

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ

ΗΡΑΚΛΕΙΟΥ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Πληροφοριακό σύστημα για την
διαχείριση photo gallery
με χρήση XML**

ΟΝΟΜΑ ΣΠΟΥΔΑΣΤΗ: ΚΑΝΤΑΣ ΓΕΩΡΓΙΟΣ

ΟΝΟΜΑ ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗ:

ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΟΥΣΙΑΣΗΣ: 20/12/2013

Πρόλογος

Η παρούσα Πτυχιακή εργασία με τίτλο «Πληροφοριακό σύστημα για την διαχείριση photo gallery με χρήση XML» επιχειρεί να βοηθήσει άτομα που ασχολούνται με το διαδίκτυο και κυρίως με την φωτογραφία, να έχουν ένα σωστό εφόδιο στην αποθήκευση του υλικού τους.

Ανέλαβα την παρούσα Πτυχιακή εργασία τον Οκτώβριο του 2011. Επιβλέπων της Πτυχιακής μου ήταν ο κ. Παπαδάκης Νικόλαος του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Ηρακλείου, τμήματος Πληροφορικής. Με την εργασία αυτή ολοκληρώνεται η φοίτηση μου στο τμήμα Πληροφορικής. Το κυριότερο είναι ότι πλαισιώνεται από τις ανεκτίμητες εμπειρίες που αποκόμισα τα χρόνια των σπουδών μου, αποτελώντας ουσιαστικά τον επίλογο της φοίτησής μου στο Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηρακλείου.

Ξεκινώντας την παρουσίαση της Πτυχιακής μου εργασίας αισθάνομαι την υποχρέωση να εκφράσω τις ευχαριστίες μου σε όλους εκείνους που με βοήθησαν να φέρω εις πέρας το έργο που μου ανατέθηκε. Πρώτα από όλα θα ήθελα να ευχαριστήσω θερμά τον κ. Παπαδάκη Νικόλαο για την ευκαιρία που μου έδωσε να εκπονήσω την παρούσα πτυχιακή εργασία, βοηθώντας με να γνωρίσω το αντικείμενο αυτό, αποκτώντας γνώσεις μέσα από τη συμβολή του. Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές του τμήματος που με δίδαξαν, μου μετέδωσαν τις γνώσεις τους και μου έδωσαν τη δυνατότητα να τις συνθέσω ώστε να πραγματοποιηθεί η παρούσα εργασία.

Τέλος, θα ήθελα να εκφράσω την απέραντη ευγνωμοσύνη και αγάπη στην οικογένεια μου και στους φίλους μου για την κατανόηση και συμπαράστασή τους σε όλη τη διάρκεια των σπουδών μου.

Πίνακας Περιεχομένων

Κεφάλαιο 1ο	7
1.1) Εισαγωγή.....	7
1.2) Περίληψη	7
1.3) Ιστορική Αναδρομή του Διαδικτύου	8
1.4) Ορισμοί και επεξηγήσεις ορολογιών	11
1.4.1) Τί είναι ο Παγκόσμιος Ιστός;.....	11
1.4.2) Τί είναι η ιστοσελίδα (web page), ο ιστότοπος (web site) και το όνομα χώρου (domain name) ;	11
1.4.3) Τί είναι ο browser;	12
1.4.5) Τί είναι το Responsive Web Design;.....	13
1.4.6) Τί είναι μία δυναμική ιστοσελίδα;.....	16
1.5) Κίνητρο για την δημιουργία της εργασίας.....	19
1.6) Κυριότερος στόχος – Γιατί είναι χρήσιμη;	19
1.7) Περιγραφή ιστοσελίδας.....	20
Κεφάλαιο 2ο	26
2.1) Τί είναι οι γλώσσες προγραμματισμού;.....	26
2.1.1) Πώς διαχωρίζονται;.....	26
2.2.2) Από τί αποτελούνται;	27
2.2) Πώς δημιουργήθηκαν; - Ιστορική αναδρομή	27
2.3) Γλώσσα προγραμματισμού XML.....	29
2. 4) Γλώσσα προγραμματισμού HTML	32
2.4.1) Διαφορές XML – HTML.....	33
2.4.2) HTML 5	34
2.5) Γλώσσα προγραμματισμού CSS	35
2.5.1) CSS3.....	36
2.6) Γλώσσα προγραμματισμού Java	37

2.6.1) Χαρακτηριστικά της Java.....	38
2.7) Γλώσσα προγραμματισμού JavaScript.....	40
2.8) Γλώσσα προγραμματισμού asp.net.....	43
2.9) Γλώσσα προγραμματισμού PHP.....	44
3ο Κεφάλαιο.....	47
3.1) Τί είναι οι βιβλιοθήκες;.....	47
3.2) Βιβλιοθήκη jQuery.....	47
4ο Κεφάλαιο.....	49
4.1) Τί είναι η τεχνολογία AJAX;.....	49
5ο Κεφάλαιο.....	50
5.1) Υλοποίηση.....	50
Κεφάλαιο 6ο.....	66
6.1) Συμπεράσματα και μελλοντικές βλέψεις.....	66
Βιβλιογραφία.....	68
Έντυπη.....	68
Ηλεκτρονική.....	69

