων).

Εκτός από τους ενσωματωμένους αναλυτές, υπάρχουν επίσης αυτόνομοι αναλυτές, οι οποίοι διαβάζουν ένα αρχείο XML και σας λένε αν εντοπίζουν κάποιο σφάλμα³⁷.

3.2.4 Χρήση της XML για διαχείριση και μορφοποίηση κειμένων

Με την χρήση της γλώσσας XML μας η δίνεται η επιπλέον δυνατότητα διαχείρισης και μορφοποίησης κειμένων μέσα σε ένα XML έγγραφο, η διαχείριση και η μορφοποίηση όμως επιτυγχάνεται με την χρήση διάφορων άλλων γλωσσών ή εργαλείων. Πιο συγκεκριμένα, η διαχείριση των δεδομένων ενός κειμένου XML επιτυγχάνεται με τη χρήση μιας γλώσσας ερωτήσεων, ενώ η μορφοποίηση ενός κειμένου XML μπορεί να γίνει με την χρήση της γλώσσας σήμανσης HTML. Θα αναφερθούμε στην συνέχεια πιο αναλυτικά.

³⁷ What is parsing and how do I do it in XML?, Silmaril Consultants, <http://xml.silmaril.ie/parsers.html> & XML Parser, W3School, https://www.w3schools.com/xml/xml_parser.asp

Η διαχείριση των δεδομένων ενός αρχείου XML, είπαμε ότι επιτυγχάνεται με την χρήση μίας γλώσσας ερωτήσεων. Οι πιο διαδεδομένες γλώσσες διαχείρισης κειμένων XML είναι οι εξής: XPath, XQuery, Lorel, UnQL, XML-QL, XQL και άλλες. Κάθε μια από αυτές τις γλώσσες ερωτήσεων έχει τις δικές τις ιδιαιτερότητες και υποστηρίζει καλύτερα ή χειρότερα ορισμένες από τις διαδικασίες διαχείρισης, όπως είναι η εισαγωγή, διαγραφή και ενημέρωση των στοιχείων, ετικετών και γνωρισμάτων ενός κειμένου, την υποβολή και ανάκτηση δεδομένων μέσα από μεγάλα κείμενα XML.

Για την διαχείριση της μορφοποίησης ενός αρχείου XML, τον τρόπο δηλαδή εμφάνισης των στοιχείων του σε μία εφαρμογή, όπως είναι ένας φυλλομετρητής ή ένας κειμενογράφος, έχουν δημιουργηθεί τεχνολογίες. Οι τεχνολογίες αυτές επιτρέπουν την εύκολη επεξεργασία της εμφάνισης ενός XML αρχείου.

Μερικές από αυτές τις τεχνολογίες είναι η γλώσσα σήμανσης XHTML, που είναι το πάντρεμα της γλώσσα XML με την HTML. Αυτό που καταφέρνουμε με την XHTML γλώσσα σήμανσης είναι να ελέγχουμε την μορφοποίηση των XML εγγράφων, καθώς μπορούμε μέσα σε ένα XML αρχείο, να χρησιμοποιήσουμε τις ετικέτες μορφοποίησης της HTML. Επίσης η γλώσσα XHTML υπακούει στην απαιτητική δομή της XML γλώσσας.

Στη συνέχεια, περιγράφονται συνοπτικά τρεις από αυτές, η XPath, η XSL και η XQuery.

3.2.4.1 Γλώσσα ερωτήσεων XPATH (XML Path Language)

Η γλώσσα XPath είναι μία γλώσσα ερωτήσεων και όπως λέει και το όνομα της, δημιουργήθηκε για να παρέχει τα μονοπάτια σε ένα XML έγγραφο. Τα μονοπάτια αυτά δηλώνουν τους κόμβους ενός εγγράφου.

Η XPath μοντελοποιεί ένα έγγραφο XML ως ένα δέντρο κόμβων. Υπάρχουν διάφοροι τύποι κόμβων, συμπεριλαμβανομένων κόμβων στοιχείων element, κόμβων χαρακτηριστικών attributes και κόμβων κειμένου text. Το XPath ορίζει έναν τρόπο υπολογισμού μιας συμβολοσειράς για κάθε τύπο κόμβου. Ορισμένοι τύποι κόμβων έχουν επίσης ονόματα. Το XPath υποστηρίζει πλήρως τους χώρους ονόματος XML (XML Names). Επομένως, το όνομα ενός κόμβου μοντελοποιείται ως ζεύγος που αποτελείται από ένα τοπικό τμήμα και ένα ενδεχομένως μηδενικό URI ονομάτων.

Η πρωταρχική συντακτική κατασκευή στη XPath είναι η έκφραση. Μια έκφραση αξιολογείται για να δώσει ένα αντικείμενο, το οποίο έχει έναν από τους ακόλουθους τέσσερις βασικούς τύπους:

- Σύνολο κόμβων (μια μη ταξινομημένη συλλογή κόμβων χωρίς διπλότυπα)
- Boolean (σωστό ή λάθος)
- Έναν αριθμό (έναν αριθμό κινητής υποδιαστολής)
- Μία συμβολοσειρά (ακολουθία χαρακτήρων UCS)³⁸

³⁸ XPath language, W3C, 2016, <https://www.w3.org/TR/1999/REC-xpath-19991116/#section-Introduction>

Η XPath γλώσσα, χρησιμοποιεί μία βιβλιοθήκη συναρτήσεων, η οποία βοηθάει πρώτα στην ανάκτηση των δεδομένων μέσα από XML κείμενα και έπειτα βοηθάει στην επεξεργασία τους. Η βιβλιοθήκη αυτή είναι επεκτάσιμη ως προς τις συναρτήσεις που έχει.

Παράδειγμα χρήσης της γλώσσα ερωτήσεων XPath

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

```
<ταινίες>
<ταινία κατηγορία="κωμωδία">
  <τίτλος γλώσσα="EN"> Το Τετράγωνο </τίτλος>
  <σκηνοθέτης>Ρούμπεν Έστλουντ</σκηνοθέτης>
  <έτος>2017</έτος>
  <διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>
```

```
<ταινία κατηγορία="φαντασίας">
  <τίτλος γλώσσα="en">Wonder Woman</τίτλος>
  <σκηνοθέτης>Πάτι Τζένκινς</σκηνοθέτης>
  <έτος>2017</έτος>
  <διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>
```

```
<ταινία κατηγορία="κωμωδία">
  <τίτλος γλώσσα="en">Εγώ, ο Απαισιότατος 3</τίτλος>
  <σκηνοθέτης>Πιερ Κοφέν </σκηνοθέτης>
  <σκηνοθέτης>Κάιλ Μπάλντα</σκηνοθέτης>
  <έτος>2017</έτος>
  <διάρκεια χρόνος="λεπτά">180</διάρκεια>
</ταινία>
</ταινίες>
```

Ανάλυση παραδείγματος με την χρήση xpath

- Επέλεξε τα στοιχεία *τίτλος* που είναι φωλιασμένα στο στοιχείο-ρίζα ταινία: `/ταινία/τίτλος`
- Επέλεξε όλα τα στοιχεία *τίτλος*: `//τίτλος`
- Επέλεξε όλα τα στοιχεία που φωλιάζονται στο στοιχείο-ρίζα ταινία: `/ταινία/*`
- Επέλεξε όλα τα στοιχεία με το γνώρισμα attribute *κατηγορία* που έχουν την τιμή *κωμωδία*: `//ταινία[@κατηγορία='κωμωδία']`
- Επέλεξε όλους τους *τίτλους* των στοιχείων που έχουν *διάρκεια* πάνω από *150 λεπτά*: `/ταινίες/ταινία[διάρκεια>150]/τίτλος`

3.2.4.2 Γλώσσα μορφοποίησης XSL (XML Style Language)

Η γλώσσα XSL είναι μία γλώσσα μορφοποίησης και όπως λέει και το όνομα της, δημιουργήθηκε για να παρέχει την δυνατότητα στο περιεχόμενο ενός XML εγγράφου να είναι

ορατό από τον χρήστη και να αποκρύπτονται οι ετικέτες. Πιο απλά, η γλώσσα XSL (XML Style Language) σχεδιάστηκε ώστε να μπορεί να γίνεται μετασχηματισμός ενός κειμένου XML σε ένα HTML έγγραφο ώστε να μπορεί να χρησιμοποιείται από έναν φυλλομετρητή που δεν μπορεί να προσπελάσει XML δεδομένα. Η XSL εφαρμόζεται επίσης για τη μετατροπή κειμένων XML σε άλλα κείμενα XML.

Το μοντέλο της XSL είναι ιεραρχικό, αυτό σημαίνει ότι υπόκειται σε κανόνες προτύπων και πρέπει να ακολουθούν την παρακάτω μορφή.

κανόνας πρότυπο = συνθήκη + πρότυπο

Η εκτέλεση της ερώτησης είναι αναδρομική και αυτό σημαίνει ότι ακολουθεί τα εξής βήματα:

- Ξενικά από το στοιχείο ρίζα
- Ελέγχει μια συνθήκη pattern στον κόμβο αυτό
- Αν αυτή ικανοποιηθεί, εκτελεί το αντίστοιχο πρότυπο template
- Το πρότυπο δημιουργεί ένα κείμενο XML σαν αποτέλεσμα
- Και έπειτα συνεχίζει αναδρομικά στα υπόλοιπα παιδιά της ρίζας

Παράδειγμα χρήσης της γλώσσα XSD

Tainies.xml

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

```
<ταινίες>
<ταινία κατηγορία="κωμωδία">
  <τίτλος γλώσσα="EN"> Το Τετράγωνο </τίτλος>
  <σκηνοθέτης>Ρούμπεν Έστλουντ</σκηνοθέτης>
  <έτος>2017</έτος>
  <διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>

<ταινία κατηγορία="φαντασίας">
  <τίτλος γλώσσα="en">Wonder Woman</τίτλος>
  <σκηνοθέτης>Πάτι Τζένκινς</σκηνοθέτης>
  <έτος>2017</έτος>
  <διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>

<ταινία κατηγορία="κωμωδία">
  <τίτλος γλώσσα="en">Εγώ, ο Απαισιότατος 3</τίτλος>
  <σκηνοθέτης>Πιερ Κοφέν </σκηνοθέτης>
  <σκηνοθέτης>Κάιλ Μπάλντα</σκηνοθέτης>
  <έτος>2017</έτος>
```

```
<διάρκεια χρόνος="λεπτά">180</διάρκεια>
</ταινία>
</ταινίες>
```

Επέστρεψε όλους τους τίτλους των ταινιών με την χρήση της XSL

```
<xsl:template>
  <xsl:apply-templates/>
</xsl:template>
  <xsl:template match="/ταινίες/*/τίτλος">
    <result>
      <xsl:value-of/>
    </result>
  </xsl:template>
```

3.2.4.3 Γλώσσα ερωτήσεων XQuery

Η γλώσσα ερωτημάτων XQuery δημιουργήθηκε για την υποβολή ερωτήσεων σε ένα XML έγγραφο. Στην ουσία, η XQuery είναι μία πολύ λειτουργική γλώσσα ερωτημάτων, στην οποία κάθε ερώτημα είναι και μία έκφραση. Οι εκφράσεις αυτές περιλαμβάνουν επτά ευρύς τύπους:

1. εκφράσεις διαδρομής (path expressions)
2. κατασκευαστές στοιχείων (element constructors)
3. εκφράσεις FLWOR (FLWOR expressions)
4. εκφράσεις που περιέχουν χειριστές και συναρτήσεις,
5. υποθετικές εκφράσεις (conditional expressions),
6. ποσοτικές εκφράσεις ή
7. εκφράσεις που δοκιμάζουν ή μετατρέπουν τύπους δεδομένων.

Η σύνταξη και η σημασιολογία των διαφορετικών αυτών επτά εκφράσεων στην XQuery διαφέρουν σημαντικά. Η XQuery είναι ένα περίπλοκο σύστημα τύπων βασισμένο στους τύπους δεδομένων των XML schemas και υποστηρίζει τη διαχείριση των κόμβων του εγγράφου σε αντίθεση με την XPath.

Στην συνέχεια, με την χρήση της XQuery θα δούμε πως μπορούμε να βρούμε κάποιον συγκεκριμένο κόμβο, πως δημιουργούμε έναν κόμβο, συνδυασμό και αναδημιουργία κόμβων.

Με λίγα λόγια, η XQuery δεν διαφέρει πολύ από τις ερωτήσεις μίας βάσης δεδομένων SQL.

Παράδειγμα χρήσης της XQuery

Tainies.xml

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

<ταινίες>
  <ταινία κατηγορία="κωμωδία">
    <τίτλος γλώσσα="EN"> Το Τετράγωνο </τίτλος>
```

```
<σκηνοθέτης>Ρούμπεν Έστλουντ</σκηνοθέτης>
<έτος>2017</έτος>
<διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>
```

```
<ταινία κατηγορία="φαντασία">
<τίτλος γλώσσα="en">Wonder Woman</τίτλος>
<σκηνοθέτης>Πάτι Τζένκινς</σκηνοθέτης>
<έτος>2017</έτος>
<διάρκεια χρόνος="λεπτά">150</διάρκεια>
</ταινία>
```

```
<ταινία κατηγορία="κωμωδία">
<τίτλος γλώσσα="en">Εγώ, ο Απαισιότατος 3</τίτλος>
<σκηνοθέτης>Πιερ Κοφέν </σκηνοθέτης>
<σκηνοθέτης>Κάιλ Μπάλντα</σκηνοθέτης>
<έτος>2017</έτος>
<διάρκεια χρόνος="λεπτά">180</διάρκεια>
</ταινία>
</ταινίες>
```

Ανεύρεση συγκεκριμένου κόμβου:

Ένα απλό παράδειγμα χρήσης της XQuery για τον εντοπισμό των κόμβων είναι το εξής:

```
document("tainies.xml")/ταινίες/ταινία
```

Η παραπάνω δήλωση ανοίγει το έγγραφο tainies.xml χρησιμοποιώντας τη συνάρτηση document(), η οποία επιστρέφει τον κόμβο ρίζα του εγγράφου. Έπειτα δηλώνεται το /ταινίες που επιλέγει το στοιχείο ταινίες που βρίσκεται στην αρχή του εγγράφου και στη συνέχεια δηλώνεται το /ταινία που επιστρέφει όλα τα στοιχεία ταινία που βρίσκονται μέσα στο στοιχείο ταινίες.

Τα στοιχεία ταινία μπορούν να επιστραφούν χρησιμοποιώντας τους χαρακτήρες //, με βάση την παρακάτω δήλωση:

```
document("tainies.xml")//ταινία
```

Δημιουργίας ενός νέου κόμβου

Για τη δημιουργία των νέων στοιχείων, χρησιμοποιούνται οι αγκύλες οι οποίες περιέχουν δηλώσεις που εκτελούνται για τη δημιουργία ανοικτού περιεχομένου.

```
<example>
<p>Αναζήτηση</p>
<eg> document("tainies.xml")//ταινία[1]/τίτλος </eg>
<p> Αποτελέσματα Αναζήτησης.</p>
<eg>{ document("tainies.xml")//ταινία[1]/τίτλος }</eg>
</example>
```


Το αποτέλεσμα αυτού του ερωτήματος είναι το εξής:

```
<example>
<p>Αναζήτηση</p>
<eg> document("tainies.xml")//ταινία[1]/τίτλος </eg>
<p> Αποτελέσματα Αναζήτησης.</p>
<eg> Το Τετράγωνο </eg>
</example>
```

Οι δηλώσεις που περιέχονται στις αγκύλες επιτρέπουν τη δημιουργία νέων XML τιμών αναδομώντας υπάρχοντες XML τιμές. Επίσης, σημαντικό είναι να σημειωθεί ότι η συνάρτηση document("tainies.xml") επιστρέφει τη κόμβο ρίζας του τρέχοντος εγγράφου.

Συνδυασμό και αναδημιουργία κόμβων

Για να καταφέρουμε να κάνουμε συνδυασμό και αναδημιουργία κόμβων πρέπει να κάνουμε χρήση των δηλώσεων FLWOR της XQUERY. Οι δηλώσεις αυτές είναι παρόμοιες με τις δηλώσεις SELECT - FROM - WHERE που χρησιμοποιούνται στη γλώσσα SQL για τη διαχείριση των βάσεων δεδομένων.

Αντίθετα με την SQL, όπου οι δηλώσεις καθορίζονται με βάση τους πίνακες, τις εγγραφές και τα χαρακτηριστικά τους, οι FLWOR δηλώσεις δεσμεύουν μεταβλητές με τιμές μέσω των for και των let συντακτικών μονάδων (clause), και χρησιμοποιούν αυτές τις αντιστοιχίες για τη δημιουργία νέων αποτελεσμάτων. Ένας συνδυασμός τέτοιων δεσμεύσεων ονομάζεται συστοιχία (tuple).

Μια απλή FLWOR δήλωση που επιστρέφει τον τίτλο και την τιμή κάθε ταινία που προβλήθηκε τη χρονιά 2017 είναι η εξής:

```
for $b in document("tainies.xml")//tainia
where $b/@έτος="2017"
return $b/τίτλος
```

Η δήλωση αυτή δεσμεύει τη μεταβλητή \$b σε μία ταινία κάθε φορά, δημιουργώντας μια σειρά από συστοιχίες. Κάθε συστοιχία περιέχει μια δέσμευση κατά την οποία η μεταβλητή \$b αντιστοιχεί σε μία συγκεκριμένη ταινία. Η συντακτική μονάδα where ελέγχει κάθε συστοιχία για το αν ισχύει η ισότητα \$b/@έτος="2017", ενώ η συντακτική μονάδα return υπολογίζεται για κάθε συστοιχία που ικανοποιούν τη συνθήκη που εκφράστηκε στη μονάδα where. Το αποτέλεσμα του παραπάνω ερωτήματος μας θα εμφανίσει τα εξής:

```
<τίτλος>Το Τετράγωνο</τίτλος>
<τίτλος>Wonder Woman</τίτλος>
<τίτλος>Εγώ, ο Απαισιότατος 3</τίτλος>
```

Τα αρχικά FLWOR προέρχονται από τα πρώτα γράμματα των συντακτικών μονάδων που χρησιμοποιούνται στις εκφράσεις FLWOR και είναι οι εξής:

- for: δεσμεύει μια ή περισσότερες μεταβλητές σε δηλώσεις,
- let: δεσμεύουν μεταβλητές σε ολόκληρο το αποτέλεσμα μιας δήλωσης, είτε προσθέτοντας αυτές τις δεσμεύσεις στις συστοιχίες που δημιουργούνται από τη μονάδα for, είτε δημιουργώντας μια απλή συστοιχία που να περιέχει αυτές τις δεσμεύσεις σε περίπτωση που δεν έχει οριστεί μια for συντακτική μονάδα.
- where: φιλτράρουν συστοιχίες, διατηρώντας μόνο αυτές που ικανοποιούν μια συνθήκη.
- order by: ταξινομούν τις συστοιχίες σε μια αλληλουχία συστοιχιών
- return: δημιουργούν το αποτέλεσμα μιας FLWOR δήλωσης για μια δωσμένη συστοιχία

Μια FLWOR δήλωση ξεκινάει με μια ή περισσότερες for και let μονάδες τοποθετημένες σε οποιαδήποτε σειρά, ακολουθούμενες από τις συντακτικές μονάδες where, order by και return. Η χρήση των where και order είναι προαιρετική ενώ η χρήση της return είναι υποχρεωτική.

3.2.5 Χρήση της XML ως Βάση Δεδομένων

Ο πρωταρχικός ρόλος δημιουργίας της γλώσσας XML ήταν αρχικά να υποστηρίξει τις εφαρμογές διαδικτύου, πλέον λόγω της ευελιξίας της έχει συνδεθεί στενά με την τεχνολογία των συστημάτων βάσεων δεδομένων.

Ένα κείμενο XML αποτελεί μια συλλογή συσχετιζόμενων δεδομένων όπως έχουμε ξανά αναφέρει και γι' αυτό το λόγο συχνά αντιμετωπίζεται σαν μια Βάση Δεδομένων. Ένα κείμενο XML, σαν βάση δεδομένων, συγκεντρώνει ορισμένα πλεονεκτήματα και μειονεκτήματα έναντι των αρχείων των κλασικών βάσεων δεδομένων. Τα οποία θα αναλύσουμε στην συνέχεια.

Πλεονεκτήματα μιας XML Βάσης Δεδομένων

Ένα κείμενο XML έχει την ιδιότητα της αυτοπεριγραφής. Μέσω των ετικετών και των γνωρισμάτων, η γλώσσα XML περιγράφει τα δεδομένα του κειμένου. Βέβαια, η σημασιολογία αυτής της περιγραφής είναι περιορισμένη.

Ένα δεύτερο πλεονέκτημα αφορά στην μεταφερσιμότητα του κειμένου XML. Η γλώσσα XML είναι αναγνώσιμη από ανθρώπους και μηχανές και είναι ανεξάρτητη του υλικού ή του λογισμικού του υπολογιστικού συστήματος. Τα κείμενα XML υιοθετούν την κωδικοποίηση κατά Unicode. Ένα επίσης σημαντικό χαρακτηριστικό των κειμένων XML είναι ότι μπορούν να περιγράψουν τόσο ιεραρχικές όσο δικτυακές δομές με απλό τρόπο.

Μειονεκτήματα μιας XML Βάσης Δεδομένων

Από την άλλη πλευρά, ένα κείμενο XML, σαν βάση δεδομένων, παρουσιάζει αρκετά μειονεκτήματα. Καταρχήν, τα στοιχεία ενός κειμένου XML είναι διατεταγμένα, σε αντίθεση με τα λογικά μοντέλα βάσεων δεδομένων (π.χ., το σχεσιακό), όπου η έννοια της διάταξης των δεδομένων δεν υφίσταται.

Ένα δεύτερο μειονέκτημα των κειμένων XML πηγάζει στις συντακτικές ιδιαιτερότητες της γλώσσας τους. Συγκεκριμένα, στην XML, σε αντίθεση με τις βάσεις δεδομένων, επιτρέπεται η ανάμειξη σχολίων και στοιχείων σε ένα κείμενο.

Επιπλέον, η γλώσσα XML επιτρέπει την υιοθέτηση διαφορετικών δομικών μονάδων για την έκφραση της αυτής κατάστασης. Συγκεκριμένα, η καταγραφή δεδομένων σε ένα κείμενο XML μπορεί να γίνει με εναλλακτική χρήση γνωρισμάτων ή στοιχείων.

Όπως αναφέρθηκε στην προηγούμενη Ενότητα, η γλώσσα XML περιβάλλεται από μια συλλογή εργαλείων. Ορισμένα από τα εργαλεία αυτά έχουν σαν στόχο τη διαχείριση των δεδομένων, που φιλοξενούνται σε κείμενα XML. Επομένως, καθώς ένα κείμενο XML αντιμετωπίζεται σαν μια βάση δεδομένων, τα παραπάνω εργαλεία μπορεί να θεωρηθεί ότι συνθέτουν το αντίστοιχο Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ). Στη συνέχεια, περιγράφονται συνοπτικά οι διαθέσιμοι μηχανισμοί για την οργάνωση (δόμηση) των δεδομένων σε ένα κείμενο XML, και την διαχείρισή τους (γλώσσες ερωτήσεων).

3.2.6 Πλεονεκτήματα και Μειονεκτήματα της χρήσης XML

Η XML είναι σήμερα μία βασική γλώσσα για την ανταλλαγή δεδομένων μεταξύ προγραμμάτων, υπολογιστικών συστημάτων, στο διαδίκτυο ακόμα και στις βάσεις δεδομένων.

Οι βασικοί παράγοντες επιτυχίας της είναι οι εξής:

- Υποστηρίζει το Unicode, επιτρέποντας την επικοινωνία σχεδόν οποιασδήποτε πληροφορίας σε οποιαδήποτε γραπτή ανθρώπινη γλώσσα.
- Μπορεί να αντιπροσωπεύει κοινές δομές δεδομένων για τις επιστήμες υπολογιστών: εγγραφές, λίστες και δέντρα.
- Η μορφή του με αυτο-τεκμηρίωση περιγράφει τη δομή και τα ονόματα πεδίων, καθώς και συγκεκριμένες τιμές.
- Οι αυστηρές απαιτήσεις σύνταξης και επεξεργασίας καθιστούν τους απαραίτητους αλγόριθμους ανάλυσης εξαιρετικά απλοί, αποτελεσματικοί και συνεπείς.
- Η XML χρησιμοποιείται έντονα ως μορφή αποθήκευσης και επεξεργασίας εγγράφων, τόσο σε απευθείας σύνδεση όσο και εκτός σύνδεσης.
- Βασίζεται στα διεθνή πρότυπα.
- Μπορεί να ενημερωθεί σταδιακά.
- Επιτρέπει την επικύρωση χρησιμοποιώντας γλώσσες σχήματος όπως XSD και Schema, οι οποίες καθιστούν ευκολότερη την κατασκευή λογισμικού.
- Η ιεραρχική δομή είναι κατάλληλη για τα περισσότερα (αλλά όχι όλα) είδη εγγράφων.
- Είναι ανεξάρτητο από την πλατφόρμα, επομένως είναι σχετικά ανθεκτικό στις αλλαγές της τεχνολογίας.
- Η συμβατότητα προς τα εμπρός και προς τα πίσω είναι σχετικά εύκολη, παρά τις αλλαγές στο DTD ή στο Schema.

Υπάρχουν όμως και μειονεκτήματα στην χρήση της XML και μερικά από αυτά είναι τα εξής:

- Η σύνταξη XML είναι περιττή ή μεγάλη σε σχέση με δυαδικές αναπαραστάσεις παρόμοιων δεδομένων, ειδικά με πίνακες δεδομένων.

- Ο πλεονασμός μπορεί να επηρεάσει την αποδοτικότητα της εφαρμογής μέσω υψηλότερων δαπανών αποθήκευσης, μετάδοσης και επεξεργασίας
- Η σύνταξη XML είναι λεπτομερή, ειδικά για τους χρήστες, σε σχέση με άλλες εναλλακτικές μορφές μετάδοσης δεδομένων που βασίζονται σε κείμενο.
- Το ιεραρχικό μοντέλο για την αναπαράσταση είναι περιορισμένο σε σύγκριση με ένα γραφικό προσανατολισμό αντικειμένων
- Η εκδήλωση αλληλεπικαλυπτόμενων (μη ιεραρχικών) σχέσεων κόμβου απαιτεί επιπλέον προσπάθεια.
- Η διάκριση μεταξύ περιεχομένου και χαρακτηριστικών σε XML φαίνεται να είναι αφύσικη για κάποιους και κάνει το σχεδιασμό δομών δεδομένων XML πιο δύσκολο.
- Οι μετασχηματισμοί, ακόμα και οι μετασχηματισμοί ταυτότητας, έχουν ως αποτέλεσμα αλλαγές στη μορφή (κενό, ταξινόμηση χαρακτηριστικών, παραπομπή χαρακτηριστικών, κενά γύρω από τα χαρακτηριστικά, νέες γραμμές). Αυτά τα προβλήματα μπορούν να κάνουν πολύ δύσκολη τη διάχυση της πηγής XML.
- Ενθαρρύνει μη σχεσιακές δομές δεδομένων (μη κανονικοποιημένα δεδομένα)

3.3 Η γλώσσα προγραμματισμού PHP

Εισαγωγή

Η PHP που σημαίνει Hypertext Preprocessor³⁹, είναι μία ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, scripting γλώσσα προγραμματισμού η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Διαδίκτυο και μπορεί να ενσωματωθεί και στην γλώσσα σήμανσης HTML. Είναι ιδανική γλώσσα προγραμματισμού για την δημιουργία ιστοτόπων με δυναμικό περιεχόμενο, καθώς μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό server του Παγκόσμιου Ιστού (όπως έχουμε αναφέρει και παραπάνω ένας τέτοιος είναι ο Apache), ώστε το τελικό περιεχόμενο να παραχθεί σε πραγματικό χρόνο. Το τελικό αυτό περιεχόμενο θα σταλεί είτε στον φυλλομετρητή του χρήστη σε μορφή κώδικα HTML είτε θα μπορεί τα αποτελέσματα να σταλούν κρυφά σε κάποιο άλλο PHP script.

Ιστορία της PHP

Η PHP γλώσσα δημιουργήθηκε από τον Rasmus Lerdorf το 1994 και κυκλοφόρησε δημοσίως τον Ιούνιο του 1995. Βασίστηκε στην γλώσσα προγραμματισμού C και δημιούργησε το πρώτο του script με όνομα php.cgi για προσωπική του χρήση. Αφορμή ήταν ότι ήθελε να κρατάει μία λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Το script αυτό πήρε διαστάσεις όταν άρχισε να το δίνει στους φίλους του, οι οποίοι αργότερα του ζητούσαν να το εμπλουτίσει με περισσότερες δυνατότητες. Η γλώσσα τότε είχε το όνομα PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997, δηλαδή μετά από δύο χρόνια, εισήχθη ως PHP / FI 2.0 και έναν χρόνο αργότερα δύο άλλοι

³⁹ PHP: Hypertext Preprocessor, Wikipedia, <https://el.wikipedia.org/wiki/PHP>

προγραμματιστές, ο Zeev Suraski και ο Andi Gutmans, ξανάγραψαν τη βάση της αρχικής έκδοσης και ξεκίνησαν την PHP 3⁴⁰.

Δύο δεκαετίες μετά την έναρξή της, η PHP σημείωσε μια πρωτοφανή ανάπτυξη και εξακολουθεί να είναι ισχυρή. Σήμερα ελέγχει πάνω από το 80% όλων των ιστοσελίδων του πλανήτη. Αυτό περιλαμβάνει μεγάλες εταιρείες μεταξύ άλλων το Facebook, το Wikipedia και το WordPress.

3.3.1 Περιγραφή εγγράφου PHP

Χρήση PHP και HTML

Η PHP και η HTML αλληλοεπιδρούν πολύ μεταξύ τους, διότι η PHP μπορεί να δημιουργήσει ένα HTML αρχείο και η HTML μπορεί να μεταφέρει πληροφορίες στην PHP. Για παράδειγμα, όταν δημιουργείτε μια πολύπλοκη σελίδα, σε κάποιο σημείο ερχόμαστε αντιμέτωποι με την ανάγκη συνδυασμού της PHP γλώσσας με την HTML για να έχουμε τα αποτελέσματα που χρειαζόμαστε. Σε πρώτη φάση αυτό μπορεί να φαίνεται πολύπλοκο, αφού η PHP και η HTML είναι δύο τελείως ξεχωριστές γλώσσες, αλλά αυτό δεν συμβαίνει. Η PHP έχει σχεδιαστεί έτσι ώστε να μπορεί να αλληλοεπιδρά με την HTML και τα PHP scripts να μπορούν να συμπεριληφθούν σε μια σελίδα HTML χωρίς κανένα πρόβλημα.

Σε μια σελίδα HTML, ο κώδικας PHP περικλείεται μέσα σε ειδικές ετικέτες PHP. Όταν ο επισκέπτης ανοίξει την σελίδα, ο server επεξεργάζεται τον κώδικα PHP και στην συνέχεια αποστέλλει την έξοδο, και όχι τον ίδιο τον κώδικα PHP) στον φυλλομετρητή του χρήστη. Στην πραγματικότητα είναι πολύ απλό να ενσωματώσουμε HTML και PHP. Ένα PHP script μπορεί να αντιμετωπιστεί ως μία HTML σελίδα, με κομμάτια PHP εισαγόμενα από εδώ και από εκεί στον κώδικα. Οτιδήποτε σε ένα PHP script που δεν συμπεριλαμβάνεται σε `<?php..... ?>` ετικέτες αγνοούνται από τον PHP μεταγλωττιστή (compiler) και μεταφέρεται απευθείας στον φυλλομετρητή⁴¹.

Σύνταξη