Πίνακας Εικόνων

Εικόνα 1: Φόρμα εισόδου.....	21
Εικόνα 3: Απεικόνιση του site σε οθόνη κινητού.....	22
Εικόνα 2: Στιγμιότυπο από τη gallery.....	22
Εικόνα 4: Βιογραφικό σημείωμα στο τέλος της σελίδας.....	23
Εικόνα 5: Στιγμιότυπο από τις μικρογραφίες των φωτογραφιών.....	24
Εικόνα 6: Φόρμα διαχείρισης φωτογραφιών.....	24
Εικόνα 7: Φόρμα επιλογής και επεξεργασίας φωτογραφιών.....	25
Εικόνα 8: Τρόπος απεικόνισης των φωτογραφιών στο φυσικό τους μέγεθος.....	25
Εικόνα 9: Φάκελοι και αρχεία του project.....	51

Εικόνα 10: Έλεγχος εισόδου του χρήστη	53
Εικόνα 11: Στιγμιότυπο κώδικα.....	54
Εικόνα 12: Διαχείριση επιλογής εισαγωγής φωτογραφιών	55
Εικόνα 13: Διαχείριση επιβεβαίωσης στοιχείων χρήστη για την εισοδό του στο σύστημα .	56
Εικόνα 14: Αρχείο xml	57
Εικόνα 15: Στιγμιότυπο από τον κώδικα του αλμπουμ	58
Εικόνα 16: Λειτουργικότητα της αναζήτησης φωτογραφιών	59
Εικόνα 17: Χρήση των media queries.....	60
Εικόνα 18: Ελέγχουμε αν υπάρχει το xml αρχείο	61
Εικόνα 19: Μέθοδος της κλάσης DataRow SelectByPictureID.....	62
Εικόνα 20: Δημιουργία του πίνακα στο xml αρχείο.....	63
Εικόνα 21: Αρχείο cb_config.js από το plugin του clearbox.....	64
Εικόνα 22: Web.config αρχείο.....	65

Κεφάλαιο 1ο

1.1) Εισαγωγή

Το 1^ο κεφάλαιο περιλαμβάνει γενικές πληροφορίες για την εν λόγω πτυχιακή εργασία. Παρακάτω ακολουθεί μία περίληψη και παραθέτονται οι λόγοι και τα κίνητρα τα οποία με ώθησαν στην επίλογή του θέματος, καθώς και μία σύνοψη των κεφαλαίων που θα ακολουθήσουν.

1.2) Περίληψη

Το αντικείμενο που πραγματεύεται η συγκεκριμένη πτυχιακή εργασία είναι η κατασκευή μίας ιστοσελίδας με σκοπό την καλύτερη εξυπηρέτηση των χρηστών στην αποθήκευση των φωτογραφιών τους.

Στο 1^ο κεφάλαιο: Όπως προαναφέρθηκε, περιλαμβάνονται κάποιες γενικές πληροφορίες για το διαδίκτυο, ορολογίες, οι λόγοι



χρησιμότητας μίας τέτοιας δημιουργίας καθώς και μία αναλυτική περιγραφή της ιστοσελίδας.

Στο 2ο κεφάλαιο: Σε αυτό το κεφάλαιο γίνεται αναφορά στις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν και για ποιο λόγο επιλέχθηκαν αυτές, και συγχρόνως αναφέρονται εναλλακτικές τεχνολογίες που θα μπορούσαν να είχαν εφαρμοσθεί.

Στο 3ο κεφάλαιο: Παρουσιάζεται ο ορισμός της έννοιας «βιβλιοθήκη» και η βιβλιοθήκη που χρησιμοποιήθηκε.

Στο 4ο κεφάλαιο: Πραγματοποιείται ανάλυση της μεθοδολογίας, η οποία ακολουθήθηκε για την υλοποίηση της παρούσας πτυχιακής εργασίας.

Στο 5ο κεφάλαιο: Στο συγκεκριμένο κεφάλαιο μοιράζομαστε τα συμπεράσματά μου όσον αφορά τη δημιουργία και την πιθανή απήχηση στον

κοινό μία τέτοια εφαρμογή. Επίσης, αναφέρονται κάποιες βλέψεις μου για μελλοντικές βελτιώσεις της ιστοσελίδας που υλοποίησα μέσα από την εκπόνηση της πτυχιακής μου εργασίας.

Ακολουθεί η βιβλιογραφία.

1.3) Ιστορική Αναδρομή του Διαδικτύου

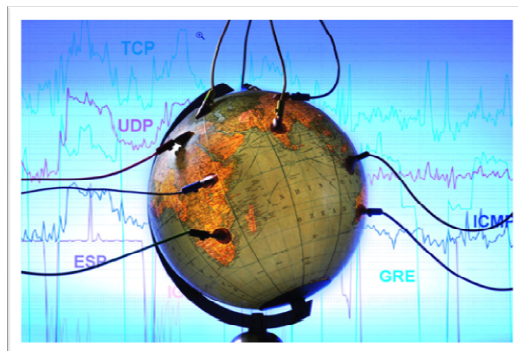
Η λέξη Διαδύκτιο είναι μία σύνθετη λέξη η οποία προέρχεται από τη σύνθεση των λέξεων Διασύνδεση Δικτύων. Ο όρος Internet στα αγγλικά προήλθε από τις λέξεις International Netwoark δηλαδή Διεθνές Δίκτυο Υπολογιστών. Συνοπτικά θα μπορούσε να πει κανείς ότι το Διαδύκτιο είναι ένα σύνολο εκατομμυρίων υπολογιστών οι οποίοι είναι συνδεδεμένοι μεταξύ τους και ανταλλάσσουν δεδομένα.