```
<html>
  <head>
    <title>Αυτό είναι ένα παράδειγμα!</title>
  </head>
  <body>
    <p>Παράδειγμα σωστής δομής ενός html αρχείου.</p>
    <?php echo "Hi, I'm a PHP script!"; ?>
  </body>
</html>
```

Η γλώσσα PHP χρησιμοποιεί τα σύμβολα `<?php.... ?>` για να μπορέσει να εισάγει ένα PHPscript μέσα σε μία html σελίδα.

Στο συγκεκριμένο παράδειγμα στην σελίδα θα εμφανίσει αυτό εδώ στην οθόνη μας:

⁴⁰ PHP: History of PHP – Manual, PHP, <http://php.net/manual/en/history.php.php>

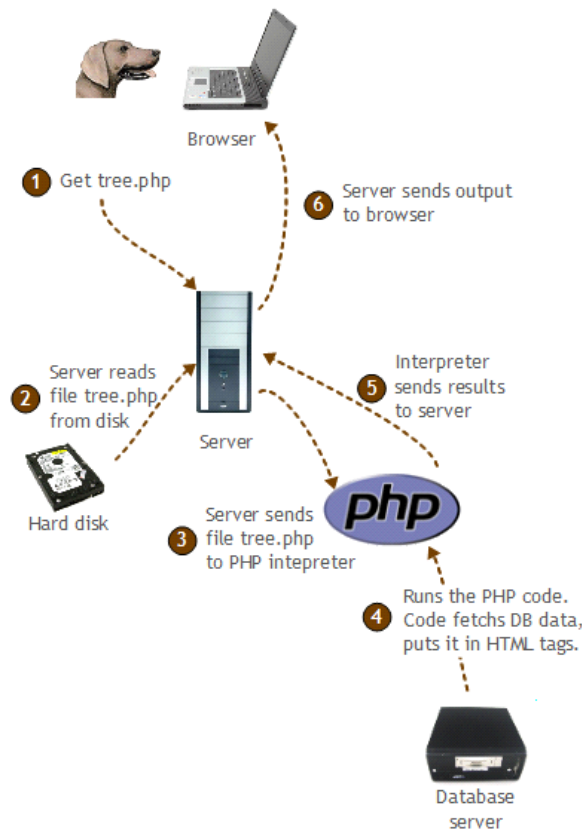
⁴¹ PHP and HTML, PHP, <http://php.net/manual/en/faq.html.php>

Παράδειγμα σωστής δομής ενός html αρχείου.
Hi, I'm a PHP script!

Όπως κάθε γλώσσα προγραμματισμού έτσι και η PHP:

- Έχει εντολές για να εμφανίζει στην οθόνη, όπως είναι το echo και print.
- Έχει μεταβλητές που δηλώνονται με το \$
- Έχει δομές ελέγχου όπως είναι η if, if...else και η elseif
- Έχει δομές επανάληψης όπως είναι η for, foreach, while και do...while
- Μπορεί να γίνει χρήση πινάκων
- Δημιουργία συναρτήσεων
- Και πολλά άλλα!

Τρόπος Λειτουργίας ενός PHP αρχείου



Εικόνα 6 - Τρόπος Λειτουργίας ενός PHP αρχείου

Με βάση την παραπάνω εικόνα θα αναλύσουμε τον τρόπο λειτουργίας ενός php αρχείου κατά την διάρκεια ζήτησης της σε μία σελίδα HTML. Έχουμε:

1. Ο χρήστης, πληκτρολογεί την διεύθυνση URL της σελίδας μέσω ενός φυλλομετρητή
2. Ο φυλλομετρητής μεταφέρει το αίτημα στον server
3. Ο οποίος με την σειρά του επεξεργάζεται το αίτημα: άνοιξε το αρχείο tree.php και ανοίγει να διαβάσει το αρχείο tree.php που βρίσκεται στον σκληρό δίσκο
4. Επόμενο βήμα, είναι ο server να στείλει το αρχείο tree.php στον PHP διερμηνέα,

5. Ο οποίος έπειτα εκτελεί το σημείο του κώδικα που πρέπει, ζητάει δεδομένα, μέσω της βάσης δεδομένων και μετά αυτά τα δεδομένα τα εισάγει σε ετικέτες HTML.
6. Και ο διερμηνέας PHP στέλνει πίσω στον server τα αποτελέσματα της εκτέλεσής του,
7. Ο οποίος server τα εμφανίζει στον φυλλομετρητή του χρήστη.

3.3.2 Πλεονεκτήματα και Μειονεκτήματα της γλώσσα προγραμματισμού PHP

Πλεονεκτήματα σε σχέση με την HTML

Μερικά από τα πλεονεκτήματα χρήσης της PHP γλώσσας σε σχέση με την HTML είναι τα εξής:

- Η HTML είναι στατική γλώσσα που σημαίνει ότι τα δεδομένα της δεν μπορούν να αλλάξουν ενώ η PHP, αντίθετα, είναι δυναμική γλώσσα. Επομένως τα δεδομένα του ιστοτόπου μπορούν να ανανεώνονται και να αλλάζουν με εντολή του χρήστη.
- Η HTML δεν είναι ασφαλής για ιστοτόπους που χρειάζονται ασφάλεια στα δεδομένα τους. Πολλοί ιστότοποι όπως είναι και τα e-shop, χρησιμοποιούν προσωπικά δεδομένα και τραπεζικά δεδομένα που χρειάζονται ασφάλεια.
- Η PHP γλώσσα μπορεί να συνδεθεί με μία βάση δεδομένων και να αντλήσει από εκεί τις πληροφορίες της.

Γενικά πλεονεκτήματα της PHP γλώσσας

Μερικά από τα πλεονεκτήματα της PHP γλώσσας είναι τα εξής:

- Ανοιχτού κώδικα: Αναπτύσσεται και συντηρείται από μια μεγάλη ομάδα προγραμματιστών της PHP, κάτι που βοηθά στη δημιουργία μιας μεγάλης κοινότητας υποστήριξης και σε μια πλούσια βιβλιοθήκη επέκτασης.
- Ταχύτητα: Είναι σχετικά γρήγορη αφού χρησιμοποιεί πολλούς πόρους συστήματος.
- Εύκολη στη χρήση: Χρησιμοποιεί την γλώσσα προγραμματισμού C σαν σύνταξη, οπότε για εκείνους που είναι εξοικειωμένοι με την C, είναι πολύ εύκολο να δημιουργήσουν εφαρμογές και ιστοσελίδες διαδικτύου.
- Σταθερότητα: Δεδομένου ότι διατηρείται από πολλούς προγραμματιστές, έτσι ώστε όταν εντοπιστούν σφάλματα, μπορούν να διορθωθούν γρήγορα.
- Ισχυρή υποστήριξη βιβλιοθήκης: Μπορείτε εύκολα να βρείτε λειτουργικές μονάδες που χρειάζεστε όπως PDF, Graph κ.α.
- Ενσωματωμένες μονάδες σύνδεσης βάσης δεδομένων: Μπορείτε να συνδεθείτε εύκολα στη βάση δεδομένων χρησιμοποιώντας την PHP, δεδομένου ότι πολλοί ιστότοποι οδηγούν δεδομένα/περιεχόμενο, επομένως θα χρησιμοποιήσουμε συχνά βάση δεδομένων, αυτό θα μειώσει σε μεγάλο βαθμό τον χρόνο ανάπτυξης των εφαρμογών ιστού.
- Μπορεί να εκτελεστεί σε πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows, Linux και Mac, είναι εύκολο για τους χρήστες να βρουν παρόχους υπηρεσιών φιλοξενίας.

Γενικά μειονεκτήματα της PHP γλώσσας

- Ασφάλεια: Δεδομένου ότι είναι ανοιχτό, έτσι ώστε όλοι οι άνθρωποι να μπορούν να δουν τον πηγαίο κώδικα, αν υπάρχουν σφάλματα στον πηγαίο κώδικα, μπορούν να χρησιμοποιηθούν από τους ανθρώπους για να εξερευνήσουν την αδυναμία της PHP.
- Δεν είναι κατάλληλη για μεγάλες εφαρμογές: Είναι δύσκολη στην συντήρηση λόγω του ότι είναι πολυτμηματική γλώσσα
- Αδύνατος τύπος: Η αναγκαστική μετατροπή μπορεί να εκπλήξει απρόσεκτους προγραμματιστές και να οδηγήσει σε απροσδόκητα σφάλματα. Για παράδειγμα, οι συμβολοσειρές "1000" και "1e3" συγκρίνονται ίσες επειδή μετατρέπονται σε αριθμούς κινητής υποδιαστολής.

3.3.3 XML και PHP

Η χρήση XML δεδομένων με PHP είναι εφικτή μέσω κατάλληλου αναλυτή parser .

Η απλή λύση είναι η simpleXML διότι:

- Παρέχει πρόσβαση στα XML στοιχεία του εγγράφου
- Μπορεί να ελέγξει αν το έγγραφο είναι ορθά μορφοποιημένο (well-formed)
- Αλλά συνήθως δεν ελέγχει αν η δομή είναι έγκυρη (valid) (βάσει κάποιου DTD ή XML Schema) και δεν υποστηρίζει namespaces⁴²

Η σύνθετη λύση είναι μέσω του DOM API και παρέχει τα εξής:

- Είναι περισσότερο ολοκληρωμένη λύση,
- Επιτρέπει πιο πολύπλοκους χειρισμούς, αλλά απαιτεί περισσότερο κώδικα⁴³.

Παραδοσιακά τα XML δεδομένα είναι σε κωδικοποίηση UTF-8.

Ελέγξτε τη συμβατότητα των δεδομένων σας και της XML βιβλιοθήκης που θα χρησιμοποιήσετε. Πιθανώς να χρειαστούν μετατροπές (μέσω ειδικών συναρτήσεων).

3.3.4 Η επέκταση simpleXML

Εισαγωγή

Το SimpleXML είναι μια επέκταση της PHP γλώσσας, η οποία εισήχθη στην έκδοση 5.0, που επιτρέπει στους χρήστες να χειρίζονται και να χρησιμοποιούν δεδομένα XML εύκολα. Η επέκταση simpleXML παρέχει έναν εύκολο τρόπο ώστε να μπορούμε να αντλήσουμε τα δεδομένα ενός XML αρχείου, όπως είναι το όνομα ενός στοιχείου element, τα χαρακτηριστικά attributes και το περιεχόμενο του, αρκεί να γνωρίζουμε τη δομή ή τη διάταξη του εγγράφου XML.

⁴² SimpleXML, PHP, <http://www.php.net/manual/en/book.simplexml.php>

⁴³ DOM API, PHP, <http://www.php.net/manual/en/book.dom.php>

Με την χρήση της simpleXML μπορούμε να μετατρέψουμε ένα XML αρχείο σε μία δομή δεδομένων που μπορείτε να μετακινηθείτε μέσα σαν μια συλλογή από πίνακες και αντικείμενα.

Εγκατάσταση

Από την PHP έκδοση 5.0, οι λειτουργίες SimpleXML είναι μέρος του πυρήνα της PHP. Δεν απαιτείται εγκατάσταση για τη χρήση αυτών των λειτουργιών.

Λειτουργίες της simpleXML

Οι λειτουργίες και οι συναρτήσεις που αποτελούν την επέκταση simpleXML είναι οι εξής:

- `__construct()` Δημιουργία ενός νέου simpleXML αντικειμένου
- `addAttribute():` Πρόσθεση ενός χαρακτηριστικού attribute σε ένα SimpleXML στοιχείο
- `addChild():` Πρόσθεση ενός στοιχείου-παιδί σε ένα simpleXML στοιχείο
- `asXML():` Επιστροφή μίας ορθά μορφοποιημένης (well-formed) XML συμβολοσειράς από ένα simpleXML αντικείμενο
- `attributes():` Επιστροφή των χαρακτηριστικών ή των τιμών από ένα στοιχείο
- `children():` Επιστροφή των παιδιών ενός συγκεκριμένου κόμβου
- `count():` Μέτρηση των παιδιών ενός συγκεκριμένου κόμβου
- `getDocNamespaces():` Επιστροφή των χώρων ονομασίας που έχουν χρησιμοποιηθεί στο έγγραφο
- `getName():` Επιστροφή της XML ετικέτας που παραπέμπεται από το simpleXML στοιχείο
- `getNamespaces():` Επιστροφή των χώρων ονομασίας που έχουν χρησιμοποιηθεί στο XML έγγραφο
- `registerXPathNamespace():` Δημιουργία περιεχομένου ενός χώρου ονομασίας για το επόμενο XPath ερώτημα
- `saveXML() ή asXML():` Αποθήκευση των αλλαγών του simpleXML αντικειμένου σαν XML δομή σε ένα αρχείο
- `simplexml_import_dom():` Επιστροφή ενός SimpleXMLElement αντικειμένου από έναν κόμβο DOM
- `simplexml_load_file():` Μετατροπή ενός XML αρχείου σε ένα αντικείμενο SimpleXMLElement
- `simplexml_load_string():` Μετατροπή μίας συμβολοσειράς XML σε ένα αντικείμενο SimpleXMLElement
- `xpath():` Εκτέλεση ενός XPath ερωτήματος σε ένα XML έγγραφο

Συναρτήσεις Επανάληψης:

- `current()`: Επιστροφή του τρέχοντος στοιχείου
- `getChildren()`: Επιστροφή του στοιχείου-παιδιού του τρέχοντος στοιχείου
- `hasChildren()`: Έλεγχος τρέχοντος στοιχείου για το αν έχει παιδιά
- `key()`: Επιστροφή του τρέχοντος κλειδιού
- `next()`: Μετακίνηση στο επόμενο στοιχείο
- `rewind()`: Επιστροφή στο πρώτο στοιχείο
- `valid()`: Έλεγχος τρέχοντος στοιχείου για το αν είναι έγκυρο⁴⁴

3.3.5 Η επέκταση DOM

Το Μοντέλο Αντικειμένου Εγγράφου (DOM Object Model - DOM⁴⁵) είναι ένα τυπικό μοντέλο αντικειμένου ανεξάρτητο από την πλατφόρμα και τη γλώσσα που αντιπροσωπεύει HTML ή XML. Αντιμετωπίζει ένα έγγραφο HTML, XHTML ή XML ως δομή δέντρου όπου κάθε κόμβος είναι ένα αντικείμενο που αντιπροσωπεύει ένα μέρος του εγγράφου. Τα αντικείμενα μπορούν να επεξεργαστούν προγραμματικά και οποιεσδήποτε ορατές αλλαγές συμβαίνουν ως αποτέλεσμα μπορεί στη συνέχεια να αντικατοπτρίζονται στην εμφάνιση του εγγράφου.

Η κύρια τυποποίηση του DOM διεκπεραιώθηκε από την Κοινοπραξία του Παγκόσμιου Ιστού (World Wide Web Consortium), η οποία η τελευταία σύσταση που ανέπτυξε ήταν το 2004. Η WHATWG⁴⁶ ανέλαβε την ανάπτυξη του προτύπου, δημοσιεύοντας το ως ζωντανό έγγραφο. Το W3C δημοσιεύει τώρα σταθερά στιγμιότυπα του προτύπου WHATWG.

Ένα πρόγραμμα περιήγησης στο διαδίκτυο δεν είναι υποχρεωμένο να χρησιμοποιεί το DOM για την απόδοση ενός εγγράφου HTML. Ωστόσο, το DOM απαιτείται από δέσμες ενεργειών JavaScript που επιθυμούν να επιθεωρήσουν ή να τροποποιήσουν μια ιστοσελίδα δυναμικά. Με άλλα λόγια, το μοντέλο αντικειμένου εγγράφου είναι ο τρόπος με τον οποίο η JavaScript βλέπει τη σελίδα HTML που περιέχει και την κατάσταση του προγράμματος περιήγησης.

Επειδή το DOM υποστηρίζει την πλοήγηση σε οποιαδήποτε κατεύθυνση (π.χ. `parent` και `previous siblings`) και επιτρέπει αυθαίρετες τροποποιήσεις, μια εφαρμογή πρέπει τουλάχιστον να απομνημονεύσει το έγγραφο που έχει διαβάσει μέχρι τώρα (ή κάποια αναλυτική μορφή του). Ως εκ τούτου, το DOM είναι πιθανό να είναι το πλέον κατάλληλο για εφαρμογές όπου το έγγραφο πρέπει να έχει πρόσβαση επανειλημμένα ή εκτός σειράς αλληλουχίας.

⁴⁴ PHP 5 SimpleXML Functions, W3Schools, https://www.w3schools.com/php/php_ref_simplexml.asp & SimpleXML Documentation, PHP, <http://us.php.net/manual/en/book.simplexml.php>

⁴⁵ Document Object Model, Wikipedia, https://en.wikipedia.org/wiki/Document_Object_Model

⁴⁶ Web Hypertext Application Technology Working Group (WHATWG), Wikipedia, <https://en.wikipedia.org/wiki/WHATWG>

Το XML DOM είναι σχεδιασμένο για να μπορεί να χρησιμοποιηθεί με οποιαδήποτε γλώσσα προγραμματισμού και με οποιοδήποτε λειτουργικό σύστημα. Με το XML DOM μπορούμε να δημιουργήσουμε ένα XML έγγραφο, να πλοηγηθούμε στη δομή του και να προσθέσουμε, τροποποιήσουμε ή διαγράψουμε τα στοιχεία του.

Εγκατάσταση

Οι λειτουργίες του XML DOM είναι μέρος του πυρήνα της PHP. Δεν απαιτείται εγκατάσταση για τη χρήση αυτών των λειτουργιών.

Λειτουργίες της XML DOM

Μερικές από τις λειτουργίες και τις συναρτήσεις που αποτελούν την επέκταση XML DOM είναι οι εξής:

- `__construct()`: Δημιουργία ενός νέου `DomDocument` αντικειμένου
- `createAttribute()`: Δημιουργία ενός νέου χαρακτηριστικού `attribute`
- `createElement()`: Δημιουργία ενός νέου στοιχείου κόμβου
- `createTextNode()`: Δημιουργία νέου κόμβου κειμένου
- `getElementById()`: Αναζητά ένα στοιχείο με συγκεκριμένο αναγνωριστικό ID
- `getElementsByTagName()`: Αναζητά όλα τα στοιχεία με το δεδομένο όνομα τοπικής ετικέτας
- `importNode()`: Εισαγωγή ενός κόμβου στο τρέχοντος έγγραφο
- `load()`: Φόρτωση XML από ένα αρχείο
- `loadXML()`: Φόρτωση XML από μία συμβολοσειρά
- `save()`: Αποθήκευση των αλλαγών του εσωτερικού δέντρου XML μέσα σε ένα αρχείο
- `saveXML()`: Αποθήκευση των αλλαγών του εσωτερικού δέντρου XML μέσα σε μία συμβολοσειρά
- `schemaValidate()`: Επικύρωση ενός εγγράφου με βάση ένα `schema`
- `schemaValidateSource()`: Επικύρωση ενός εγγράφου με βάση ένα `schema`
- `validate()`: Επικύρωση του εγγράφου με βάση το DTD

Κεφάλαιο 4

Κύριο μέρος Πτυχιακής Εργασίας

4.1 Γενική Περιγραφή

Για την δημιουργία και την διαχείριση μίας δικτυοκεντρικής βάσης δεδομένων για μια ασφαλιστική εταιρεία, για αρχή έπρεπε να κατασκευάσουμε έναν δυναμικό ιστότοπο, με τον οποίο θα μπορούσαμε να κάνουμε την επεξεργασία και την αποθήκευση των δεδομένων μας στην XML βάση δεδομένων μας. Η σύνδεση της XML βάσης δεδομένων μας με την εφαρμογή της ασφαλιστικής εταιρείας που δημιουργήσαμε και όλες οι λειτουργίες της, υλοποιήθηκαν με την γλώσσα προγραμματισμού PHP και την χρήση των επεκτάσεων simpleXML και XML DOM.

Πρόσβαση στην εφαρμογή έχουν όλοι οι υπάλληλοι της εταιρείας, κάνοντας χρήση των προσωπικών κωδικών εισαγωγής τους στο σύστημα. Το σύστημα μας, παρέχει έναν ιστότοπο για τους πελάτες έτσι ώστε να μπορούν να ενημερώνονται για τα καινούργια πακέτα ασφαλιστρών και παρέχει και έναν ιστότοπο/εφαρμογή για τους υπαλλήλους της ασφαλιστικής εταιρείας όπου εκεί είναι όλες οι λειτουργίες της εταιρείας.

Δεδομένα αποθήκευσης στην Βάση Δεδομένων μας

Αναλυτικά η εφαρμογή μας αποθηκεύει πληροφορίες για όλα τα παρακάτω:

- Πληροφορίες για τα διάφορα τμήματα της εταιρείας.
- Πληροφορίες για τους πελάτες.
- Πληροφορίες για τις κατηγορίες των αυτοκινήτων.
- Πληροφορίες για τα ασφαλιστήρια συμβόλαια.
- Πληροφορίες για τα αυτοκίνητα.
- Πληροφορίες για τα ατυχήματα που συμβαίνουν με ασφαλισμένα αυτοκίνητα
- Πληροφορίες για τους υπαλλήλους της εταιρείας.

Και πιο αναλυτικά:

- Για κάθε πελάτη αποθηκεύουμε το ονοματεπώνυμο, την διεύθυνση, την ηλικία, το τηλέφωνο του, την ημερομηνία απόκτησης του διπλώματος, τον αριθμό ταυτότητας.
- Για κάθε τμήμα αποθηκεύουμε το όνομα του το οποίο είναι μοναδικό, την τοποθεσία που είναι τα γραφεία του.
- Για κάθε υπάλληλο αποθηκεύουμε το ονοματεπώνυμο, την διεύθυνση, το τηλέφωνο του.
- Για κάθε ατύχημα αποθηκεύουμε ένα κωδικό που είναι μοναδικό, την ημερομηνία και ώρα που έγινε το ατύχημα, την διεύθυνση.
- Για κάθε κατηγορία αποθηκεύουμε το εύρος των κυβικών που καλύπτει, το αν απευθύνεται σε επιβατικά ή εμπορικά και το ύψος των ασφαλιστρών αν εξάμηνο.

- Για κάθε αυτοκίνητων αποθηκεύουμε πληροφορία για το ποιος είναι ο αριθμός κυκλοφορία, ποια είναι η ημερομηνία ταξινόμησης του.
- Για κάθε ασφαλιστήριο αποθηκεύουμε τον κωδικό του, την ημερομηνία που γίνεται και το ποσό.

Επιπλέον:

- Κάθε αυτοκίνητο έχει ένα πελάτη ως οδηγό. Ένας πελάτης μπορεί να έχει πολλά αυτοκίνητα.
- Κάθε ατύχημα συνδέεται με τουλάχιστον ένα συγκεκριμένο αυτοκίνητο.
- Επίσης για κάθε ατύχημα αποθηκεύουμε το κόστος της ζημίας που υπέστη το κάθε αυτοκίνητο. Αν στο ατύχημα έχει εμπλακεί αυτοκίνητο που δεν ανήκει στα ασφαλισμένα της εταιρίας πρέπει να αποθηκεύουμε και γι' αυτό την ζημιά που υπέστη.
- Για κάθε ατύχημα είναι υπεύθυνος ένας και μόνο ένας υπάλληλος.
- Κάθε αυτοκίνητο ανήκει σε μια και μόνο μια κατηγορία αυτοκινήτων.
- Ένα ατύχημα συνδέεται με τουλάχιστον ένα πελάτη. Πρέπει να αποθηκεύουμε πληροφορία αν έφταιγε ή όχι ο πελάτης.
- Κάθε υπάλληλος δουλεύει σε ένα τμήμα. Κάθε τμήμα μπορεί να έχει πολλούς υπαλλήλους.
- Κάθε ασφαλιστήριο αντιστοιχεί σε ένα και μόνο αυτοκίνητο και ένα και μόνο πελάτη και σε ένα και μόνο υπάλληλο.
- Σε κάθε τμήμα μόνο ένας υπάλληλος είναι διευθυντής. Κάθε τμήμα έχει οπωσδήποτε διευθυντή.

Διαδικασίες που πρέπει να υποστηρίζονται:

- Εισαγωγή και διαγραφή για όλες τις κατηγορίες που αναφέρθηκαν πιο πάνω.
- Με την εισαγωγή ενός αυτοκινήτου η αντιστοίχιση στην κατηγορία πρέπει να γίνεται αυτόματα.
- Τα ασφάλιστρα που πληρώνει κάθε πελάτης διαμορφώνονται ως εξής: την πρώτη φορά είναι όσα ορίζει η κατηγορία του αυτοκινήτου που έχει, εκτός αν είναι νέος οδηγός οπότε έχει 20% αύξηση. Αν περάσει ένας χρόνος χωρίς ατύχημα έχει 10% μείωση (αυτό συνεχίζεται μέχρι το 30% λιγότερο από το κανονικό). Αν κάνει ατύχημα και έχει την ευθύνη αυτός έχει αύξηση 30%. Αυτή η αύξηση ισχύει για κάθε ατύχημα. Αν περάσει χρόνος χωρίς να προκαλέσει άλλο ατύχημα έχει μείωση 10% (αυτό συνεχίζεται μέχρι το 30% κάτω του κανονικού). Αν ο πελάτης έχει δεύτερο αυτοκίνητο έχει επιπλέον 5% μείωση του ασφαλίστρου που θα πλήρωνε σε όλα τα αυτοκίνητα. Αν προκαλέσει ατύχημα αυτή η μείωση εξαλείφεται. Θέλομε όταν δημιουργείται ένα ασφάλιστρο να γίνεται αυτόματος υπολογισμός του ασφαλίστρου.

Συγκεντρωτικές Αναφορές

- Συγκεντρωτική αναφορά για τα έσοδα/έξοδα κάθε μήνα. Τα έξοδα προκύπτουν όταν γίνει ατύχημα και φταίει ο πελάτης τότε πρέπει να πληρώσει την ζημιά στο άλλο αυτοκίνητο.

- Συγκεντρωτική αναφορά για τα έσοδα/έξοδα κάθε μήνα αν κατηγορία αυτοκινήτων, αν φύλλο πελατών και αν ηλικία (πχ. 20-30, 31-40 κτλ)
- Συγκεντρωτική αναφορά τα ατυχήματα αν φύλλο πελατών και αν ηλικία (πχ. 20-30, 31-40 κτλ)
- Συγκεντρωτική αναφορά για τα συμβόλαια και τις εισπράξεις που κάνει κάθε υπάλληλος/τμήμα σε ένα συγκεκριμένο χρονικό διάστημα.

4.2 Διαδικασία Ανάπτυξης

Για την υλοποίηση της εφαρμογής μας απαραίτητη είναι η εγκατάσταση ενός local web server, στον οποίο θα κατασκευάσουμε το σύστημα μας. Καλύτερη λύση θεωρήσαμε τον Apache, ο οποίος διατίθεται δωρεάν.

Εμείς εγκαταστήσαμε το XAMPP, το οποίο είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache και ένα διεργαστή για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP. Και έπειτα γράψαμε τον κώδικα με την βοήθεια του Sublime Text Editor, ο οποίος είναι ένα εξελιγμένο πρόγραμμα επεξεργασίας κειμένου για κώδικα όπως είναι PHP και markup γλώσσες όπως είναι HTML και CSS. Όπως επίσης και τα αρχεία xml για την βάση δεδομένων μας.

Μετά την εγκατάσταση του XAMPP προχωρήσαμε στην δημιουργία των αρχείων για τον ιστότοπο μας, στον φάκελο htdocs. Στον φάκελο δηλαδή που αναγνωρίζει ο localhost για το τρέξιμο της εφαρμογής μας.

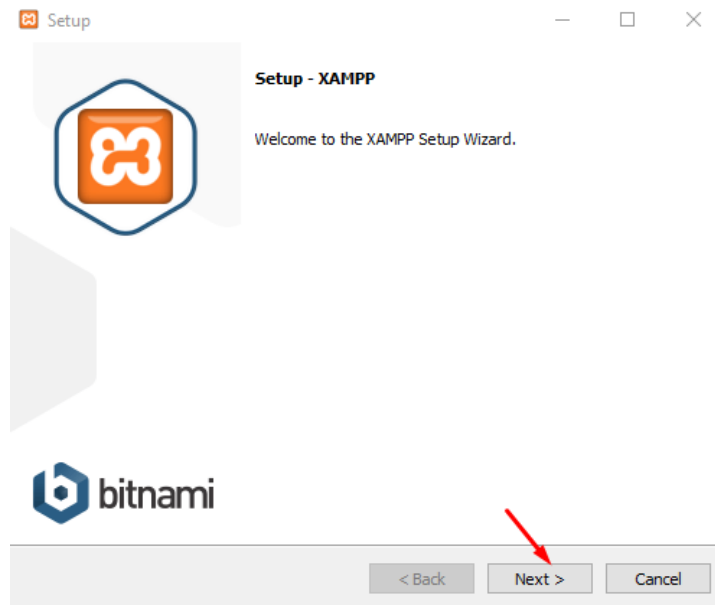
Η εφαρμογή μας αποτελείται από αρχεία PHP και XML. Πάνω στην γλώσσα PHP βασίστηκε όλο το σύστημα μας για την δημιουργία του δυναμικού μας ιστοτόπου και των φορμών που χρησιμοποιήθηκαν για την αποθήκευση και επεξεργασία των δεδομένων μας στα XML αρχεία. Τα XML αυτά αρχεία υλοποιούν την βάση δεδομένων του ασφαλιστικού μας συστήματος.

4.3 Εγκατάσταση Προγράμματος XAMPP

Το Xampp όπως αναφερθήκαμε και πιο πριν είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache.

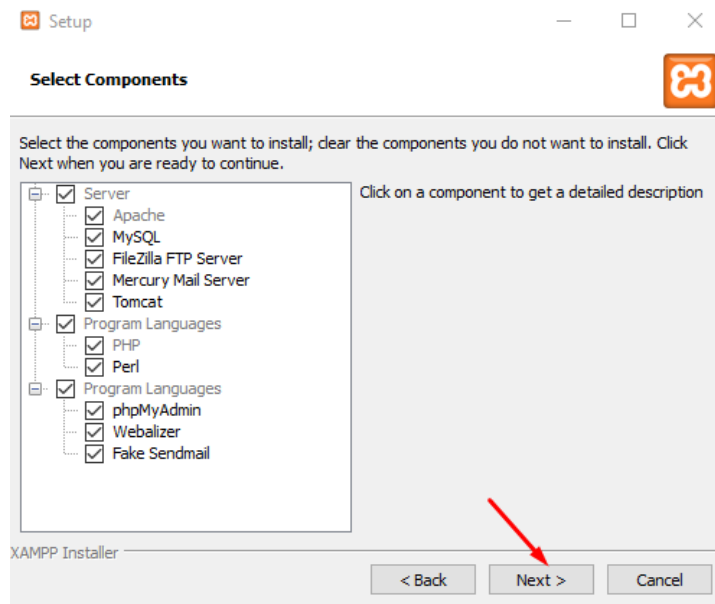
Το Xampp το κατεβάσαμε δωρεάν από: <https://www.apachefriends.org/download.html>. Και έπειτα αφού κατεβάσαμε την έκδοση ανάλογα το λειτουργικό μας σύστημα, προχωρήσαμε στην εγκατάσταση του apache.

Αφού ολοκληρώθηκε η λήψη, εκτελούμε το αρχείο xampp-win32-7.2.2-0-VC15-installer.exe που αποθηκεύτηκε στις λήψεις. Οπότε αναλυτικά προχωρήσαμε στα παρακάτω βήματα για την ολοκλήρωση της εγκατάστασης.



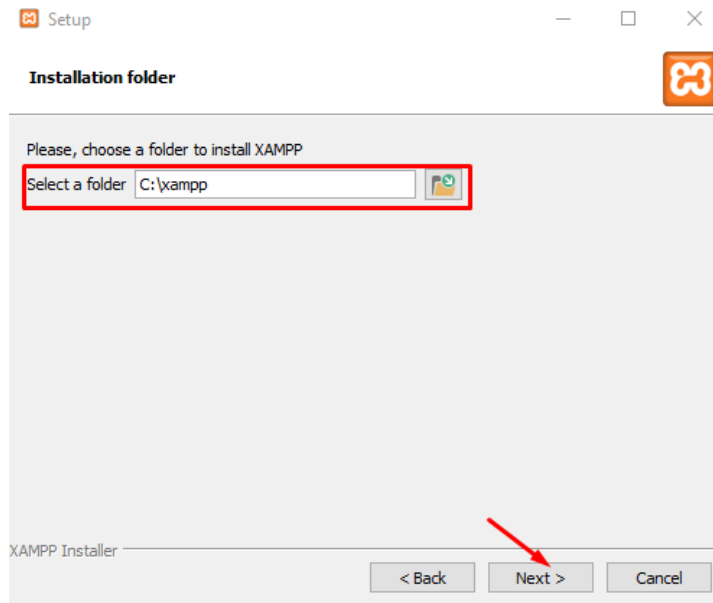
Εικόνα 7 - Εγκατάσταση XAMPP – Εκκίνηση

Μόλις ανοίξει ο διάλογος εγκατάστασης του προγράμματος πατάμε το κουμπί «Next» για να συνεχίσουμε.



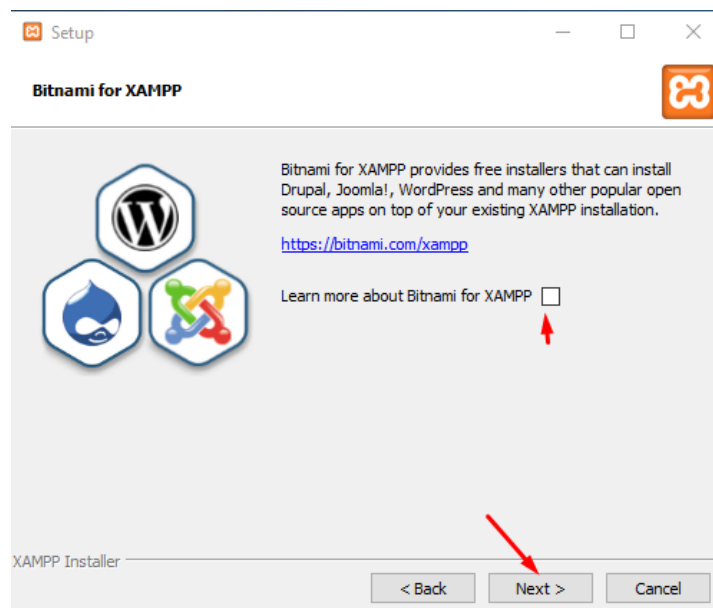
Εικόνα 8 - Επιλογή των λειτουργιών που θέλουμε να εγκαταστήσουμε

Έπειτα επιλέγουμε τι λειτουργίες θέλουμε να εγκαταστήσουμε, εκτός από τον Apache, ο οποίος είναι προεπιλεγμένος και είναι απενεργοποιημένη η επιλογή του. Αφού επιλέξουμε τι θέλουμε πατάμε «Next».



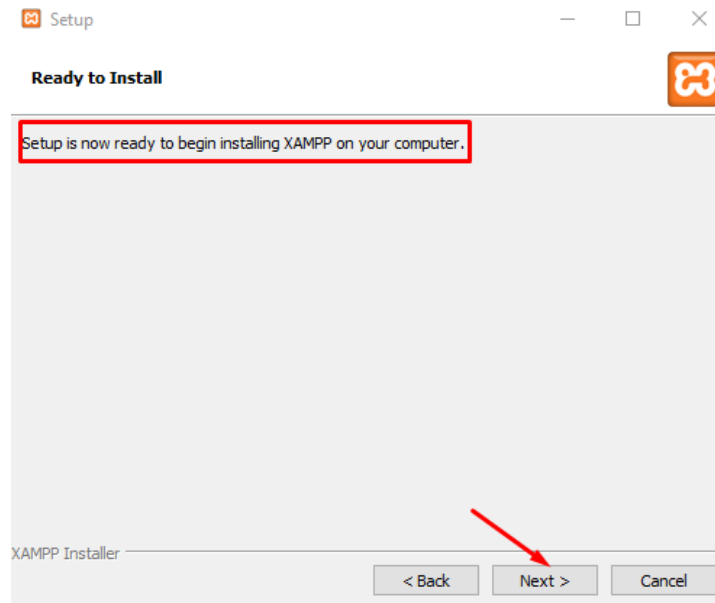
Εικόνα 9 - Επιλογή φακέλου εγκατάστασης του Xampp

Έπειτα, μας δίνεται η δυνατότητα να επιλέξουμε εμείς που θέλουμε να εγκατασταθεί το πρόγραμμα Xampp. Προεπιλεγμένη θέση είναι στον σκληρό δίσκο C, σε ένα δικό του φάκελο Xampp. Αφού επιλέξουμε πατάμε «Next»



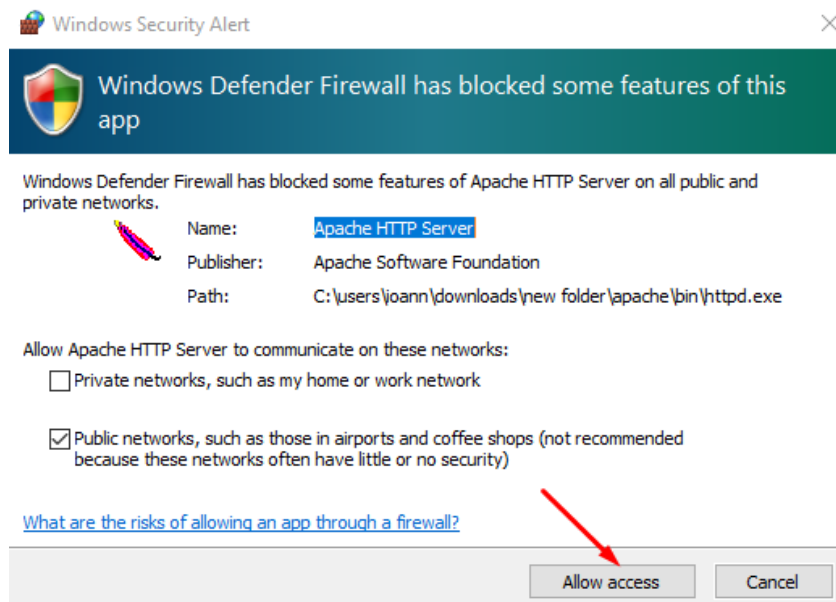
Εικόνα 10 - Bitnami for Xampp

Επίσης, πλέον με την εγκατάσταση του Xampp και αμέσως πριν προχωρήσουμε στην τελική εγκατάσταση του, μας ενημερώνουν ότι μπορούμε να εγκαταστήσουμε τα πολύ γνωστά CSM, Wordpress, Joomla και Drupal. Όποιος θέλει να μάθει περισσότερα, αρκεί να επιλέξει το κουτάκι. Αν όχι, απλά πατάει «Next»



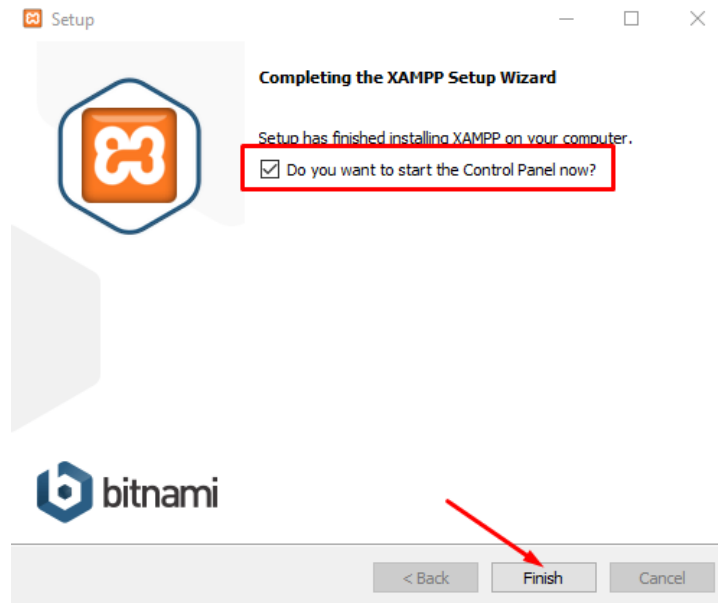
Εικόνα 11 - Ένα κλικ πριν την εγκατάσταση του Xampp

Εάν είμαστε σίγουροι για τις προηγούμενες επιλογές μας και δεν θέλουμε να αναιρέσουμε κάποια από αυτές, απλά πατάμε «Next»



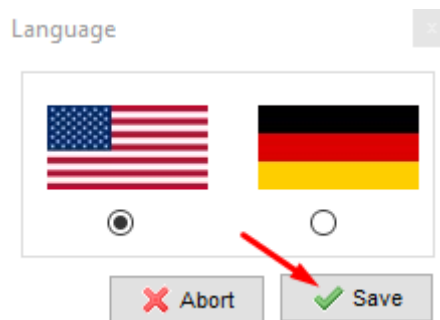
Εικόνα 12 - Installation of Xampp / Allow Access on Windows Firewall

Για να μπορέσει να γίνει η εγκατάσταση του Xampp, πρέπει να του δώσουμε πρόσβαση στο Τείχος Προστασίας των Windows. Απλά πατάμε «Allow access»



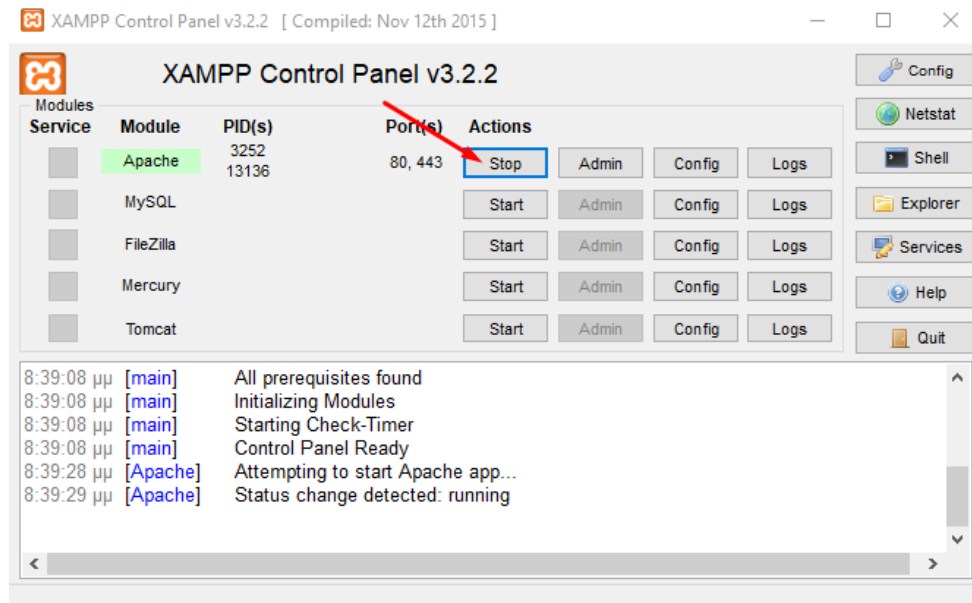
Εικόνα 13 - Ολοκλήρωση εγκατάστασης Xampp και τρέξιμο προγράμματος

Αφού όλα πάνε καλά το τελικά βήμα είναι να κάνουμε εκκίνηση της εφαρμογής, οπότε επιλέγουμε το «Do you want to start Control Panel now?» και πατάμε «Finish».



Εικόνα 14 - Επιλογή γλώσσας για το Xampp

Όταν πρώτο εκτελεστεί, ζητάει γλώσσα. Το Xampp μέχρι στιγμής δεν υποστηρίζει τα Ελληνικά. Οι μόνες γλώσσες που έχει είναι Αγγλικά και Γερμανικά. Επιλέγουμε Αγγλικά και πατάμε «Save»

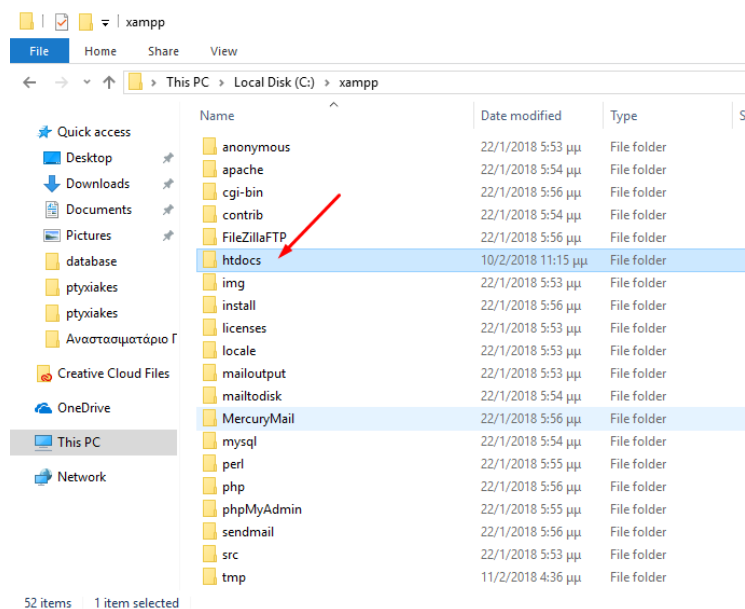


Εικόνα 15 - Εκκίνηση του Apache Server – Xampp

Αφού επιλέξαμε και γλώσσα εμφανίστηκε ο πίνακας ελέγχου του Xampp όπου μπορούμε να κάνουμε εκκίνηση είτε στον Apache, είτε στην βάση δεδομένων MySQL κτλ.

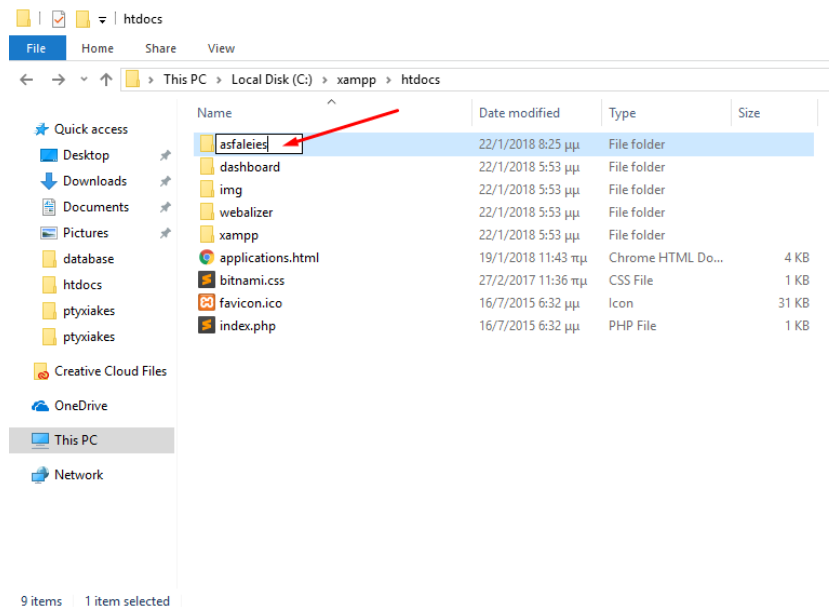
Εμείς χρειαζόμαστε μόνο την εκκίνηση του Apache, οπότε πατήσαμε «Start».

Αν όλα είναι εντάξει, ο Apache παίρνει χρώμα πράσινο και την θέση του Start έχει πάρει το Stop.



Εικόνα 16 - Εύρεση φακέλου htdocs στον φάκελο Xampp

Αφού τρέξαμε τον Server μας, είμαστε έτοιμοι να πάμε στα αρχεία του Xampp στον σκληρό δίσκο C, και να ψάξουμε ανάμεσα στους φακέλους του να βρούμε τον φάκελο htdocs. Ο φάκελος htdocs, είναι ο φάκελος όπου θα αποθηκεύουμε τα αρχεία της εφαρμογής μας.



Εικόνα 17 - Δημιουργία φακέλου εργασία asfaleies Xampp

Ο φάκελος htdocs είναι ο φάκελος του server που αναγνωρίζει και τα τρέχει. Οπότε το επόμενο μας βήμα είναι να δημιουργήσουμε τον δικό μας φάκελο, όπου θα έχουμε τα αρχεία της εργασίας μας, και τον φάκελο αυτόν τον ονομάσαμε asfaleies.

Αν και ο φάκελος είναι κενός, εμείς μπορούμε να τον τρέξουμε πληκτρολογώντας στον φυλλομετρητή μας τον παρακάτω σύνδεσμο (χωρίς την χρήση του http://):

localhost/asfaleies/

4.4 Διαδικασία Υλοποίηση της Βάσης Δεδομένων μας

Σε αυτή την ενότητα θα αναφερθούμε στην διαδικασία της δημιουργίας της Βάσης Δεδομένων μας με την χρήση της γλώσσας XML. Η διαδικασία υλοποίησης ήταν ίδια σχεδόν ίδια σε όλα τα σημεία του κώδικα οπότε στην συνέχεια θα παραθέσουμε ένα παράδειγμα αποθήκευσης, επεξεργασίας και διαγραφής ενός στοιχείου από την βάση μας.

Η κύρια γλώσσα προγραμματισμού που χρησιμοποιήσαμε είναι η γλώσσα PHP και με την βοήθεια των επεκτάσεων της, από την έκδοση 5.0 και μετά, της simpleXML και του DOM (αναφερθήκαμε σε αυτά αναλυτικά στο Κεφάλαιο 3), καταφέραμε να έχουμε μία πολύ λειτουργική βάση δεδομένων που δεν έχει να ζηλέψει σε τίποτα την MySQL.

Στο παράδειγμα που θα ακολουθήσει θα δημιουργήσουμε ενός νέου Τμήματος και στην συνέχεια θα το επεξεργαστούμε και θα το διαγράψουμε. Με σκοπό να αναφερθούμε στις συναρτήσεις της simpleXML και του DOM που χρησιμοποιήσαμε.

Εδώ θα βρείτε όλα τα Τμήματα του Car Insurance και την τοποθεσία τους!

+ ✎ ✕

Εμφάνιση: 100 Αναζήτηση:

εγγραφών

Id	Όνομα Τμήματος	Τοποθεσία	Υπεύθυνος Τμήματος
1	Πληροφορικής	Ηράκλειο	Χρυσάνθη Παναγιωτάκη
2	Ασφαλειών	Ηράκλειο	Γιώργος Ποντικάκης
3	Λογιστήριο	Ηράκλειο	Σοφία Ντελλάκη
4	Γραμματεία	Ηράκλειο	Μάριος Κωστάκης
5	Αποζημιώσεων	Ηράκλειο	Ελένη Βλαχάκη
6	Προσωπικού	Ηράκλειο	Αθηνά Πιπεράκη
7	Marketing	Ηράκλειο	Κώστας Βρανάς

Εμφανίζονται 7 από τις 7 εγγραφές Προηγούμενο 1 Επόμενο

Εικόνα 18 - Όλες οι εγγραφές για τα Τμήματα

4.4.1 Εισαγωγή νέας εγγραφής στην Βάση Δεδομένων

Εισαγωγή καινούργιου Τμήματος!

Κωδικός Τμήματος

Όνομα Τμήματος

Τοποθεσία Τμήματος

+ Επιλέξτε:
Επιλογή Διευθυντή Τμήματος

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Εικόνα 19 - Φόρμα Εισαγωγής νέας εγγραφής Τμήματος εταιρείας

Στην «Εικόνα 18» βλέπουμε την φόρμα εισαγωγής ενός καινούργιου Τμήματος. Τα στοιχεία που καλούμαστε να συμπληρώσουμε για το Τμήμα είναι τα εξής:

- Κωδικός Τμήματος
- Όνομα Τμήματος
- Τοποθεσία Τμήματος
- Και τον Υπεύθυνο Διευθυντή του τμήματος αυτού (drop down μενού, που έχει όλους τους υπαλλήλους της ασφαλιστικής εταιρείας που δεν είναι ήδη Διευθυντές σε κάποιο τμήμα)

Tmimata.xml – Το αρχείο που περιέχει όλα τα στοιχεία για τα Τμήματα

Όλη η βάση μας αποτελείται από τα αρχεία xml της. Το κάθε αρχείο xml το έχουμε ονομάσει έτσι, ώστε να γνωρίζουμε τη πληροφορίες περιέχουν, για παράδειγμα το xml που περιέχει όλα τα δεδομένα για τους Υπάλληλους, το έχουμε ονομάσει Υπαλλίλοι.xml. Όλα τα αρχεία των βάσεων μας βρίσκονται στον φάκελο asfaleies/admin/assets/database.

Μέσα στο συγκεκριμένο αρχείο λοιπόν θα αποθηκευτούν τα δεδομένα που θα εισάγουμε στην φόρμα όταν πατήσουμε Αποθήκευση. Το Tmimata.xml περιέχει τις εξής ετικέτες που θα μας χρειαστούν:

```
<tmimata>
  <tmima>
    <id_tmimatos></id_tmimatos>
    <onoma_tmimatos></onoma_tmimatos>
    <topothesia_tmimatos></topothesia_tmimatos>
    <dieufthintis_tmimatos></dieufthintis_tmimatos>
  </tmima>
</tmimata>
```

/tmimata/add.php – Το αρχείο εισαγωγής νέου Τμήματος

Όλα τα αρχεία της εργασίας μας, τα έχουμε ταξινομήσει και τα έχουμε ονομάσει με τέτοιο τρόπο ώστε να είναι πολύ κατανοητό τι περιέχουν και τι κάνουν. Στο συγκεκριμένο παράδειγμα, που θέλουμε να εισάγουμε νέο Τμήμα, αρκεί να μπούμε στον φάκελο tmimata/add.php, στον ίδιο φάκελο υπάρχουν και τα αρχεία edit.php και del.php όπου έχουν να κάνουν με την επεξεργασία και διαγραφή ενός τμήματος, όπως θα δούμε και στην συνέχεια.

Μέσα λοιπόν σε αυτό το αρχείο και σε κάθε αρχείο add.php έχουμε την φόρμα μας που είναι δημιουργημένη με την γλώσσα HTML και την PHP. Η φόρμα μας και όλες οι φόρμες μας χρησιμοποιούν την μέθοδο POST για να πάρουμε τα δεδομένα της όταν πατηθεί το κουμπί Αποθήκευση.

//Καθαρό δείγμα κώδικα της φόρμας εισαγωγής νέου Τμήματος που βρίσκεται στο αρχείο add.php. Αφαιρέθηκαν οι περιττές ετικέτες HTML που αφορούν την μορφοποίηση του κειμένου.

```
<form class=" " action="add.php" method="post">
  <input type="number" name="id_tmimatos">
  <label>Κωδικός Τμήματος</label>

  <input type="text" name="onoma_tmimatos">
  <label >Όνομα Τμήματος</label>

  <input type="text" name="topothesia_tmimatos">
  <label >Τοποθεσία Τμήματος</label>

  <select name="dieufthintis_tmimatos">
```

```
<option value="" disabled selected>Επιλογή Διευθυντή  
Τμήματος</option>
```

```
<?php foreach($xml2->children() as $ypallilos) {  
  
    echo "<option value=\"". $ypallilos->  
>onomatepwnymo_ypallilou . "\">". $ypallilos->  
>onomatepwnymo_ypallilou ."</option>";}?>  
    </select>  
    <label class="kat-auto-label">Επιλέξτε:</label>  
  
    <button type="submit" name="action">Αποθήκευση  
        </button>  
  
</form>
```

Σημαντικό ρόλο στον παραπάνω κώδικα και στην φόρμα μας είναι τα σημεία:

action="add.php", method="post" μέσα στην ετικέτα form. Αυτό το σημείο δείχνει ότι η μέθοδος αποστολής των δεδομένων μας είναι ο POST και το αρχείο στο οποίο θα στείλει τα δεδομένα είναι το add.php

Και μέσα σε όλα τα input, υπάρχει το name, σε αυτή την ιδιότητα, χρησιμοποιείται το όνομα της μεταβλητής που θα αποθηκεύσει και θα στείλει τα δεδομένα. Παρακάτω θα δούμε την χρήση τους αναλυτικά.

Μόλις εισάγουμε τα δεδομένα που θέλουμε και πατήσουμε Αποθήκευση, πάμε στο επόμενο στάδιο, της αποθήκευσης των δεδομένων.

Δημιουργία μίας νέας εγγραφής με την χρήση του DOM και Αποθήκευση τους στην Βάση Δεδομένων

```
//Κώδικας Εισαγωγής  
$xml = new DOMDocument("1.0", "UTF-8");  
$xml->preserveWhiteSpace = false;  
$xml->formatOutput = true;  
$xml->load("../assets/database\Tmimata.xml");  
$rootTag = $xml->getElementsByTagName("tmimata")->item(0);  
$dataTag = $xml->createElement("tmima");  
  
$id_tmimatosTag = $xml->createElement("id_tmimatos",  
$REQUEST['id_tmimatos']);  
$onoma_tmimatosTag = $xml->createElement("onoma_tmimatos",  
$REQUEST['onoma_tmimatos']);  
$topothesia_tmimatosTag = $xml->  
>createElement("topothesia_tmimatos",  
$REQUEST['topothesia_tmimatos']);  
$dieufthintis_tmimatosTag = $xml->
```

```
>createElement("dieufthintis_tmimatos",
$_REQUEST['dieufthintis_tmimatos']);

$dataTag->appendChild($id_tmimatosTag);
$dataTag->appendChild($onoma_tmimatosTag);
$dataTag->appendChild($topothesis_tmimatosTag);
$dataTag->appendChild($dieufthintis_tmimatosTag);
$rootTag->appendChild($dataTag);

$xml->save("../assets/database/Tmimata.xml");
```

Αφού έχουμε λοιπόν πατήσει Αποθήκευση, το παραπάνω σημείο είναι υπεύθυνο της διαχείρισης των δεδομένων. Πιο αναλυτικά:

```
$xml = new DOMDocument("1.0", "UTF-8");
//Δημιουργία ενός νέου DOMDocument αντικειμένου.