Πιο αναλυτικά, η ιστορία μας λέει τα εξής: στο τέλος της δεκαετίας του 60 οι ΗΠΑ τρομοκρατούνταν στη σκέψη μίας επερχόμενης πυρηνικής εισβολής από την Ρωσία, η οποία είχε ήδη στείλει στο διάστημα τον δορυφόρο Σπούντικ και ήταν τεχνολογικά πιο ανεπτυγμένη. Έτσι δημιούργησαν την υπηρεσία προηγμένων αμυντικών ερευνών ARPA (Advanced Research Projects Agency), γνωστή ως DARPA (Defense Advanced Research Projects Agency) σήμερα, και της ανέθεσαν την εξής αποστολή: τη δημιουργία ενός δικτύου επικοινωνίας το οποίο θα ήταν ικανό να ανταποκριθεί σε μία πιθανή επίθεση.

Θεμέλιο της δημιουργίας αυτού του δικτύου στάθηκε η θεωρία του Τζ. Λικλάιντερ για το «γαλαξιακό δίκτυο» η οποία υποστήριζε ότι συνδεδεμένοι μεταξύ τους υπολογιστές θα μπορούσαν να ανταλλάξουν πληροφορίες. Το ερώτημα όμως που προέκυψε ήταν αν ένας κόμβος δεχόταν επίθεση ποια θα ήταν η δίοδος επικοινωνίας για τους υπόλοιπους υπολογιστές; Έτσι ο Πολ Μπάραν με το σχεδιασμό ενός κατακεκομμένου δικτύου επικοινωνίας βασισμένο στην ψηφιακή τεχνολογία και ο Λέοναρντ Κλάιρονκ με την θεωρία ανταλλαγής πακέτων βοήθησαν για να δημιουργηθεί το πρώτο είδος διαδικτυακής επικοινωνίας.

Το ARPANET όπως ονομάστηκε, λειτούργησε το 1969 και χρησιμοποιούσε τέσσερεις κόμβους με συνδυασμένους τέσσερεις μίνι υπολογιστές (mini computers 12k), σε δύο πανεπιστήμια στην Καλιφόρνια, στο Στάνφορντ και στη Γιούτα.

Χαρακτηριστικό του δικτύου αυτού ήταν ότι η ταχύτητά του έφθανε τα 50 kbps και έτσι επιτεύχθηκε η πρώτη dial up σύνδεση μέσω γραμμών τηλεφώνου.



Μέσα σε τρία χρόνια οι συνδεδεμένοι υπολογιστές είχαν ξεπεράσει τους είκοσι και τότε δημιουργείται και η πρώτη μορφή ηλεκτρονικού ταχυδρομείου e-mail. Το ARPANET όμως χρησιμοποιούσε το πρωτόκολλο NCP (NETWORK CONTROL PROTOCOL) το οποίο ήταν συμβατό με συγκεκριμένο τύπο υπολογιστών. Για αυτό το λόγο δημιουργήθηκε το πρωτόκολλο TCP (TRANSMISSION CONTROL PROTOCOL) το 1974 από τους Σερφ και Κάαν, το οποίο ένωσε όλα τα δίκτυα που υπήρχαν τότε. Τέσσερα χρόνια αργότερα ήρθε να προστεθεί στο υπάρχον πρωτόκολλο το Internet Protocol (IP), δηλαδή TCP/IP.

Έτος	Γεγονός
1957	Δημιουργία του ARPA
1969	Δημιουργία του ARPAnet
1982	Υιοθέτηση του TCP/IP
1983	Ενσωμάτωση του TCP/IP στο UNIX
1986	Δημιουργία του NSFnet
1990	Δημιουργία του HTTP στο CERN
1992	Κυκλοφορία του MOSAIC
1993	Κυκλοφορία του NETSCAPE NAVIGATOR
1995	Κυκλοφορία του INTERNET EXPLORER

Έτος	Αριθμός υπολογιστών
1977	111
1981	213
1983	562
1984	1.000
1986	5.000
1987	10.000
1989	100.000
1992	1.000.000
2001	150.000.000-175.000.000
2002	>200.000.000
2010	80% του πλανήτη θα είναι στο διαδίκτυο

Το 1984 δημιουργήθηκε το πρώτο DOMAIN NAME SYSTEM DNS το οποίο καταγράφει 1000 κεντρικούς κόμβους και οι συνδεδεμένοι υπολογιστές έχουν από ένα κωδικοποιημένο αριθμό. Στα επόμενα χρόνια ακολούθησαν και άλλα δίκτυα όπως το NSFNET το 1986, το Usenet, το Fidonet και το Bitnet.

Το ότι το Διαδίκτυο έχει τόση μεγάλη απήχηση οφείλεται στο ότι έχει τεράστιες δυνατότητες στην επικοινωνία και στην πληροφόρηση του κοινού για δεκάδες θέματα, είναι πολύ εύκολο και απλό στη χρήση του και επίσης είναι πολύ απλή η διαδικασία σύνδεσης ενός χρήστη να συνδεθεί στο Διαδίκτυο μέσω του απαραίτητου εξοπλισμού που θα διαθέτει. Κλείνοντας την ιστορική αναδρομή να προσθέσω ότι ο όρος Ίντερνετ ξεκίνησε να χρησιμοποιείται την εποχή που συνδέθηκε το ARPANET με το NSFNet και internet σήμαινε οποιοσδήποτε δίκτυο χρησιμοποιούσε το πρωτόκολλο TCP/IP. Το 1989 ο Τιμ Μπέρνερσ-Λι στο κέντρο CERN δημιούργησε τον Παγκόσμιο Ιστό, μία πλατφόρμα που χρησιμοποιείται μέχρι και σήμερα.

1.4) Ορισμοί και επεξηγήσεις ορολογιών

Στον κόσμο του διαδικτύου υπάρχουν πολλοί όροι οι οποίοι είναι δυσνόητοι για όσους δεν γνωρίζουν τη γλώσσα του Ίντερνετ. Για να γίνουν κατανοητά όλα τα οποία θα αναφερθούν παρακάτω αναφέρω κάποιους από τους όρους, τους οποίους θα συναντήσετε αρκετά στη συγκεκριμένη πτυχιακή εργασία, μαζί με την επεξήγησή τους.

1.4.1) Τί είναι ο Παγκόσμιος Ιστός;

Όπως είπαμε και πρωτίτερα είναι ένα δίκτυο με συνδεδεμένους μεταξύ τους υπολογιστές σε αλυσιδωτή κλίμακα (τοπικό δίκτυο-εθνικό δίκτυο-υπερεθνικό δίκτυο) το οποίο χρησιμοποιεί συγκεκριμένα πρωτόκολα επικοινωνίας, το λεγόμενο http. Κάθε δίκτυο-δομική μονάδα του διαδικτύου αποτελείται από συνδεδεμένους υπολογιστές σε τοπικό επίπεδο, όπως για παράδειγμα το δίκτυο υπολογιστών των γραφείων μιας επιχείρησης. Αυτά τα δίκτυα με τη σειρά τους συνδέονται σε ευρύτερα δίκτυα, όπως εθνικά και υπερεθνικά. Η τεχνολογία του ιστού καθιστά δυνατή την δημιουργία "υπερκειμένων", μία διασύνδεση δηλαδή πάρα πολλών μη ιεραρχημένων στοιχείων που παλαιότερα ήταν απομονωμένα. Τα στοιχεία αυτά μπορούν να πάρουν και άλλες μορφές πέραν της μορφής του γραπτού κειμένου, όπως εικόνας και ήχου. Επιπροσθέτως, να αναφέρω ότι ο Παγκόσμιος Ιστός είναι ένα και μοναδικό δίκτυο παγκόσμιας κλίμακας.

1.4.2) Τί είναι η ιστοσελίδα (web page), ο ιστότοπος (web site) και το όνομα χώρου (domain name) ;

Μία ιστοσελίδα (web page) είναι ένα αρχείο πληροφοριών, ένα είδος εγγράφου του παγκόσμιου ιστού το οποίο μπορεί να έχει τη μορφή μιας εικόνας, ενός κειμένου, ήχου ή βίντεο. Αυτό το είδος εγγράφου ανήκει στον παγκόσμιο ιστό και πολλές ιστοσελίδες μαζί αποτελούν έναν ιστότοπο (ιστοχώρος, διακτυακός τόπος, web site) ο

οποίος έχει ένα όνομα χώρου (domain name). Οι ιστοσελίδες συνδέονται μεταξύ τους και μπορεί ο χρήστης να μεταβεί από τη μία στην άλλη, επιλέγοντας συνδέσμους που υπάρχουν στο κείμενο ή στις φωτογραφίες της ιστοσελίδας. Οι σύνδεσμοι προς άλλες σελίδες εμφανίζονται συνήθως υπογραμμισμένοι και με μπλε χρώμα για να είναι ξεκάθαρο στον επισκέπτη ότι πρόκειται για σύνδεσμο προς άλλη ιστοσελίδα, χωρίς όμως πάντα να είναι αυτό απαραίτητο.

1.4.3) Τί είναι ο browser;

Ο browser (*φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού*) είναι το πρόγραμμα με το οποίο οι χρήστες μπορούν να επισκέπτονται τις ιστοσελίδες στο διαδίκτυο. Ουσιαστικά, είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με, κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν υπερσυνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων, για αυτό η εμφάνιση μιας ιστοσελίδας μπορεί να διαφέρει ανάλογα με τον browser.

1.4.4) Τί είναι οι Web developers και οι Web designers;

Οι Web developers και οι Web designers είναι τα άτομα τα οποία είναι υπεύθυνα για αυτό που συναντάμε σαν αναγνώστες όταν επισκεφθούμε μία ιστοσελίδα. Δηλαδή, είναι προγραμματιστές οι οποίοι ασχολούνται με την ανάπτυξη διάφορων εφαρμογών στον ιστότοπο.