$xml->preserveWhiteSpace = false;
$xml->formatOutput = true;

//Υπεύθυνο σημείο κώδικα, ως προς την σωστή μορφοποίηση του XML αρχείου, με τα κενά
$xml->load("../assets/database/Tmimata.xml");

//Φόρτωση αρχείου XML: Tmimata.xml

$rootTag = $xml->getElementsByTagName("tmimata")->item(0);
$dataTag = $xml->createElement("tmima");

//Εδώ δηλώνουμε την ρίζα του XML, η οποία είναι το στοιχείο tmimata

$id_tmimatosTag = $xml->createElement("id_tmimatos",
$_REQUEST['id_tmimatos']);
$onoma_tmimatosTag = $xml->createElement("onoma_tmimatos",
$_REQUEST['onoma_tmimatos']);
$topothesis_tmimatosTag = $xml-
>createElement("topothesis_tmimatos",
$_REQUEST['topothesis_tmimatos']);
$dieufthintis_tmimatosTag = $xml-
>createElement("dieufthintis_tmimatos",
$_REQUEST['dieufthintis_tmimatos']);

//Εισαγωγή νέου κόμβου, γίνεται με χρήση της συνάρτησης:
createElement ( string $name [, string $value ] )

Στην ουσία σε αυτό το σημείο του κώδικα δημιουργούνται οι νέες ετικέτες και αποθηκεύονται και τα νέα δεδομένα.

$dataTag->appendChild($id_tmimatosTag);
$dataTag->appendChild($onoma_tmimatosTag);
$dataTag->appendChild($topothesis_tmimatosTag);
```

```
$dataTag->appendChild($dieufthintis_tmimatosTag);  
$rootTag->appendChild($dataTag);
```

//Κλείσιμο όλων των παραπάνω ετικετών

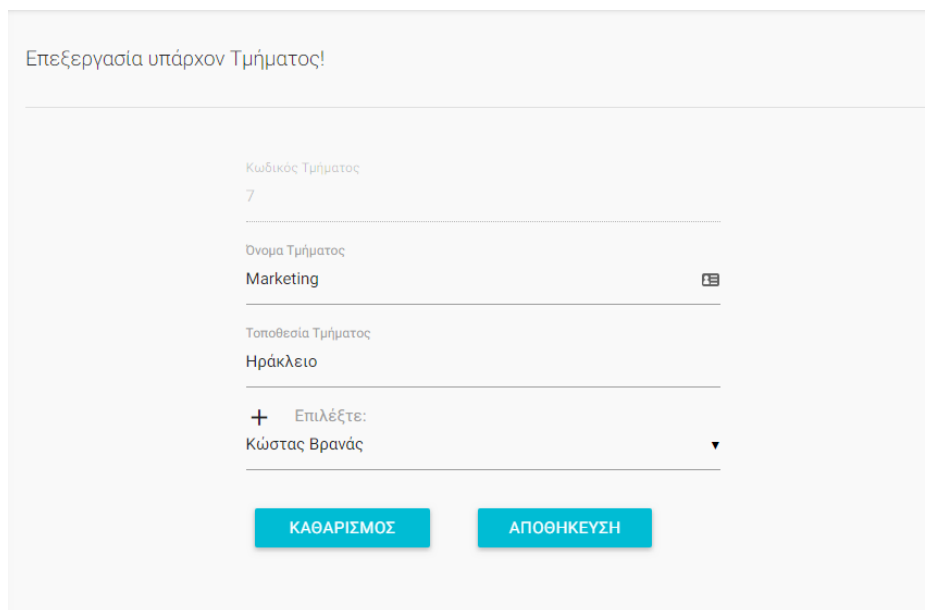
```
$xml->save("../assets/database\Tmimata.xml");
```

//Αποθήκευση όλων των αλλαγών στο αρχείο Tmimata.xml

Δείγμα Εισαγωγής:

```
<tmima>  
  <id_tmimatos>7</id_tmimatos>  
  <onoma_tmimatos>Marketing</onoma_tmimatos>  
  <topothesia_tmimatos>Ηράκλειο</topothesia_tmimatos>  
  <dieufthintis_tmimatos>Κώστας Βρανάς</dieufthintis_tmimatos>  
</tmima>
```

4.4.2 Επεξεργασία μίας εγγραφής από την Βάση Δεδομένων



Επεξεργασία υπάρχον Τμήματος!

Κωδικός Τμήματος
7

Όνομα Τμήματος
Marketing

Τοποθεσία Τμήματος
Ηράκλειο

+ Επιλέξτε:
Κώστας Βρανάς

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Εικόνα 20 - Φόρμα επεξεργασίας μίας υπάρχον εγγραφής του Τμήματος εταιρείας

Στην «Εικόνα 19» βλέπουμε την φόρμα επεξεργασίας ενός υπάρχον Τμήματος. Τα στοιχεία είναι ήδη συμπληρωμένα, με αυτά που υπάρχουν στην Βάση Δεδομένων, το μόνο που μένει είναι να κάνουμε την αλλαγή που θέλουμε και να πατήσουμε Αποθήκευση. Η μόνη αλλαγή που δεν μπορεί να γίνει είναι στο id, οπότε έχουμε:

- Όνομα Τμήματος
- Τοποθεσία Τμήματος
- Και τον Υπεύθυνο Διευθυντή του τμήματος αυτού

Tmimata.xml – Το αρχείο που περιέχει όλα τα στοιχεία για τα Τμήματα

Στο αρχείο μας τώρα έχουμε το παρακάτω με το id=7, που είναι για επεξεργασία:

```
<tmimata>
  <tmima>
    <id_tmimatos>7</id_tmimatos>
    <onoma_tmimatos>Marketing</onoma_tmimatos>
    <topothesisia_tmimatos>Ηράκλειο</topothesisia_tmimatos>
    <dieufthintis_tmimatos>Κώστας
    Βρανάς</dieufthintis_tmimatos>
  </tmima>
</tmimata>
```

/tmimata/edit.php?id=7 – Επεξεργασία του Τμήματος με βάση το ID

Όπως αναφερθήκαμε και παραπάνω, το αρχείο επεξεργασίας βρίσκεται μέσα στον ίδιο φάκελο με το αρχείο εισαγωγής. Η διαφορά τους εδώ είναι ότι η επεξεργασία χρειάζεται ένα id ώστε να μπορέσει να κάνει την επεξεργασία του κόμβου που πρέπει.

Μέσα λοιπόν σε αυτό το αρχείο και σε κάθε αρχείο edit.php έχουμε την φόρμα μας, με τα δεδομένα προς αλλαγή. Η φόρμα μας και όλες οι φόρμες μας χρησιμοποιούν την μέθοδο POST για να πάρουμε τα δεδομένα της όταν πατηθεί το κουμπί Αποθήκευση.

//Καθαρό δείγμα κώδικα της φόρμας επεξεργασίας ενός Τμήματος που βρίσκεται στο αρχείο edit.php. Αφαιρέθηκαν οι περιττές ετικέτες HTML που αφορούν την μορφοποίηση του κειμένου.

```
<form class=" " action="edit.php?id=<?php echo $read_id; ?>"
method="post">
  <input type="number" name="id_tmimatos" disabled
value="<?php echo $read_id; ?>"
  <label>Κωδικός Τμήματος</label>

  <input type="text" name="onoma_tmimatos" value="<?php
echo $read_onoma_tmimatos; ?>"
  <label >Όνομα Τμήματος</label>

  <input type="text" name="topothesisia_tmimatos"
value="<?php echo $read_topothesisia_tmimatos; ?>"
  <label >Τοποθεσία Τμήματος</label>

  <select name="dieufthintis_tmimatos">
  <option value="" disabled selected>Επιλογή Διευθυντή
Τμήματος</option>

<?php foreach($xml2->children() as $ypallilos) {
  $res=strcmp($ypallilos-
>onomatepwynomo_ypallilou,$read_dieufthintis_tmimatos);
```

```

                echo '<option value="'. $ypallilos-
>onomatepwnymo_ypallilou.'"
'. (($res==0)?'selected="selected"':"").'>'. $ypallilos-
>onomatepwnymo_ypallilou.'</option>';}
        </select>
        <label>Επιλέξτε:</label>

        <button type="submit" name="action">Αποθήκευση
                </button>

</form>

```

Όμως για αν εμφανιστούν τα δεδομένα στην φόρμα με την χρήση του `<?php echo $read_id; ?>`, έχουμε τον παραπάνω κώδικα που χρειάζεται:

```

if (isset($_GET['id'])) {

$value_id = $_GET["id"];
$xml=simplexml_load_file("../assets/database/Tmimata.xml");
$xml2=simplexml_load_file("../assets/database/Ypalliloi.xml");

    foreach($xml->tmima as $tmima)
    {
        if($tmima->id_tmimatos == $value_id) {
            $read_id = $tmima->id_tmimatos;
            $read_onoma_tmimatos = $tmima->onoma_tmimatos;
            $read_topothesis_tmimatos = $tmima-
>topothesis_tmimatos;
            $read_onoma_tmimatos = $tmima->onoma_tmimatos;
            $read_dieufthintis_tmimatos = $tmima-
>dieufthintis_tmimatos;
        }
    }
}

```

Με το σημείο αυτό του κώδικα, όταν κάνουμε χρήση του `edit.php?id=7` η μέθοδος GET παίρνει τον αριθμό του id. Έπειτα χρησιμοποιώντας την simpleXML, φορτώνουμε το αρχείο Tmimata.xml και το αρχείο Ypalliloi.xml (το αρχείο των υπαλλήλων χρειάζεται για το drop down μενού στην φόρμα μας) με την χρήση της δήλωσης: `simplexml_load_file()`

Έπειτα κάνουμε χρήση της `foreach (array_expression as $value)`⁴⁷ η οποία ανήκει στην PHP και είναι μία μέθοδος επανάληψης σε πίνακες ή αντικείμενα. Εμείς την χρησιμοποιήσαμε ώστε να ψάξει σε όλες τις εγγραφές του XML αρχείου Tmimata.xml, κάνοντας έλεγχο με την χρήση της `if`. Αν το id που πήραμε με την μέθοδο GET είναι ίδιο με κάποιο από τις υπάρχον εγγραφές, αν υπάρχει τότε οι μεταβλητές `$read_id`, `$read_onoma_tmimatos` κτλ παίρνουν τις

⁴⁷ foreach, PHP, <http://php.net/manual/en/control-structures.foreach.php>

αντίστοιχες τιμές που βρίσκονται μέσα στο XML αρχείο. Οι μεταβλητές αυτές χρησιμοποιούνται για την εμφάνιση στην φόρμα μας.

Ο υπόλοιπος κώδικας στην φόρμα μας δεν αλλάζει, μένει ίδιος με τον κώδικα του add.php.

Μόλις αλλάξουμε τα δεδομένα που θέλουμε και πατήσουμε Αποθήκευση, πάμε στο επόμενο στάδιο, της αλλαγής των δεδομένων στην Βάση Δεδομένων μας.

Αλλαγή δεδομένων και Αποθήκευση τους στην Βάση Δεδομένων

//Κώδικας Αλλαγής Δεδομένων

```
if(isset($_REQUEST['action'])) {  
  
    foreach($xml->tmima as $tmima)  
    {  
        if($tmima->id_tmimatos == $value_id) {  
  
            $tmima->id_tmimatos = $value_id;  
            $tmima->onoma_tmimatos = $_REQUEST['onoma_tmimatos'];  
            $tmima->topothesis_tmimatos =  
$_REQUEST['topothesis_tmimatos'];  
            $tmima->dieufthintis_tmimatos =  
$_REQUEST['dieufthintis_tmimatos'];  
  
            $xml->asXML("../assets/database/Tmimata.xml");  
        }  
    }  
}
```

Αφού έχουμε λοιπόν πατήσει Αποθήκευση, το παραπάνω σημείο είναι υπεύθυνο της αλλαγής των δεδομένων. Πιο αναλυτικά όπως βλέπουμε είναι πιο εύκολο να αλλάξουμε τα δεδομένα, καθώς το μόνο που χρειάζεται είναι να τρέξουμε μία foreach, να βρούμε το αντίστοιχο id ξανά, και εφόσον έχουμε ήδη κάνει χρήση του

```
$xml=simplexml_load_file("../assets/database/Tmimata.xml");
```

Και με την χρήση του post, από την φόρμα μας, αποθηκεύουμε κατευθείαν την πληροφορία μέσω του \$_REQUEST στο στοιχείο που πρέπει.

Και τέλος, δεν ξεχνάμε να κάνουμε αποθήκευση των αλλαγών, χρησιμοποιώντας την συνάρτηση: asXML(string) της simpleXML.

Δείγμα Αλλαγής:

```
<tmima>  
    <id_tmimatos>7</id_tmimatos>  
    <onoma_tmimatos>Marketing</onoma_tmimatos>  
    <topothesis_tmimatos>Χανιά</topothesis_tmimatos>  
    <dieufthintis_tmimatos>Κώστας Βρανάς</dieufthintis_tmimatos>  
</tmima>
```

4.4.3 Διαγραφή μίας εγγραφής από την Βάση Δεδομένων

Για την διαγραφή ενός κόμβου από ένα XML αρχείο χρησιμοποιούμε το DOM. Η διαγραφή γίνεται πάλι με βάση το id και ο κώδικας που την υλοποιεί είναι ο εξής:

```
if (isset($_GET['id'])) {

$value_id = $_GET["id"];

$xml=simplexml_load_file("../assets/database/Tmimata.xml");

foreach($xml->tmima as $tmima)
{
    if($tmima->id_tmimatos == $value_id) {
        $dom=dom_import_simplexml($tmima);
        $dom->parentNode->removeChild($dom);
    }
}

$xml->asXML("../assets/database/Tmimata.xml");
}
```

Ο κώδικας είναι ακριβώς όπως τον κώδικα που βρίσκεται στην επεξεργασία των δεδομένων. Πάλι εδώ χρησιμοποιούμε την foreach για να βρεθεί ο κόμβος με το id που έχει δηλωθεί και με την χρήση του dom_import_simplexml(\$tmima) και του parentNode->removeChild(\$dom) γίνεται η διαγραφή του συγκεκριμένου κόμβου.

4.5 Υλοποίηση Εφαρμογής

Σε αυτή την ενότητα θα κάνουμε χρήση στιγμιότυπων από την εφαρμογής μας με επεξηγήσεις και όπου χρειάζεται και με επεξήγηση κώδικα. Εφόσον έχουμε κάνει Start τον Server μας, ανοίγουμε τον φυλλομετρητή μας και μπαίνουμε στην διεύθυνση όπως προ είπαμε:

localhost/asfaleies

Αν δεν πληκτρολογήσουμε κάποιο άλλο όνομα αρχείου, ο Server ψάχνει για τα αρχεία index.html ή index.php. Εμείς έχουμε το αρχείο index.php όπου είναι η αρχική μας σελίδα. Θα αναφερθούμε πολύ γρήγορα στον απλό ιστότοπο που δημιουργήσαμε ώστε μετά να αναφερθούμε λεπτομερώς στο σύστημα διαχείρισης των ασφαλειών.

Στο απλό ιστότοπο όπου μπορούν να έχουν πρόσβαση όλοι. Είναι ένας απλός ιστότοπος ο οποίος περιέχει μία αρχική σελίδα (Εικόνα 19), μία σελίδα με λίγα λόγια για την εταιρεία, μία σελίδα για το Υπηρεσίες, το οποίο προσφέρει ενημέρωση για τα ασφαλιστήρια και τέλος μία σελίδα επικοινωνίας, της οποίας τα μηνύματα εμφανίζονται στο σύστημα διαχείρισης.

Για να εισέλθει κάποιος στο σύστημα διαχείρισης αρκεί να πατήσει το κουμπί Είσοδος πάνω δεξιά.

ΑΣΦΑΛΕΙΕΣ ΑΥΤΟΚΙΝΗΤΩΝ
INSURANCE GROUP

Αρχική Σχετικά Υπηρεσίες Επικοινωνία ΕΙΣΟΔΟΣ

Ασφάλειες Αυτοκινήτων

Τώρα οι ασφάλειες αυτοκινήτων στα μέτρα σας!

ΜΑΘΕΤΕ ΠΕΡΙΣΣΟΤΕΡΑ

Basic Insurance
από **150€**

Το βασικό πρόγραμμα ασφάλισης αυτοκινήτου, περιλαμβάνει την υποχρεωτική από το νόμο ασφάλιση για το αυτοκίνητό σας, καθώς και μία σειρά από επιπλέον καλύψεις που σας προσφέρουν ακόμα μεγαλύτερη σιγουριά.

Plus Insurance
από **250€**

Το πακέτο καλύψεων Plus Insurance μπορεί να προστεθεί στο βασικό πρόγραμμα Basic και προσφέρει μία σειρά από επιπλέον καλύψεις και παροχές για μεγαλύτερη προστασία του αυτοκινήτου σας.

Extra Plus Insurance
από **350€**

Το πακέτο καλύψεων Extra Plus Insurance μπορεί να προστεθεί στο πρόγραμμα Plus Saver και προσφέρει μία σειρά από επιπλέον καλύψεις και παροχές για την απόλυτη προστασία του αυτοκινήτου σας.

Δυσκολεύεστε να αποφασίσετε;

Πείτε μας τις ανάγκες σας κι εμείς θα σας βοηθήσουμε!

[ΡΩΤΗΣΤΕ ΜΑΣ](#)

Ασφάλιση στα μέτρα σας

Εσείς επιλέγετε το πακέτο που ταιριάζει στις ανάγκες, τις προτιμήσεις και τον προϋπολογισμό σας.

Δωρεάν Φροντίδα Ατυχήματος

Όλα τα ασφαλιστικά πακέτα μας περιλαμβάνουν δωρεάν Φροντίδα Ατυχήματος, που παρέχεται από την ασφαλιστική εταιρεία μας.

Ασφάλιστρα σε άτοκες δόσεις

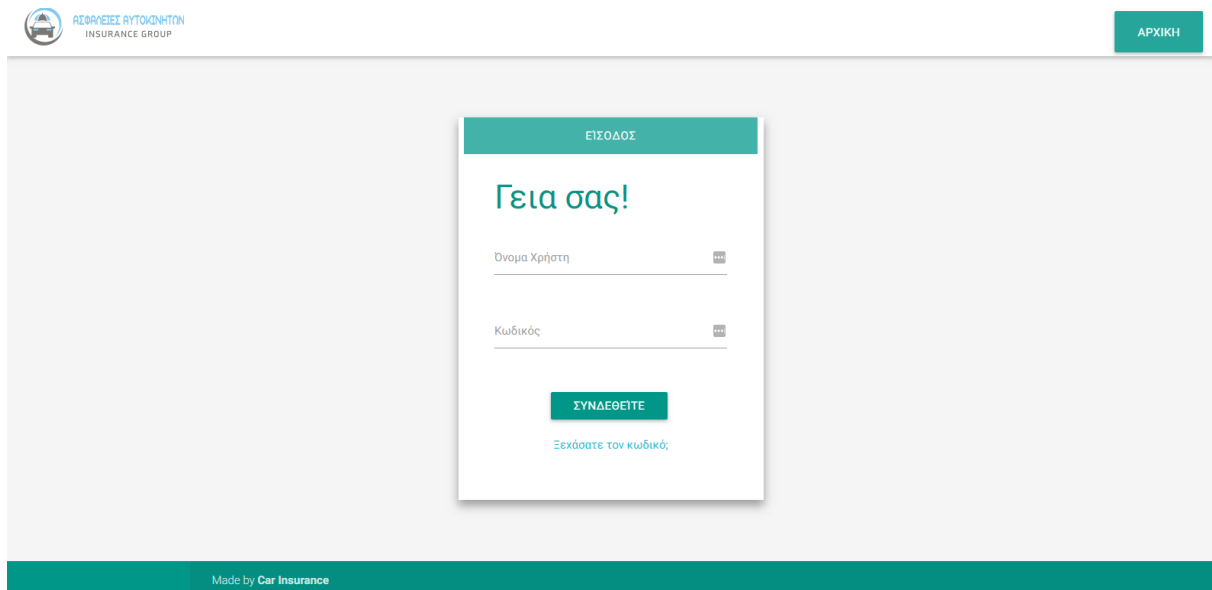
Χρησιμοποιώντας οποιαδήποτε πιστωτική κάρτα μπορείτε να καταβάλλετε τα ασφάλιστρα σας μέχρι και σε 12 άτοκες μηνιαίες δόσεις.

Made by Car Insurance

Εικόνα 21 - Αρχική Σελίδα Ιστοτόπου

4.5.1 Είσοδος στο σύστημα Διαχείρισης

Πατώντας κάποιος στο κουμπί Είσοδος λοιπόν, ανοίγει μία σελίδα που πρέπει να κάνει Log in με τα στοιχεία του. Στοιχεία εισόδου έχουν μόνο όλοι οι υπάλληλοι της εταιρείας.



Εικόνα 22 - Σελίδα Εισόδου για Διαχειριστές

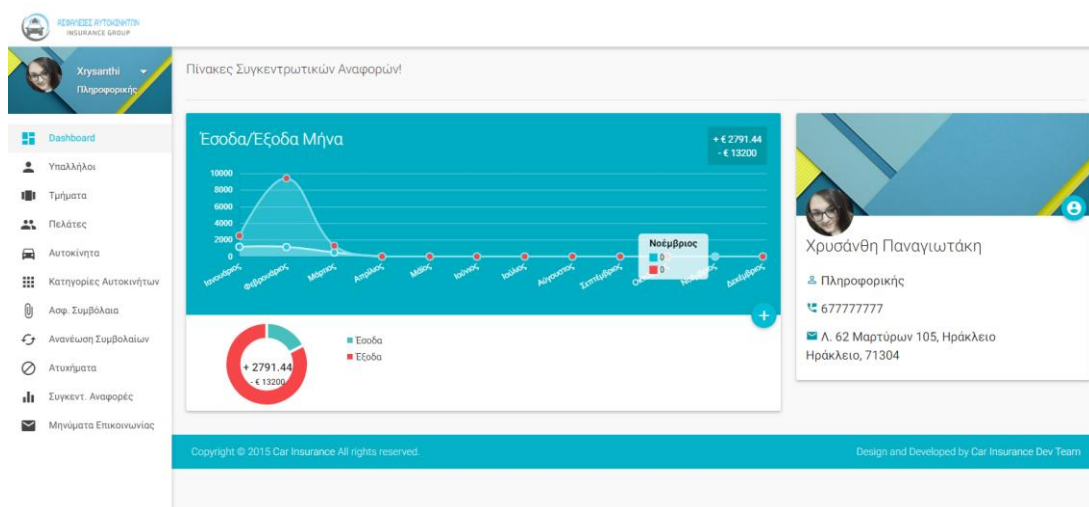
Τα στοιχεία εισόδου μας είναι:

- xrysanthi
- xrysanthi123

Όπως επίσης και

- varvara
- varvara123

4.5.2 Αρχική Σελίδα Συστήματος Διαχείρισης



Εικόνα 23 - Αρχική Σελίδα Συστήματος Διαχείρισης

Η αρχική σελίδα του συστήματος διαχείρισης της ασφαλιστικής μας εταιρίας, περιλαμβάνει:

- Το μενού στα αριστερά, το οποίο εμφανίζεται σε όλες τις σελίδες.
- Πάνω αριστερά, επίσης φαίνεται ποιος υπάλληλος είναι συνδεδεμένος και έχει την επιλογή Αποσύνδεσης.
- Στην κεντρική σελίδα εμφανίζεται μία μηνιαία αναφορά εσόδων / εξόδων
- Και στα δεξιά της, μερικά στοιχεία του συνδεδεμένου χρήστη που οι πληροφορίες αυτές αντλούνται από την βάση δεδομένων μας.

4.5.3 Υπάλληλοι

Πατώντας από το μενού αριστερά το Υπάλληλοι, μας εμφανίζει μία λίστα με όλους τους Υπαλλήλους της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Εδώ θα βρείτε όλα τα στοιχεία των Υπαλλήλων της Ασφαλιστικής μας εταιρείας!

Εμφάνιση: 100 Αναζήτηση:

Id	Όνοματεπώνυμο	Οδός	Αριθμός	Πόλη - Περιοχή	Νομός	TK	Τηλέφωνο	Τμήμα Εργασίας
1	Χρυσάνθη Παναγιωτάκη	Λ. 62 Μαρτύρων	105	Ηράκλειο	Ηράκλειο	71304	677777777	Πληροφορικής
2	Βαββάρα Φουκάκη	Πυράνθου	30	Ηράκλειο	Ηράκλειο	71303	691234566	Πληροφορικής
3	Γιώργος Ποντικάκης	Δικαιοσύνης	3	Ηράκλειο	Ηράκλειο	71303	6987546661	Ασφαλείων
4	Μαρία Μπεκατωράκη	Εβανς	15	Ηράκλειο	Ηράκλειο	71303	6934978218	Ασφαλείων
5	Σοφία Ντελλάκη	Εβανς	15	Ηράκλειο	Ηράκλειο	71303	6978242660	Λογιστήριο
6	Κώστας Βρανάς	Μεγάρου	33	Ηράκλειο	Ηράκλειο	71303	6978242743	Marketing
7	Μάριος Κωστάκης	Τάλως	12	Ηράκλειο	Ηράκλειο	71303	6978562660	Γραμματεία
8	Ελένη Βλαχάκη	Σμπώκου	7	Ηράκλειο	Ηράκλειο	71303	6978239660	Αποζημιώσεων
9	Αθηνά Πιπεράκη	Λεβήνου	45	Ηράκλειο	Ηράκλειο	71304	6976832610	Προσωπικού

Εμφανίζονται 9 από τις 9 εγγραφές

Εικόνα 24 - Μενού Υπάλληλοι

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Υπαλλήλου, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

4.5.3.1 Αρχείο XML για τους Υπαλλήλους

Το αρχείο της βάσης δεδομένων των Υπαλλήλων βρίσκεται στο `asfaleies/admin/assets/database/Ypalliloi.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το `Ypalliloi.xml` περιέχει τις εξής ετικέτες:

```
<ypalliloi>
  <ypallilos>
    <id_ypallilou>1</id_ypallilou>
    <onomatepwynomo_ypallilou>Χρυσάνθη
    Παναγιωτάκη</onomatepwynomo_ypallilou>
    <dief_odos_ypallilou>Λ. 62 Μαρτύρων</dief_odos_ypallilou>
    <dief_arithmos_ypallilou>105</dief_arithmos_ypallilou>
    <dief_poli_ypallilou>Ηράκλειο</dief_poli_ypallilou>
    <dief_nomos_ypallilou>Ηράκλειο</dief_nomos_ypallilou>
    <dief_tk_ypallilou>71304</dief_tk_ypallilou>
    <phone_ypallilou>677777777</phone_ypallilou>
    <tmima_ergasias_ypallilou>Πληροφορικής</tmima_ergasias_yp
allilou>
    <username>xrysanthi</username>
    <password>$2y$10$IBZujo6pOK86YINQA7v.PO255.YTJe.LzIoyhr2Cb
    GjUBXLxMmhLq</password>
    <fwtografia_ypallilou>xrysanthi-
panagiwtaki.jpg</fwtografia_ypallilou>
  </ypallilos>
</ypalliloi>
```

4.5.3.2 Εισαγωγή ενός νέου Υπαλλήλου

Η Εισαγωγή ενός Υπαλλήλου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1. Το μόνο που διαφέρει είναι ότι υπάρχουν περισσότερα πεδία για συμπλήρωση.

Προσθήκη καινούργιου Υπαλλήλου!

Μονοκλάσικς Κωδικός:

Κωδικός Υπαλλήλου

Όνοματεπώνυμο Υπαλλήλου:

Όνοματεπώνυμο

Διεύθυνση Υπαλλήλου:

Οδός Αριθμός

Πόλη - Περιοχή Νομός

Τ.Κ.

Λογιά Στοιχεία:

Τηλέφωνο Υπαλλήλου

+ Επιλέξτε: Λίστα Τμημάτων Εργασιών

Στοιχεία Σύνδεσης Online:

Username Password

Φωτογραφία Υπαλλήλου:

FILE

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2010 Car Insurance. All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 25- Φόρμα Εισαγωγής νέου Υπαλλήλου

4.5.3.3 Επεξεργασία ενός υπάρχον Υπαλλήλου

Η Επεξεργασία ενός Υπαλλήλου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2. Το μόνο που διαφέρει είναι ότι υπάρχουν περισσότερα πεδία για επεξεργασία.

Επεξεργασία υπάρχον Υπαλλήλου!

Μονοκλάσικς Κωδικός:

Κωδικός Υπαλλήλου

Όνοματεπώνυμο Υπαλλήλου:

Όνοματεπώνυμο Χρυσάνθη Παναγιωτίδα

Διεύθυνση Υπαλλήλου:

Οδός Αριθμός

A. 62 Μαρτύρων 105

Πόλη - Περιοχή Νομός

Ηράκλειο Ηράκλειο

Τ.Κ.

71304

Λογιά Στοιχεία:

Τηλέφωνο Υπαλλήλου

61777777

+ Επιλέξτε: Πληροφορικής

Στοιχεία Σύνδεσης Online:

Username Password

xyrsanthe Πληροφορικής

FILE xyrsanthe-panagiwtaki.jpg

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2010 Car Insurance. All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 26 - Φόρμα Επεξεργασίας ενός Υπαλλήλου

4.5.3.4 Διαγραφή κόμβου ενός υπάρχον Υπαλλήλου

Η Διαγραφή ενός Υπαλλήλου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

4.5.4 Τμήματα

Πατώντας από το μενού αριστερά το Τμήματα, μας εμφανίζει μία λίστα με όλα τα Τμήματα της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Εδώ θα βρείτε όλα τα Τμήματα του Car Insurance και την τοποθεσία τους!

Εμφάνιση: 100

Id	Όνομα Τμήματος	Τοποθεσία	Υπεύθυνος Τμήματος
1	Πληροφορικής	Ηράκλειο	Χρυσάνθη Παναγιωτάκη
2	Ασφαλειών	Ηράκλειο	Γιώργος Ποντικάκης
3	Λογιστήριο	Ηράκλειο	Σοφία Ντελλάκη
4	Γραμματεία	Ηράκλειο	Μάριος Κωστάκης
5	Αποζημιώσεων	Ηράκλειο	Ελένη Βλαχάκη
6	Προσωπικού	Ηράκλειο	Αθηνά Πιπεράκη
7	Marketing	Ηράκλειο	Κώστας Βρανάς

Εμφανίζονται 7 από τις 7 εγγραφές

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 27 - Μενού Τμήματα

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Τμήματος, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη Ενότητα 4.4.

4.5.4.1 Αρχείο XML για τα Τμήματα

Το αρχείο της βάσης δεδομένων των Τμημάτων βρίσκεται στο asfaleies/admin/assets/database/Tmimata.xml.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

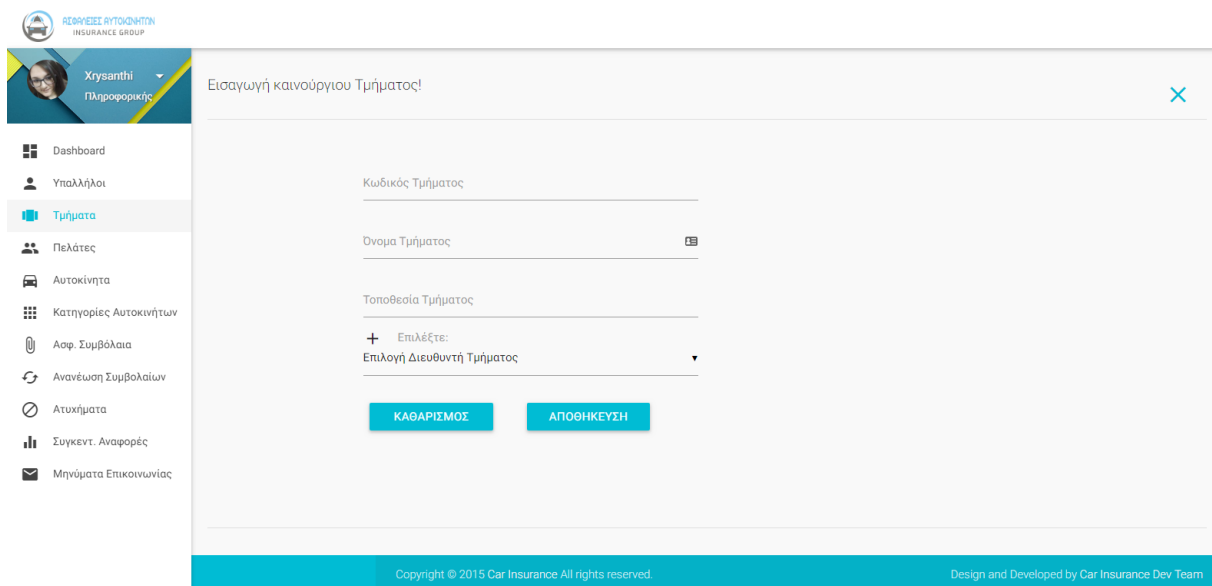
Το Tmimata.xml περιέχει τις εξής ετικέτες:

```
<tmimata>
  <tmimata>
```

```
<id_tmimatos>1</id_tmimatos>  
<onoma_tmimatos>Πληροφορικής</onoma_tmimatos>  
<topothesisia_tmimatos>Ηρόκλειο</topothesisia_tmimatos>  
<dieufthintis_tmimatos>Χρυσάνθη  
Παναγιωτάκη</dieufthintis_tmimatos>  
</tmima>  
</tmimata>
```

4.5.3.2 Εισαγωγή ενός νέου Τμήματος

Την εισαγωγή ενός Τμήματος την αναλύσαμε ακριβώς στην Ενότητα 4.4.1.

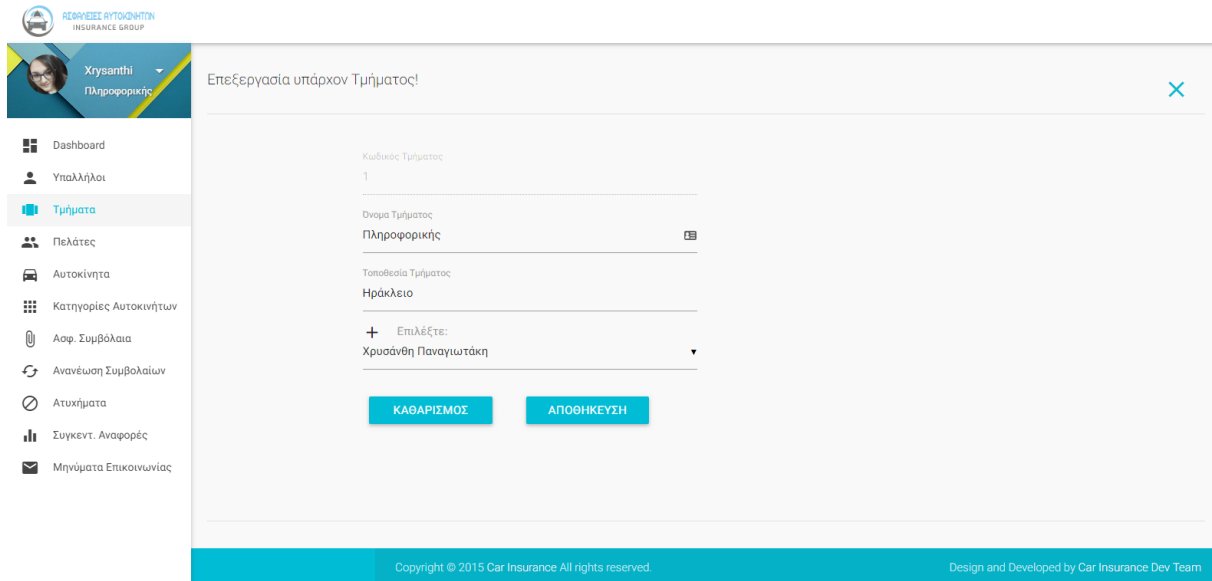


The screenshot shows a web application interface for adding a new department. The header includes the logo for 'ΑΕΦΥΣΕΣ ΑΥΤΟΚΙΝΗΤΩΝ INSURANCE GROUP' and the user profile 'Χρυσάνθη Πληροφορικής'. A sidebar menu on the left lists various navigation options, with 'Τμήματα' (Departments) highlighted. The main content area is titled 'Εισαγωγή καινούργιου Τμήματος!' (Add New Department!) and contains a form with the following fields: 'Κωδικός Τμήματος' (Department Code), 'Όνομα Τμήματος' (Department Name), 'Τοποθεσία Τμήματος' (Department Location), and a dropdown menu for 'Επιλέξτε: Επιλογή Διευθυντή Τμήματος' (Select: Department Director Selection). At the bottom of the form are two buttons: 'ΚΑΘΑΡΙΣΜΟΣ' (Clear) and 'ΑΠΟΘΗΚΕΥΣΗ' (Save). The footer of the page contains copyright information: 'Copyright © 2015 Car Insurance All rights reserved.' and 'Design and Developed by Car Insurance Dev Team'.

Εικόνα 28 - Εισαγωγή ενός Τμήματος

4.5.4.3 Επεξεργασία ενός υπάρχον Τμήματος

Την Επεξεργασία ενός Τμήματος την αναλύσαμε ακριβώς στην Ενότητα 4.4.2.



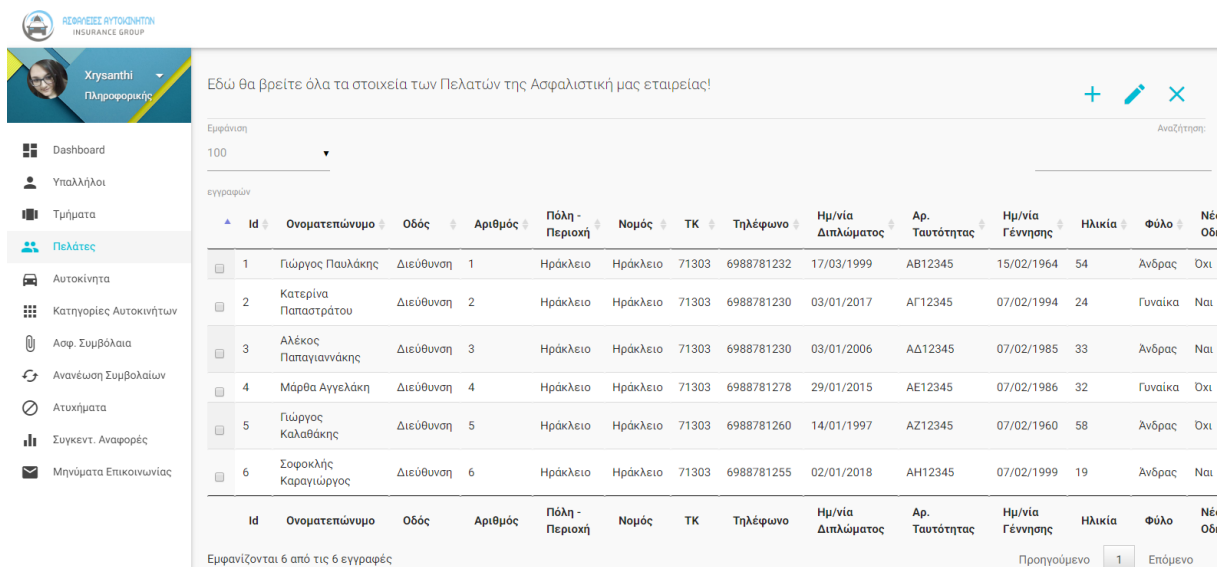
Εικόνα 29 - Επεξεργασία Τμήματος

4.5.4.4 Διαγραφή κόμβου ενός υπάρχον Υπαλλήλου

Την Διαγραφή ενός Τμήματος την αναλύσαμε ακριβώς στην Ενότητα 4.4.3.

4.5.5 Πελάτες

Πατώντας από το μενού αριστερά το Πελάτες, μας εμφανίζει μία λίστα με όλους τους Πελάτες της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.



Εικόνα 30 - Μενού Πελάτες

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Πελάτη, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

4.5.5.1 Αρχείο XML για τους Πελάτες

Το αρχείο της βάσης δεδομένων των Υπαλλήλων βρίσκεται στο `asfaleies/admin/assets/database/Pelates.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το `Pelates.xml` περιέχει τις εξής ετικέτες:

```
<pelates>
  <pelati>
    <id_pelati>1</id_pelati>
    <onomatepwynomo_pelati>Γιώργος
    Παυλάκης</onomatepwynomo_pelati>
    <dief_odos_pelati>Διεύθυνση</dief_odos_pelati>
    <dief_arithmos_pelati>1</dief_arithmos_pelati>
    <dief_poli_pelati>Ηράκλειο</dief_poli_pelati>
    <dief_nomos_pelati>Ηράκλειο</dief_nomos_pelati>
    <dief_tk_pelati>71303</dief_tk_pelati>
    <tilefwno_pelati>6988781232</tilefwno_pelati>
    <imerominia_diplwmatos_pelati>17/03/1999</imerominia_dipl
    wmatos_pelati>
    <arithmos_tautotitas_pelati>AB12345</arithmos_tautotitas_
    pelati>
    <imerominia_gennisis_pelati>15/02/1964</imerominia_gennis
    is_pelati>
    <ilikia_pelati>54</ilikia_pelati>
    <fylo_pelati>Ανδρας</fylo_pelati>
    <neos_odigos_pelatis>Οχι</neos_odigos_pelatis>
  </pelati>
</pelates>
```

4.5.5.2 Εισαγωγή ενός νέου Πελάτη

Η Εισαγωγή ενός Πελάτη, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1. Το μόνο που διαφέρει είναι ότι υπάρχουν περισσότερα πεδία για συμπλήρωση.

Εισαγωγή καινούργιου Πελάτη!

Μονοδικός Κωδικός Πελάτη:

Κωδικός Πελάτη

Όνοματεπώνυμο Πελάτη:

Όνοματεπώνυμο

Διεύθυνση Πελάτη:

Οδός Αριθμός

Πόλη - Περιοχή Νομός

Τ.Κ.

Λοιπά Στοιχεία:

Ημ/νία Γέννησης Πελάτη Ηλικία Πελάτη

+ Επιλέξτε:
Φύλο Πελάτη

Τηλέφωνο Πελάτη

Ημ/νία Διπλώματος Πελάτη

Αρ. Ταυτότητας Πελάτη

Είναι νέος οδηγός ο Πελάτης:

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 31 - Φόρμα Εισαγωγής Νέου Πελάτη

4.5.5.3 Επεξεργασία ενός υπάρχον Πελάτη

Η Επεξεργασία ενός Πελάτη, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2. Το μόνο που διαφέρει είναι ότι υπάρχουν περισσότερα πεδία για επεξεργασία.

Επεξεργασία υπάρχον Πελάτη!

Κωδικός Πελάτη: 1

Όνοματεπώνυμο Πελάτη: Γιώργος Παυλάκης

Διεύθυνση Πελάτη: Οδός: Διεύθυνση 1, Πόλη - Περιφέρεια: Ηράκλειο, Νομός: Ηράκλειο

Τ.Κ.: 71303

Λοιπά Στοιχεία: Ημέρ/νία Γέννησης Πελάτη: 15/02/1964, Ηλικία Πελάτη: 54

Επιλέξτε: Άνδρας

Τηλέφωνο Πελάτη: 6988781232

Ημέρ/νία Διπλώματος Πελάτη: 17/03/1999

Αρ. Ταυτότητας Πελάτη: AB12345

Είναι νέος οδηγός ο Πελάτης:

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

4.5.5.4 Διαγραφή κόμβου ενός υπάρχον Πελάτη

Η Διαγραφή ενός Πελάτη, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

4.5.6 Αυτοκίνητα

Πατώντας από το μενού αριστερά το Αυτοκίνημα, μας εμφανίζει μία λίστα με όλα τα Αυτοκίνητα της εταιρείας, σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Πελάτη, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

The screenshot shows a web application interface for 'ΑΣΦΑΛΙΣΤΙΚΟ ΟΜΙΛΟ' (Insurance Group). The user is logged in as 'Χρυσανθί Πληροφορικής'. The left sidebar contains a menu with options like Dashboard, Υπαλλήλοι, Τμήματα, Πελάτες, Αυτοκίνητα (selected), Κατηγορίες Αυτοκινήτων, Ασφ. Συμβόλαια, Ανανέωση Συμβολαίων, Ατυχήματα, Συγκεντ. Αναφορές, and Μηνύματα Επικοινωνίας. The main content area displays a table of vehicles with the following data:

Id	Οδηγός Αυτοκινήτου	Αρ. Ταυτότητας	Αριθμός Κυκλοφορίας	Ημερομηνία Ταξινόμησης	Κατηγορία Αυτοκινήτου
1	Γιώργος Παυλάκης	AB12345	AZH4356	02/01/2018	A (Εως 300)
2	Κατερίνα Παπαστράτου	ΑΓ12345	AKH4356	15/02/2018	A (Εως 300)
3	Αλέκος Παπαγιαννάκης	ΑΔ12345	AEK1234	22/03/2018	B (301-785)
4	Κατερίνα Παπαστράτου	ΑΓ12345	AKI1687	04/04/2018	Δ (1.072-1.357)
5	Σοφοκλής Καραγιώργος	ΑΗ12345	AEP1234	01/01/2018	Ε (1.358-1.548)

At the bottom of the table, it indicates 'Εμφανίζονται 5 από τις 5 εγγραφές' and a pagination control showing '1' of 1 page.

Εικόνα 32 - Μενού Αυτοκίνητα

4.5.6.1 Αρχείο XML για τα Αυτοκίνητα

Το αρχείο της βάσης δεδομένων των Αυτοκινήτων βρίσκεται στο `asfaleies/admin/assets/database/Autokinita.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το `Autokinita.xml` περιέχει τις εξής ετικέτες:

```
<autokinita>
  <autokinito>
    <id_autokinitou>1</id_autokinitou>
    <odigos_autokinitou>Γιώργος Παυλάκης</odigos_autokinitou>
    <at_pelati_autokinitou>AB12345</at_pelati_autokinitou>

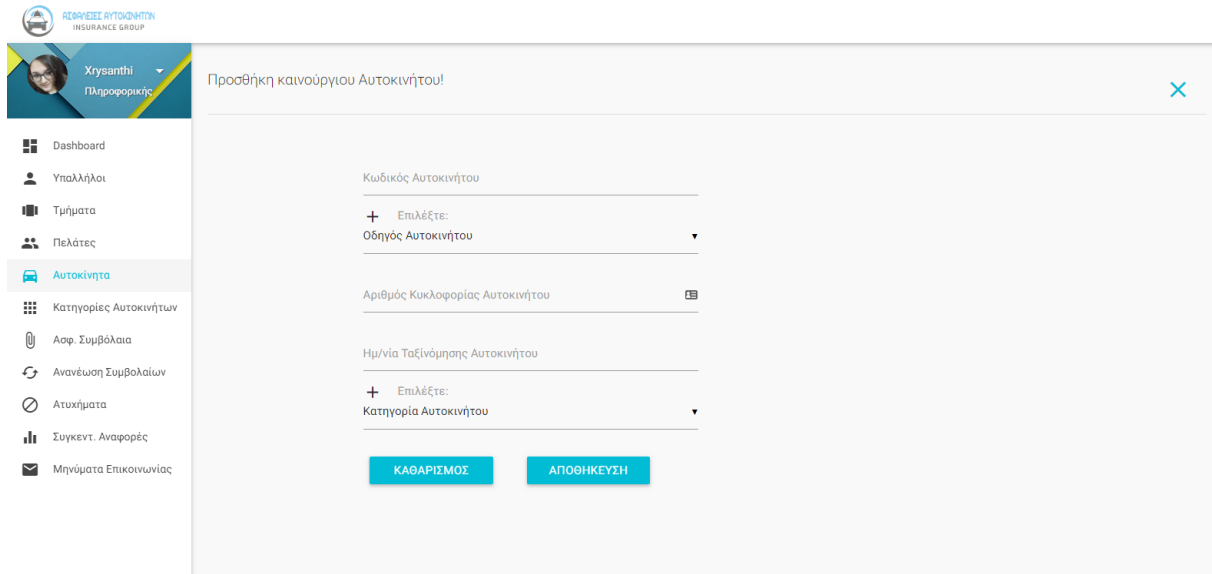
    <arithmos_kykloforias_autokinitou>AZH4356</arithmos_kykloforias_autokinitou>

    <imerominia_taksinomisis_autokinitou>02/01/2018</imerominia_taksinomisis_autokinitou>
```

```
<katigoria_autokinitou>A (Έως 300)</katigoria_autokinitou>  
<asf_symvolaio>1</asf_symvolaio>  
</autokinito>  
</autokinita>
```

4.5.6.2 Εισαγωγή ενός νέου Αυτοκινήτου

Η Εισαγωγή ενός Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1.



The screenshot displays the 'Προσθήκη καινούργιου Αυτοκινήτου!' (Add new car!) form. On the left is a sidebar with the user's name 'Χρυσανθή Πληροφορικής' and a menu with options like 'Dashboard', 'Υπαλλήλοι', 'Τμήματα', 'Πελάτες', 'Αυτοκίνητα', 'Κατηγορίες Αυτοκινήτων', 'Ασφ. Συμβόλαια', 'Ανανέωση Συμβολαίων', 'Ατυχήματα', 'Συγκεντ. Αναφορές', and 'Μηνύματα Επικοινωνίας'. The main form contains the following fields:

- Κωδικός Αυτοκινήτου: A text input field.
- Επιλέξτε: Οδηγός Αυτοκινήτου: A dropdown menu.
- Αριθμός Κυκλοφορίας Αυτοκινήτου: A text input field with a calendar icon.
- Ημ/νία Ταξινόμησης Αυτοκινήτου: A text input field.
- Επιλέξτε: Κατηγορία Αυτοκινήτου: A dropdown menu.

At the bottom of the form are two buttons: 'ΚΑΘΑΡΙΣΜΟΣ' (Clear) and 'ΑΠΟΘΗΚΕΥΣΗ' (Save).

Εικόνα 33 - Εισαγωγή νέου Αυτοκινήτου

4.5.6.3 Επεξεργασία ενός υπάρχον Αυτοκινήτου

Η Επεξεργασία ενός Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2.

Κωδικός Αυτοκινήτου

+ Επιλέξτε: Οδός Αυτοκινήτου

Αριθμός Κυκλοφορίας Αυτοκινήτου

Ημ/νία Ταξινόμησης Αυτοκινήτου

+ Επιλέξτε: Κατηγορία Αυτοκινήτου

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Εικόνα 34 - Επεξεργασία ενός Αυτοκινήτου

4.5.6.4 Διαγραφή κόμβου ενός υπάρχον Αυτοκινήτου

Η Διαγραφή ενός Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

4.5.7 Κατηγορίες Αυτοκινήτων

Πατώντας από το μενού αριστερά το Κατηγορίες Αυτοκινήτων, μας εμφανίζει μία λίστα με όλους τις Κατηγορίες Αυτοκινήτων της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Εδώ θα βρείτε όλα τα στοιχεία των Κατηγοριών των Αυτοκινήτων!

Εμφάνιση: 100 Αναζήτηση

εγγραφών	Id	Εύρος Κυβικών	Τύπος Αυτοκινήτου	Ασφάλιστρα Εξαμήνου
	1	A (Εως 300)	Επιβατικό	120
	2	B (301-785)	Επιβατικό	203
	3	Γ (786-1.071)	Επιβατικό	250
	4	Δ (1.072-1.357)	Εμπορικό	298
	5	Ε (1.358-1.548)	Εμπορικό	360
	6	ΣΤ (1.549-1.738)	Εμπορικό	450

Εμφανίζονται 6 από τις 6 εγγραφές

Προηγούμενο 1 Επόμενο

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 35 - Μενού Κατηγορίες Αυτοκινήτων

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή μίας νέας Κατηγορίας Αυτοκινήτων, η Επεξεργασία μίας υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

4.5.7.1 Αρχείο XML για τις Κατηγορίες Αυτοκινήτων

Το αρχείο της βάσης δεδομένων για τις Κατηγορίες Αυτοκινήτων βρίσκεται στο `asfaleies/admin/assets/database/Kat-Autokinitwn.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το `Kat-Autokinitwn.xml` περιέχει τις εξής ετικέτες:

```
<katigories>
  <katigoria>
    <id_katigorias>1</id_katigorias>
    <evros_kyvikwn_katigorias>A (Έως
300)</evros_kyvikwn_katigorias>
    <typos_autokinitou_katigorias>Επιβατικό</typos_autokinito
u_katigorias>
    <asfalistra_eksaminou_katigorias>120</asfalistra_eksamino
u_katigorias>
  </katigoria>
</katigories>
```

4.5.7.2 Εισαγωγή μίας νέας Κατηγορίας Αυτοκινήτου

Η Εισαγωγή μίας νέας Κατηγορίας Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1.

Εισαγωγή καινούργιας Κατηγορίας Αυτοκινήτου!

Κωδικός Κατηγορίας

+ Επιλέξτε: Κατηγορίες Κυβικών

+ Επιλέξτε: Τύπος Αυτοκινήτου

Ασφάλιστρα Εξάμηνο (€)

0

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 36 - Φόρμα Εισαγωγής μίας νέας Κατηγορίας Αυτοκινήτου

4.5.7.3 Επεξεργασία μίας Κατηγορίας Αυτοκινήτου

Η Επεξεργασία μίας Κατηγορίας Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2.

Επεξεργασία υπάρχον Κατηγορίας Αυτοκινήτου!

Κωδικός Κατηγορίας

1

+ Επιλέξτε: Α' (Εως 300)

+ Επιλέξτε: Επιβατικό

Ασφάλιστρα Εξάμηνο (€)

120

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗ

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 37 - Φόρμα Επεξεργασίας μίας Κατηγορίας Αυτοκινήτου

4.5.7.4 Διαγραφή κόμβου μίας Κατηγορίας Αυτοκινήτου

Η Διαγραφή μίας Κατηγορίας Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

4.5.8 Ασφαλιστικά Συμβόλαια

Πατώντας από το μενού αριστερά το Ασφαλιστικά Συμβόλαια, μας εμφανίζει μία λίστα με όλα τα Ασφαλιστικά Συμβόλαια της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Εδώ θα βρείτε όλα τα στοιχεία των Ασφαλιστικών Συμβολαίων μας!

Εμφάνιση: 100

Αναζήτηση

Id	Πελάτης	Αυτοκίνητο	Κατ. Αυτοκινήτου	Ημερομηνία Συμβολαίου	Ποσόν	Υπ. Υπάλληλος
1	Αλέκος Παπαγιαννάκης	ΑΕΚ1234	B (301-785)	03/01/2018	487.2	Γιώργος Ποντικάκης
2	Κατερίνα Παπαστράτου	ΑΚΗ4356	A (Εως 300)	01/02/2018	273.6	Βαρβάρα Φουκάκη
3	Αλέκος Παπαγιαννάκης	ΑΕΚ1234	B (301-785)	01/03/2018	487.2	Χρυσάνθη Παναγιωτάκη
4	Σοφοκλής Καραγιώργος	ΑΕΡ1234	E (1.358-1.548)	15/02/2018	864	Γιώργος Ποντικάκης
5	Κατερίνα Παπαστράτου	ΑΚΙ1687	Δ (1.072-1.357)	15/01/2018	679.44	Μαρία Μπεκατωράκη

Εμφανίζονται 5 από τις 5 εγγραφές

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

Εικόνα 38 - Μενού Ασφαλιστικά Συμβόλαια

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Ασφαλιστικού Συμβολαίου, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

4.5.8.1 Αρχείο XML για τα Ασφαλιστικά Συμβόλαια

Το αρχείο της βάσης δεδομένων για τα Ασφαλιστικά Συμβόλαια βρίσκεται στο `asfaleies/admin/assets/database/Asf-Symvolaia.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το `Asf-Symvolaia.xml` περιέχει τις εξής ετικέτες:

```
<symvolaia>
  <symvolaio>
    <id_symvolaiou>1</id_symvolaiou>
    <autokinita_pelati_symvolaiou>ΑΕΚ1234</autokinita_pelati_
symvolaio>
    <pelatis_symvolaiou>Αλέκος
Παπαγιαννάκης</pelatis_symvolaiou>
```

```

<katigoria_autokinitou_symvolaiou>B (301-
785)</katigoria_autokinitou_symvolaiou>
<imerominia_symvolaiou>03/01/2018</imerominia_symvolaiou>
<imerominia_ananewsisis_symvolaiou>03/01/2018</imerominia_a
nanewsisis_symvolaiou>
<poson_symvolaiou>487.2</poson_symvolaiou>
<yref_ypallilos_symvolaiou>Γιώργος
Ποντικάκης</yref_ypallilos_symvolaiou>
</symvolaiou>
</symvolaiou>

```

4.5.8.2 Εισαγωγή ενός νέου Ασφαλιστικού Συμβολαίου

Η Εισαγωγή ενός νέου Ασφαλιστικού Συμβολαίου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1. Το μόνο που διαφέρει είναι ότι εδώ χρειάζεται να γίνει αυτόματος υπολογισμός του ποσού πληρωμής των ασφαλιστηρίων.

Εικόνα 39 - Φόρμα Εισαγωγής ενός νέου Ασφαλιστηρίου Αυτοκινήτου

Η ιδιαιτερότητα εδώ είναι στον υπολογισμό του Ασφαλιστηρίου. Το ποσόν διαμορφώνεται αυτόματα με βάση:

Τα ασφάλιστρα που πληρώνει κάθε πελάτης διαμορφώνονται ως εξής: την πρώτη φορά είναι όσα ορίζει η κατηγορία του αυτοκινήτου που έχει, εκτός αν είναι νέος οδηγός οπότε έχει 20% αύξηση. Αν περάσει ένας χρόνος χωρίς ατύχημα έχει 10% μείωση (αυτό συνεχίζεται μέχρι το 30% λιγότερο από το κανονικό). Αν κάνει ατύχημα και έχει την ευθύνη αυτός έχει αύξηση 30%. Αυτή η αύξηση ισχύει για κάθε ατύχημα. Αν περάσει χρόνος χωρίς να προκαλέσει άλλο ατύχημα έχει μείωση 10% (αυτό συνεχίζεται μέχρι το 30% κάτω του κανονικού). Αν ο πελάτης έχει δεύτερο αυτοκίνητο έχει επιπλέον 5% μείωση του ασφαλιστρου που θα πλήρωνε σε όλα τα αυτοκίνητα. Αν προκαλέσει ατύχημα αυτή η μείωση εξαλείφεται. Θέλουμε όταν δημιουργείται ένα ασφάλιστρο να γίνεται αυτόματος υπολογισμός του ασφαλιστρου.

```
//ΥΠΟΛΟΓΙΣΜΟΣ ΑΣΦΑΛΙΣΤΡΩΝ
$poson_asfalistrou = 0;
$count_cars = 0;

foreach($xml2->autokinito as $autokinito)
{
if($autokinito->arithmos_kykloforias_autokinitou ==
$_REQUEST['autokinita_pelati_symvolaiou']) {

foreach($xml3->kategoria as $katigoria) {
$res = strcmp($katigoria->evros_kyvikwn_katigorias,
$autokinito->kategoria_autokinitou);

if($res == 0 ) {
(float)$poson_asfalistrou = (float)$katigoria-
>asfalistra_eksaminou_katigorias*2.0;
}
}

foreach($xml4->pelati as $pelati) {
$res2 = strcmp($pelati->arithmos_tautotitas_pelati,
$autokinito->at_pelati_autokinitou);

if($res2 == 0 ) {
if($pelati->neos_odigos_pelatis == "Ναί") {
(float)$poson_asfalistrou = (float)$poson_asfalistrou + 20 *
(float)$poson_asfalistrou / 100;
}
}
}

foreach($xml2->autokinito as $autokinito2) {
$res3 = strcmp($autokinito2-
>at_pelati_autokinitou,$autokinito->at_pelati_autokinitou);

if($res3 == 0 ) {
$count_cars++;
}
}

if($count_cars > 1) {
$poson_asfalistrou = $poson_asfalistrou - 5.0 *
$poson_asfalistrou / 100;
}

$poson_symvolaiouTag = $xml->createElement("poson_symvolaiou",
$poson_asfalistrou);
```

```

$autokinito->asf_symvolaiio = 1;
$xml2->asXML("../assets/database/Autokinita.xml");
}
}
    
```

4.5.8.3 Επεξεργασία ενός Ασφαλιστηρίου Αυτοκινήτου

Η Επεξεργασία ενός Ασφαλιστηρίου Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2.

The screenshot shows a web application interface for Car Insurance. On the left is a navigation menu with options like Dashboard, Υπαλλήλοι, Τμήματα, Πελάτες, Αυτοκίνητα, Κατηγορίες Αυτοκινήτων, Ασφ. Συμβολαία (highlighted), Ανανέωση Συμβολαίων, Ατυχήματα, Συγκεντ. Αναφορές, and Μηνύματα Επικοινωνίας. The main content area is titled 'Επεξεργασία υπάρχον Ασφαλιστηρίου Συμβολαίου!'. It contains a form with the following fields:

- Κωδικός Ασφ. Συμβολαίου: 3
- Επιλέξτε: AEK1234
- Ημερομηνία Συμβολαίου: 01/03/2018
- Ποσόν Συμβολαίου: 487.2

At the bottom of the form are two buttons: 'ΚΑΘΑΡΙΣΜΟΣ' and 'ΑΠΟΘΗΚΕΥΣΗ'. The footer of the application shows 'Copyright © 2015 Car Insurance All rights reserved.' and 'Design and Developed by Car Insurance Dev Team'.

Εικόνα 40 - Φόρμα Επεξεργασίας ενός Ασφαλιστηρίου Αυτοκινήτου

Ο κώδικας υπολογισμού του ποσού, παραμένει ο ίδιος.

4.5.8.4 Διαγραφή κόμβου ενός Ασφαλιστηρίου Αυτοκινήτου

Η Διαγραφή ενός Ασφαλιστηρίου Αυτοκινήτου, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

4.5.9 Ανανέωση Συμβολαίων

Πατώντας από το μενού αριστερά το Ανανέωση Συμβολαίων, μας εμφανίζει μία λίστα με όλους τις Ανανεώσεις Συμβολαίων της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Εικόνα 41- Μενού Ανανέωσης Συμβολαίων

Οι δυνατότητες που έχουμε εδώ είναι μόνο η ανανέωση συμβολαίου.

4.5.9.1 Αρχείο XML για τις Ανανεώσεις Συμβολαίων

Το αρχείο της βάσης δεδομένων για τις Ανανεώσεις Συμβολαίων βρίσκεται στο `asfaleies/admin/assets/database/Asf-Symvolaio.xml`.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

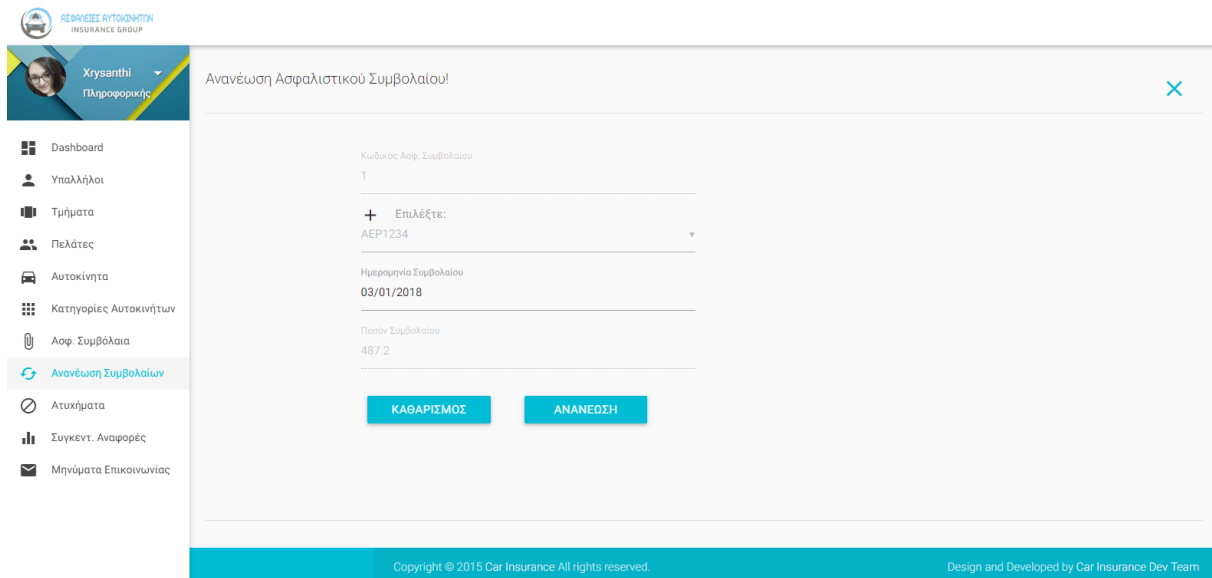
Το `Asf-Symvolaio.xml` περιέχει τις εξής ετικέτες:

```
<symvolaia>
  <symvolaio>
    <id_symvolaio>1</id_symvolaio>
    <autokinita_pelati_symvolaio>ΑΕΚ1234</autokinita_pelati_symvolaio>
    <pelatis_symvolaio>Αλέκος
    Παπαγιαννάκης</pelatis_symvolaio>
    <kategoria_autokinitou_symvolaio>B (301-785)</kategoria_autokinitou_symvolaio>
    <imerominia_symvolaio>03/01/2018</imerominia_symvolaio>
    <imerominia_ananewsisis_symvolaio>03/01/2018</imerominia_ananewsisis_symvolaio>
    <poson_symvolaio>487.2</poson_symvolaio>
    <yfef_ypallilos_symvolaio>Γιώργος
    Ποντικάκης</yfef_ypallilos_symvolaio>
  </symvolaio>
</symvolaia>
```

Δεν έχει δικό του αρχείο xml και ο λόγος είναι ότι το Ανανέωση Συμβολαίου, αλλάζει τα ασφάλιστρα μόλις περάσει ένας χρόνος.

4.5.9.2 Ανανέωση ενός Συμβολαίου

Στην ανανέωση συμβολαίου το μόνο που είναι ενεργό είναι η ημερομηνία. Στην ουσία η ανανέωση συμβολαίου γίνεται κάθε χρόνο και μόλις πατήσουμε το κουμπί «Ανανέωση», γίνεται και αυτόματα η αλλαγή του ποσού στα Ασφάλιστρα.



Εικόνα 42 - Φόρμα Ανανέωσης Συμβολαίου

Για την ανανέωση δέχεται σαν όρισμα το id, όπως γίνεται και στην επεξεργασία. Ο κώδικας ανανέωσης είναι ο εξής:

```
if(isset($_REQUEST['action'])) {
    foreach($xml->symvolaio as $symvolaio)
    {
        if($symvolaio->id_symvolaiou == $value_id) {
            $symvolaio->imerominia_ananewsisis_symvolaiou =
            $_REQUEST['imerominia_ananewsisis_symvolaiou'];