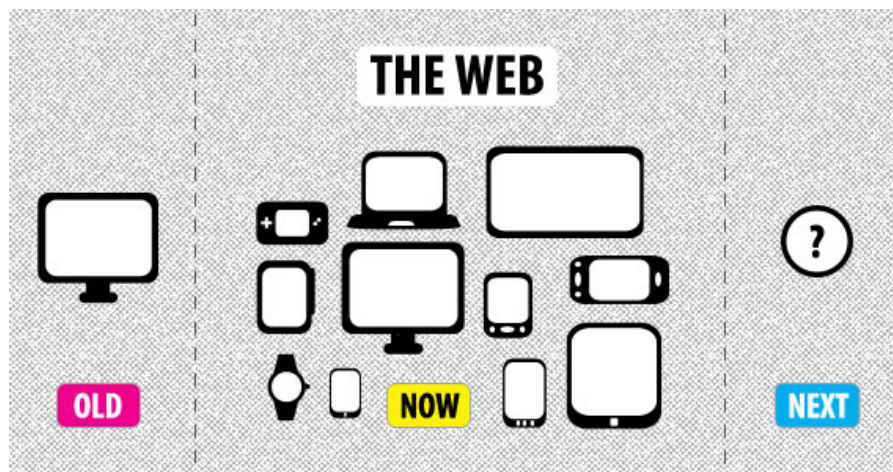
Ας διαχωρίσουμε όμως τις έννοιες Web developers και Web designers, σύμφωνα με ένα άρθρο του Povilas Korop, ενός Web developer από τη Λιθουανία. Ο Korop έθεσε το ερώτημα εάν οι Web developers είναι και Web designers και το αντίστροφο. Και οι δύο ομάδες δημιουργούν websites αλλά η διαφορά έγκυται στο γεγονός ότι οι Web

developers δημιουργούν και γράφουν κώδικες και οι Web designers δημιουργούν με βάση τη φαντασία και το συναίσθημα.



1.4.5) Τί είναι το Responsive Web Design;

Την έννοια responsive site τη γνωρίσαμε πρώτη φορά στο άρθρο του Ethan Marcotte στο alistapart.com το 2010. Οι Web developers και Web designers μέχρι τότε ήταν συνηθισμένοι στο να δημιουργούν ιστοσελίδες και διάφορα designs (σχέδια) σύμφωνα με το εκάστοτε format (σχήμα, διάταξη) και τις διαστάσεις του μέσου που θα τα περιελάμβαναν.



Συμβαδίζοντας όμως με την εξέλιξη της τεχνολογίας παρατηρούμε πως υπάρχει πλέον στην αγορά μεγάλη ποικιλία από συσκευές οι οποίες συνδέονται στο διαδίκτυο, μικρές, μεγάλες, επιτραπέζιες, υπολογιστές, τηλέφωνα, tablets κλπ. των οποίων οι διαστάσεις ποικίλουν. Έτσι ο Marcotte θέλησε να τους «ξυπνήσει» και να τους δείξει ότι πρέπει να δημιουργήσουν μία νέα εποχή στο web design, το Responsive web design και κατ'επέκταση το Responsive Site. Το Responsive Site έχει στόχο την ευελιξία όσον αφορά την αλλαγή μεγέθους, την κύλιση ή την μετακίνηση στο ευρύ φάσμα των συσκευών που υπάρχουν σήμερα, όπως αναφέραμε και παραπάνω. Είναι, δηλαδή ένα web design το οποίο θα προσφέρει στον θεατή – αναγνώστη καλύτερες συνθήκες για πλοήγηση.

Για την ιστορία, πίσω από το responsive κρύβεται το adaptive design και το progressive enhancement, τεχνικές σχεδιασμού, που δεν ήταν καινούριες αλλά αποτέλεσαν τις ρίζες του responsive web design.

Το Responsive Design είναι δηλαδή μία τεχνική ή ένα σύνολο από τεχνικές οι οποίες βοηθούν τους designers και τους επιτρέπουν την δημιουργία σελίδων οι οποίες να προσαρμόζονται σε κάθε οθόνη με το να λαμβάνουν υπόψην τους την ανάλυση της συσκευής στην οποία μπήκε ο χρήστης, και τον προσανατολισμό, (orientation) με τον οποίο κρατάει ο χρήστης τη συσκευή, κάθετα ή οριζόντια (tablet, iphone). Οι τεχνικές αυτές είναι οι παρακάτω:

✓ Fluid grids

Με τον όρο fluid grids αναφερόμαστε στην ιδιότητα που προσδίδουμε σε μια ιστοσελίδα να αποκτά ελαστικότητα και προσαρμοστικότητα σε οθόνες ποικίλων αναλύσεων και μεγεθών. Αυτό σημαίνει ότι τα μέρη που απαρτίζουν μια σελίδα «ρέουν» μέσα σε αυτή ώστε να προσαρμοστούν στο περιβάλλον του χρήστη.

Σε μια ιστοσελίδα δομημένη με τη χρήση adaptive grids ορίζουμε οι ίδιοι τις διαστάσεις και τις συμπεριφορές των στοιχείων της σελίδας μας ανάλογα με τις αναλύσεις της οθόνης. Αντίθετα με τη χρήση fluid grids δεν χρειάζεται να κάνουμε πολλές παρεμβάσεις καθώς τα στοιχεία της σελίδας «ρέουν φυσικά» μέσα στις διαστάσεις του parent container.

✓ Media queries

Τα media queries είναι μια τεχνική ή πιο σωστά ένα module του CSS3 που επιτρέπει στα περιεχόμενα μιας σελίδας να προσαρμόζονται σε διαφορετικές αναλύσεις της οθόνης. Πρόκειται στην ουσία για έναν έλεγχο που γίνεται βάσει συγκεκριμένων κριτηριών (όπως το πλάτος ή το ύψος της οθόνης της συσκευής μας) για το πια μορφή θα πάρουν τελικά τα περιεχόμενα της σελίδας μας. Αν δηλαδή τα κριτήρια που έχουμε ορίσει είναι αληθή τότε γίνεται και η διαμόρφωση των περιεχομένων της σελίδας όπως οι ίδιοι έχουμε ορίσει και επιθυμούμε. Αν και τα media queries κάνανε την εμφάνιση τους το 1994, μόλις το 2012 προτάθηκαν σαν standards από τον διεθνή οργανισμό για τον Παγκόσμιο Ιστό (W3C) έπειτε πλέον και από την επίσημη υποστήριξη τους και από τους browsers.

✓ Flexible images

Η προσαρμογή όμως δεν αναφέρεται μόνο στο πλάτος του περιεχομένου όπως ήταν γνωστό από τα fluid sites (sites που καταλαμβάνουν όλη την οθόνη)· στο συγκεκριμένο format, είναι δυνατόν να γίνει αλλαγή στη δομή, στις εικόνες, στα οπτικά μέσα, στα κείμενα και γενικά ότι περιλαμβάνει η σελίδα. Η ιστοσελίδα, λοιπόν προσαρμόζεται ανάλογα με την ανάλυση της εκάστοτε οθόνης και ο χρήστης βλέπει την αντίστοιχη έκδοση του site ή το βλέπει να αλλάζει δυναμικά μορφή και δομή όταν μικρύνει το μέγεθος του παραθύρου του browser του.

No. 306

ARTICLES • TOPICS • ABOUT • CONTACT • CONTRIBUTORS

MAY 25, 2010

Responsive Web Design
by **ETHAN MARCOTTE**

Published in: CSS, Layout, User Interface Design

Discuss this article • Share this article

Adapt, respond and overcome

“The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page. We should embrace the fact that the web doesn't have the same constraints, and design for this flexibility. But first, we must “accept the ebb and flow of things.”

John Allsopp, *“A Dao of Web Design”*

Το άρθρο του Ethan Marcotte στο alistapart.com

1.4.6) Τί είναι μία δυναμική ιστοσελίδα;

Όταν λέμε ότι μία σελίδα είναι δυναμική εννοούμε ότι η σελίδα δημιουργείται δυναμικά την στιγμή που ο χρήστης την επισκέπτεται και αλληλεπιδρά με τον εξυπηρετητή ιστοσελίδων. Ο τρόπος λειτουργίας στηρίζεται πάνω στο δυναμικό μηχανισμό που δημιουργεί την ιστοσελίδα διαβάζοντας τις πληροφορίες που εισάγονται σε ένα περίγραμμα (Template) οι οποίες προέρχονται από βάσεις

δεδομένων και στέλνονται πάλι πίσω στο web browser μέσω του διακομιστή (Apache). Μία δυναμική ιστοσελίδα επιτρέπει μέσω σελίδων διαχείρισης την εισαγωγή, την ανανέωση και την διαγραφή πληροφοριών και έτσι προσαρμόζει το περιεχόμενό της και την εμφάνισή της σύμφωνα με τις διεργασίες που υφίσταται. Άρα, παρόλο που η κατασκευή μίας δυναμικής ιστοσελίδας κοστίζει λίγο παραπάνω, μας είναι πολύ χρήσιμη αν θέλουμε μία δυναμική παρουσία στο διαδίκτυο διότι μας επιτρέπει να αλλάζουμε τις πληροφορίες της άμεσα.

Ως πλεονεκτήματα μιας δυναμικής ιστοσελίδας μπορούν να θεωρηθούν τα παρακάτω:

- ✓ δημιουργία παρουσίας επιχείρησης στο διαδίκτυο
- ✓ αύξηση αναγνωσιμότητας επιχείρησης ή/και προϊόντων
- ✓ προώθηση επιχείρησης ή/και προϊόντων μέσω διαφημίσεων και διαδικτυακών εκστρατειών
- ✓ αύξηση πωλήσεων, των πελατών και των συνεργατών καθώς και την υποστήριξη αυτών
- ✓ εύκολη επικοινωνία με πελάτες
- ✓ παροχή πληροφοριών σχετικά με νέα και ανακοινώσεις για την επιχείρηση ή/και τα προϊόντα

1.4.7) Τί είναι μία στατική ιστοσελίδα;

Μία στατική ιστοσελίδα θα μπορούσαμε να πούμε ότι είναι ένα μόνιμο σύνολο πληροφοριών το οποίο δεν μεταβάλλεται. Το περιεχόμενο και η διάταξή τους είναι δυνατόν να τροποποιηθούν μόνο μετά από αίτημα του προγραμματιστή για αναβάθμιση. Μια απλή σελίδα html (έγγραφο html) που περιλαμβάνει κείμενο, συνδέσμους και φωτογραφίες για παράδειγμα, είναι ένα απλό παράδειγμα στατικής σελίδας. Τα δεδομένα σε μια στατική ιστοσελίδα δεν αλλάζουν δυναμικά. Έχουν σταθερό, αμετάβλητο περιεχόμενο. Τα πλεονεκτήματά της είναι ότι είναι οικονομική στην κατασκευή της διότι είναι πιο απλή, είναι πιο γρήγορη και χρειάζεται μικρό χώρο στο webserver.