            $date1=strtotime($read_autokinita_pelati_symvolaiou);
            $date2=strtotime($symvolaio->imerominia_ananewsisis_symvolaiou);
            $diff=$date2-$date1;
            $x=floor($diff/3600/24);

            if($x>=365 && $x<730) {
```

```

        $symvolaio->poson_symvolaiou =
$read_poson_symvolaiou - 10.0 * $read_poson_symvolaiou/100;
    }

    else if($x>=730 && $x<1095){
        $symvolaio->poson_symvolaiou =
$read_poson_symvolaiou - 20.0 * $read_poson_symvolaiou/100;
    }

    else if($x>=1095){
        $symvolaio->poson_symvolaiou =
$read_poson_symvolaiou - 30.0 * $read_poson_symvolaiou/100;
    }
}

$xml->asXML("../assets/database/Asf-
Symvolaia.xml");
header('Location: ../ananeosi-symv.php');
}
}

```

4.5.10 Ατυχήματα

Πατώντας από το μενού αριστερά το Ατυχήματα, μας εμφανίζει μία λίστα με όλα τα Ατυχήματα της εταιρείας σε ένα αναλυτικό πίνακα, με όλα τα στοιχεία τους.

Id	Κωδ. Αυτοκ.	Πελάτης	Κατ. Αυτοκ.	Ημ/νία Ατυχήματος	Ώρα Ατυχήματος	Οδός	Αριθμός	Πόλη - Περιοχή	Νομός	ΤΚ	Ποσό Υπαίτιου	Ποσό Αναπαίτιου	Υπαί Πελά
1	AZH4356	Γιώργος Παυλάκης	A (Εως 300)	17/01/2018	20:30	Λ. 62 Μαρτύρων	56	Ηράκλειο	Ηράκλειο	71303	1500	1000	Ναι
2	AZH4356	Αλέκος Παπαγιαννάκης	B (301-785)	06/02/2018	18:30	Κνωσσού	3	Ηράκλειο	Ηράκλειο	71303	3000	1500	Ναι
3	AEP1234	Σοφοκλής Καραγιώργος	E (1.358-1.548)	15/03/2018	22:15	Διεύθυνση	1	Ηράκλειο	Ηράκλειο	71303	800	500	Ναι
4	AKI1687	Κατερίνα Παπαστράτου	Δ (1.072-1.357)	08/02/2018	05:15	Διεύθυνση	1	Ηράκλειο	Ηράκλειο	71303	1400	3500	Ναι

Εικόνα 43 - Μενού Ατυχήματα

Οι δυνατότητες που έχουμε εδώ είναι η Εισαγωγή ενός νέου Ατυχήματος, η Επεξεργασία ενός υπάρχον και η Διαγραφή. Λειτουργίες τις οποίες είδαμε αναλυτικά στην προηγούμενη ενότητα.

4.5.10.1 Αρχείο XML για τα Ατυχήματα

Το αρχείο της βάσης δεδομένων για τα Ατυχήματα βρίσκεται στο asfaleies/admin/assets/database/Atyximata.xml.

Μέσα από το συγκεκριμένο λοιπόν αρχείο θα γίνονται όλες οι αλλαγές, είτε αποθήκευσης νέας εγγραφής, είτε επεξεργασίας υπάρχον είτε διαγραφής ενός ολόκληρου κόμβου.

Το Atyximata.xml περιέχει τις εξής ετικέτες:

```
<atyximata>
  <atyxima>
    <id_atyximatos>1</id_atyximatos>
    <ak_autok_atyximatos>AZH4356</ak_autok_atyximatos>
    <pelatis_atyximatos>Γιώργος Παυλάκης</pelatis_atyximatos>
    <katigoria_autokinitou_atyximatos>A (Έως
300)</katigoria_autokinitou_atyximatos>
    <imerominia_atyximatos>17/01/2018</imerominia_atyximatos>
    <wra_atyximatos>20:30</wra_atyximatos>
    <dief_odos_atyximatos>Λ. 62
Μαρτύρων</dief_odos_atyximatos>
    <dief_arithmos_atyximatos>56</dief_arithmos_atyximatos>
    <dief_poli_atyximatos>Ηράκλειο</dief_poli_atyximatos>
    <dief_nomos_atyximatos>Ηράκλειο</dief_nomos_atyximatos>
    <dief_tk_atyximatos>71303</dief_tk_atyximatos>
    <zimia_pelati_atyximatos>1500</zimia_pelati_atyximatos>
    <zimia_allou_odigou_atyximatos>1000</zimia_allou_odigou_a
tyximatos>
    <yfefthinos_atyximatos>Ναι</yfefthinos_atyximatos>
    <yfef_ypallilos_atyximatos>Χρυσάνθη
Παναγιωτάκη</yfef_ypallilos_atyximatos>
  </atyxima>
</atyximata>
```

4.5.10.2 Εισαγωγή ενός νέου Ατυχήματος

Η Εισαγωγή ενός νέου Ατυχήματος, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.1.

Κωδικός Ατυχήματος

+ Επιλέξτε: Αρ. Κυκλοφ. Αυτοκινήτου

Οδηγός: Αυτοκινήτου

Ημέρια Ατυχήματος Ημερ. Ατυχήματος

Διεύθυνση Ατυχήματος:

Οδός Αριθμός

Πόλη - Περιοχή Νομός

Τ.Κ.

Εξόδα Ατυχήματος:

Πιστώρ Σημάς Πελάτη
0

Πιστώρ Σημάς Άλλου Οδηγού
0

Είναι υπεύθυνος ο Πελάτης;

ΚΑΘΑΡΙΣΜΟΣ ΑΠΟΡΗΚΕΥΣΗ

Copyright © 2015 Car Insurance. All rights reserved. Design and Developed by Car Insurance Dev. Team

Εικόνα 44 - Φόρμα Εισαγωγής ενός νέου Ατυχήματος

4.5.10.3 Επεξεργασία ενός Ατυχήματος

Η Επεξεργασία ενός Ατυχήματος, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.2.

Επεξεργασία υπάρχουν Ατυχήματος!

Κωδικός Ατυχήματος
1

+ Επιλέξτε:
AZH4356

Οδηγός Αυτοκινήτου
Γεώργιος Παυλάκης

Ημ/νία Ατυχήματος
17/01/2018

Ωρα Ατυχήματος
20:30

Διεύθυνση Ατυχήματος:

Οδός Λ. 62 Μαρτύριων	Αριθμός 56
Πόλη - Περιφέρεια Ηράκλειο	Νομός Ηράκλειο

Τ.Κ.
71303

Εξοδα Ατυχήματος:

Ποσόν Ζημιάς Πελάτη
1500

Ποσόν Ζημιάς Άλλου Οδηγού
1000

Είναι υπεύθυνος ο Πελάτης:

Copyright © 2015 Car Insurance All rights reserved. Design and Developed by Car Insurance Dev Team

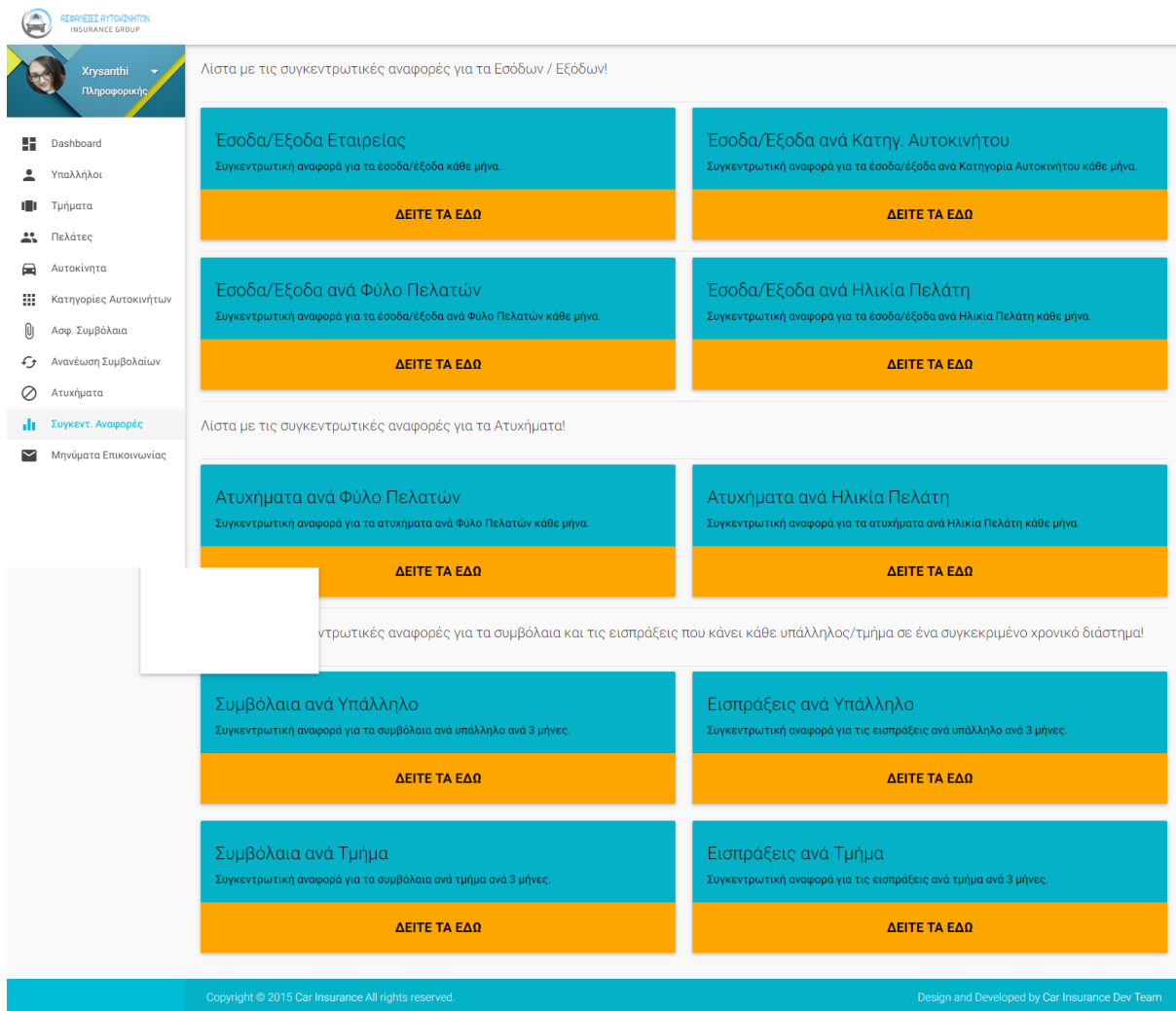
Εικόνα 45 - Φόρμα Επεξεργασίας ενός Ατυχήματος

4.5.10.4 Διαγραφή κόμβου ενός Ατυχήματος

Η Διαγραφή ενός Ατυχήματος, γίνεται ακριβώς με τον ίδιο τρόπο που αναλύσαμε στην Ενότητα 4.4.3.

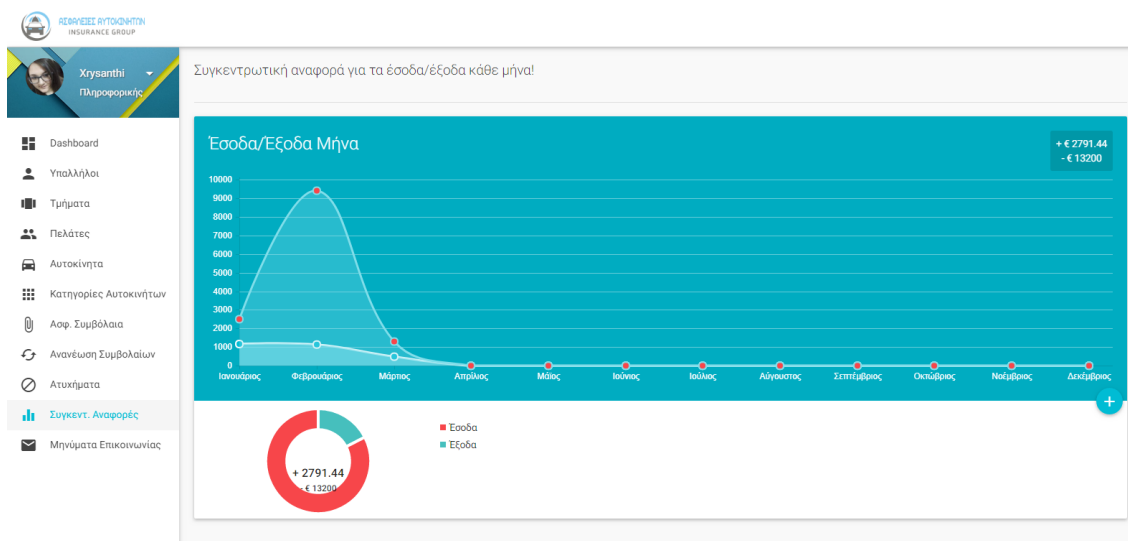
4.5.11 Συγκεντρωτικές Αναφορές

Πατώντας από το μενού αριστερά το Συγκεντρωτικές Αναφορές, μας εμφανίζει μία λίστα με όλες τις συγκεντρωτικές αναφορές.



Εικόνα 46 - Μενού Συγκεντρωτικές Αναφορές

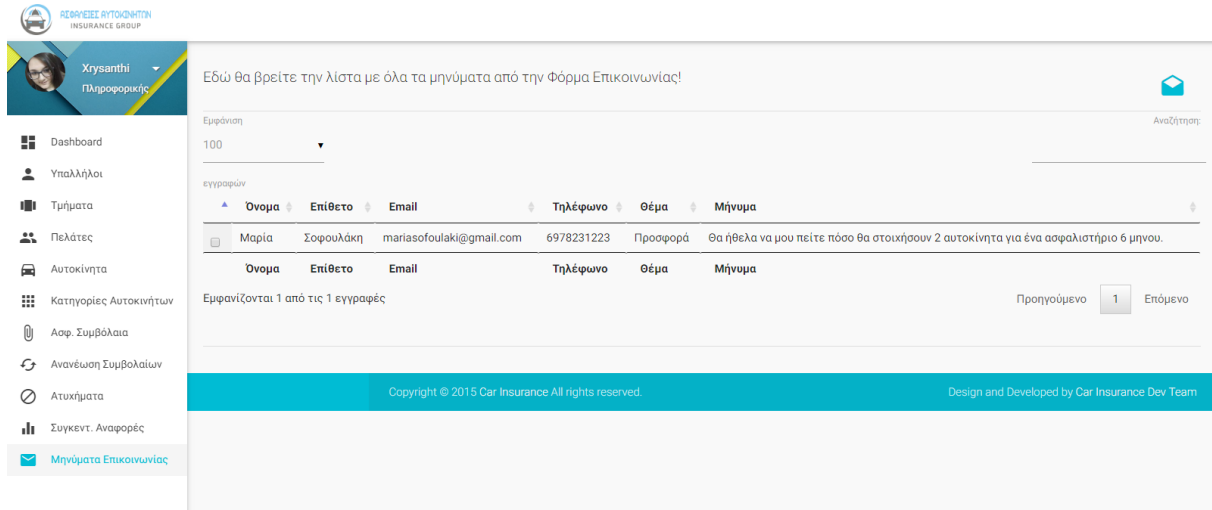
4.5.11.1 Υπόδειγμα μίας συγκεντρωτικής Αναφοράς



Εικόνα 47 - Συγκεντρωτική αναφορά για τα έσοδα/έξοδα κάθε μήνα

4.5.12 Μηνύματα Φόρμα Επικοινωνίας

Πατώντας από το μενού αριστερά το Μηνύματα Επικοινωνίας, μας εμφανίζει μία λίστα με όλα τα μηνύματα που έχουν σταλεί από την φόρμα Επικοινωνίας στο απλό μας ιστότοπο.



Εικόνα 48 - Μενού Μηνύματα Επικοινωνίας

Οι δυνατότητες που έχουμε εδώ η ανάγνωση μόνο των μηνυμάτων.

4.5.12.1 Αρχείο XML για τα Μηνύματα Επικοινωνίας

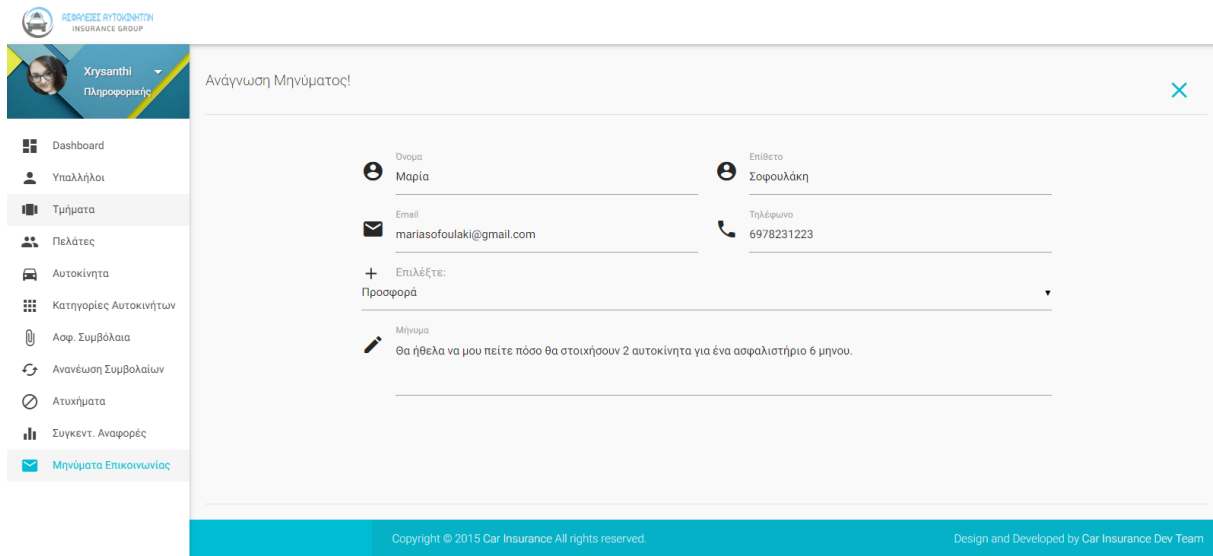
Το αρχείο της βάσης δεδομένων για τα Μηνύματα Επικοινωνίας βρίσκεται στο `asfaleies/admin/assets/database/contact_form.xml`.

Μέσα σε αυτό το αρχείο αποθηκεύονται όλα τα μηνύματα της φόρμας επικοινωνίας.

Το `contact_form.xml` περιέχει τις εξής ετικέτες:

```
<contact>
  <messages>
    <name>Μαρία</name>
    <lname>Σοφουλάκη</lname>
    <email>mariasofoulaki@gmail.com</email>
    <phone>6978231223</phone>
    <select>Προσφορά</select>
    <message>Θα ήθελα να μου πείτε πόσο θα στοιχήσουν 2
αυτοκίνητα για ένα ασφαλιστήριο 6 μηνου.</message>
  </messages>
</contact>
```

4.5.12.2 Ανάγνωση ενός Μηνύματος Επικοινωνίας



Εικόνα 49 - Φόρμα Ανάγνωσης ενός Μηνύματος Επικοινωνίας

Κεφάλαιο 5

Αποτελέσματα

5.1 Συμπεράσματα - Αποτελέσματα

Η πτυχιακή μας εργασία με θέμα «Δικτυοκεντρική Βάση Δεδομένων για μια Ασφαλιστική Εταιρεία» είναι μία πολύ καλή εφαρμογή διαχείρισης για Ασφαλιστικές Εταιρείες. Δημιουργήσαμε έναν δυναμικό ιστότοπο, η οποία χρησιμοποίησε την γλώσσα XML ως Βάση Δεδομένων και την γλώσσα προγραμματισμού ως την βασική γλώσσα διαχείρισης και επεξεργασίας των δυναμικών ιστοσελίδων και της βάσης δεδομένων μας.

Όμως το βασικό κομμάτι της πτυχιακής μας ήταν η εκμάθηση και η χρήση της γλώσσας XML. Καθώς η γλώσσα XML είναι μία σχετικά καινούργια γλώσσα ως βάση δεδομένων, δεν υπήρχαν πολλά παραδείγματα για να μας καθοδηγήσουν. Όμως σίγουρα μπορεί να ανταπεξέλθει σαν μία βάση δεδομένων, καθώς δεν υπάρχει κάτι που δεν μπορεί να κάνει, όπως κάνουν γνωστές βάσεις δεδομένων. Είναι μία γλώσσα που έχει αρκετό μέλλον.

Χαιρόμαστε που μας δόθηκε η δυνατότητα να μάθουμε και να χρησιμοποιήσουμε σύγχρονα και πολύτιμα εργαλεία για την δημιουργία μίας δυναμικής ιστοσελίδας, με περαιτέρω προοπτική για ανάπτυξη και εκπαίδευση πάνω σε αυτόν τον τομέα.

5.2 Βελτιώσεις

Οι βελτιώσεις που μπορούν να γίνουν μελλοντικά στην εφαρμογή μας, είναι αρχικά η ασφάλεια, καθώς όταν πρόκειται για προσωπικά δεδομένα και ευαίσθητα στοιχεία που δεν πρέπει να μπορεί να έχει ο πρόσβαση εύκολα ο καθένας. Επίσης, άλλη μία βελτίωση θα ήταν η ηλεκτρονική υποβολή ενός ασφαλιστηρίου, αυτός που ενδιαφέρεται για ένα ασφαλιστήριο να μπορεί να το κάνει μόνος του online, αυτό βέβαια συνεπάγεται και το σύστημα να δέχεται online πληρωμές.

Βιβλιογραφία

1. Η γλώσσα σήμανσης XML, Δημήτριος Σάμπσον, Πανεπιστημιακές Σημειώσεις Δεκεμβρίου 2003, www.fme.aegean.gr/sites/default/files/dsampson_xml_lectures-notes-dec2003.pdf
2. "Introduction to XML". Course Slides. Pierre Geneves. October 2012
3. Learning XML, Erik T. Ray, 2001
4. XML Bible 1st Edition, Elliotte Rusty Harold, Wiley; 1 edition (August 31, 1999)
5. How the Internet Works (8th Edition) 8th Edition, Preston Gralla, (December 1, 2006)
6. Modern PHP: New Features and Good Practices 1st Edition, Josh Lockhart, O'Reilly Media; 1 edition (March 1, 2015)

Διαδικτυακές Πηγές

1. Wikipedia, 2017, <http://en.wikipedia.org>
2. Ελληνική Wikipedia, 2017, <http://el.wikipedia.org>
3. W3Schools, 2017, <http://www.w3schools.com/>
4. Word Wide Web Consortium, 2017, <http://www.w3.org>
5. XAMPP Apache + MySQL + PHP + Perl, 2015, <https://www.apachefriends.org/index.HTML>
6. Apache HTTP Server, 2017, <http://httpd.apache.org/>