Μεγάλο πλεονέκτημα των στατικών ιστοσελίδων είναι επίσης ότι είναι πιο «SEO friendly» . Το SEO σημαίνει Search engine optimization και στα ελληνικά «Βελτιστοποίηση Σελίδας για τις μηχανές αναζήτησης», δηλαδή, η στατική ιστοσελίδα «φορτώνει» πιο γρήγορα. Αυτό συμβαίνει γιατί στις στατικές ιστοσελίδες υπάρχει δυνατότητα τοποθέτησης στον κώδικα html, των meta tags τα οποία αναγνωρίζουν οι μηχανές αναζήτησης και κατατάσσουν την στατική σελίδα πάνω από μια δυναμική που δεν έχει ακριβώς αυτή την δυνατότητα. Αυτό είναι πολύ ισχυρό πλεονέκτημα των στατικών ιστοσελίδων, γιατί δυνδύεται με τον όγκο των επισκέψεων των χρηστών στην συγκεκριμένη ιστοσελίδα και άρα αυξάνει το κέρδος. Οι μηχανές αναζήτησης είναι αυτές που παραπέμπουν τους χρήστες στις ιστοσελίδες και άρα όσο καλύτερα κατεταγμένη η σελίδα στις μηχανές αναζήτησης σε διάφορες λέξεις-κλειδιά, τόσο μεγαλύτερο πλεονέκτημα έχει έναντι των ανταγωνιστών της. Στην αγορά γίνεται σκληρή μάχη ανάμεσα στους SEO experts για διάφορες λέξεις - κλειδιά. Είναι αξιοσημείωτο, ότι μόνο στην ελληνική αγορά για λέξεις - κλειδιά όπως fashion, gadgets, ρούχα κ.λπ. για να επιτευχθεί η πρώτη θέση στις μηχανές αναζήτησης, δαπανούνται μέχρι και 20 χιλιάδες ευρώ.

Στα κατά μιας στατικής ιστοσελίδας είναι ότι χρήστης πρέπει να έχει γνώσεις προγραμματισμού, για να μπορεί να ανανεώσει το περιεχόμενο της σελίδας. Κι αυτό γιατί θα πρέπει να επεμβαίνει στον κώδικα. Σε μια στατική ιστοσελίδα ο χρήστης δεν έχει κανένα είδους αλληλεπίδραση με τη σελίδα και όποια άλλα μειονεκτήματα συνεπάγονται σε αυτό. Η επιλογή μιας στατικής ιστοσελίδας είναι η κατάλληλη εάν κανείς δεν έχει σκοπό να ανανεώνει συχνά τις σελίδες, οι οποίες δεν θα ξεπεράσουν



τις 15 - 20. Οι στατικές ιστοσελίδες ενδείκνυνται για μικρές επιχειρήσεις που θέλουν να κάνουν μια απλή παρουσίαση των προϊόντων τους στο διαδίκτυο, χωρίς να θέλουν να τα ανανεώνουν συχνά. Η κατασκευή μιας στατικής ιστοσελίδας από τους web designers έχει χαμηλότερο κόστος, σε σχέση με τις δυναμικές, και αυτό είναι λογικό γιατί είναι πιο απλή η κατασκευή τους.

1.5) Κίνητρο για την δημιουργία της εργασίας

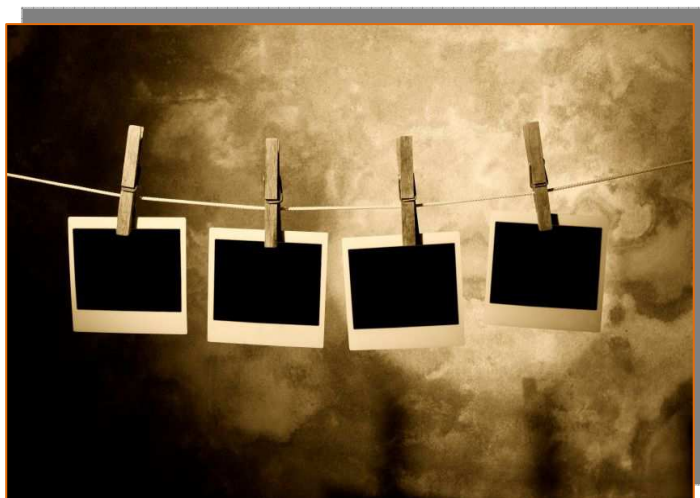
Στην εποχή που ζούμε, όλοι έχουμε αντηληφθεί ότι είναι πλέον πάρα πολύ εύκολο να απαθανατίσουμε τις αγαπημένες μας στιγμές οποιαδήποτε στιγμή θέλουμε, μιας και κάμερες συναντάμε σε όποια συσκευή και αν διαθέτουμε, κινητό τηλέφωνο, υπολογιστή, λάπτοπ, tablet κλπ.

Το πρόβλημα όμως που υπάρχει είναι ότι τις περισσότερες φορές ο όγκος των φωτογραφιών που διαθέτουμε είναι αρκετά μεγάλος και προκύπτει το πρόβλημα της αποθήκευσης. Η εκτύπωση όπως θα γνωρίζουμε πολλοί από εμάς δεν συμφέρει αλλά και πάλι υπάρχει ο φόβος της αλλοίωσης ή της απώλειας μίας φωτογραφίας. Έτσι δημιουργήθηκε η ιδέα για ένα ηλεκτρονικό άλμπουμ στο οποίο θα υπάρχει ένας χώρος, μία περιοχή, για όποιον το επιθυμεί, στον οποίο θα υπάρχουν οι φωτογραφίες του.

1.6) Κυριότερος στόχος – Γιατί είναι χρήσιμη;

Ο απότερος στόχος της πτυχιακής εργασίας είναι να δημιουργηθεί μία ιστοσελίδα για το κοινό, στην οποία οι επισκέπτες θα μπορούν να δημιουργήσουν τον προσωπικό τους λογαριασμό για να αποθηκεύουν το υλικό τους.

Ουσιαστικά είναι ένα διαδικτυακό, ψηφιακό, ηλεκτρονικό άλμπουμ φωτογραφιών που βοηθάει τους χρήστες να οργανώνουν τις φωτογραφίες τους χωρίς να ανησυχούν μήπως χαθούν ή αλλοιωθούν με το πέρασμα του χρόνου. Τα αρχεία παραμένουν στην περιοχή του χρήστη για όσο καιρό αυτός το επιθυμεί και δεν χρειάζεται να γεμίζει την μνήμη του υπολογιστή του.



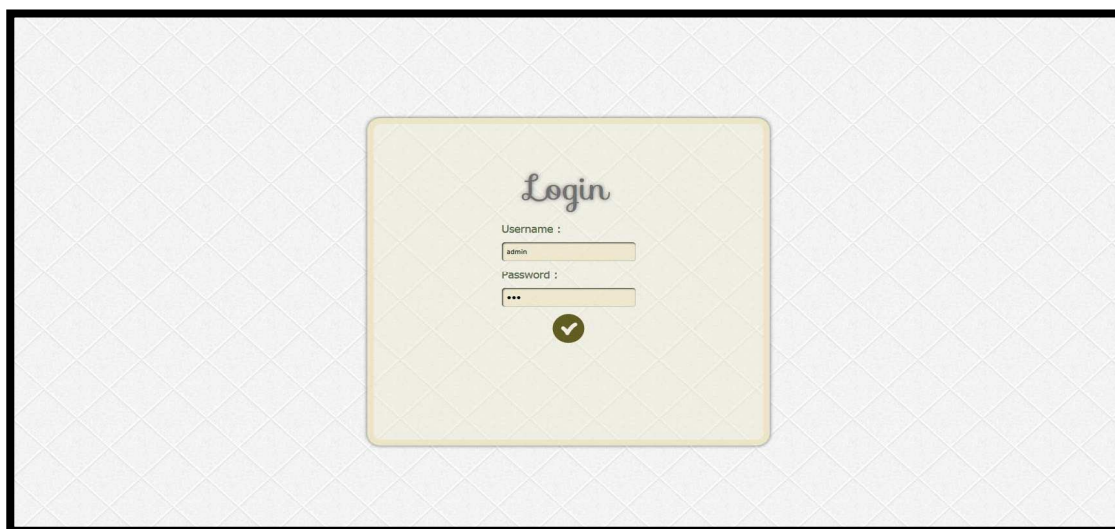
1.7) Περιγραφή ιστοσελίδας

Η συγκεκριμένη ιστοσελίδα, επιτρέπει στον χρήστη να δημιουργήσει την δική του περιοχή στο διαδίκτυο, στην οποία θα ανεβάζει τις φωτογραφίες του, τις οποίες θα βλέπει και θα επεξεργάζεται μόνον ο ίδιος.

Η ιστοσελίδα έχει δημιουργηθεί με τέτοιο τρόπο ώστε να είναι πολύ εύκολο στο χρήστη να ανεβάσει και να επεξεργαστεί τα αρχεία του. Η ιστοσελίδα είναι ανοιχτή για όλο το κοινό και δεν υπάρχει κάποια χρέωση προς τον χρήστη που θα θελήσει να δημιουργήσει τον προσωπικό του λογαριασμό. Τα αρχεία, τα οποία θα ανεβάσει ο εκάστοτε χρήστης, είναι δυνατόν να τα επεξεργαστεί μόνο ως προς το όνομα, δηλαδή να τα μετονομάσει και να γράψει μία σύντομη περιγραφή της κάθε φωτογραφίας. Έτσι, ο χρήστης έχει τη δυνατότητα να αναζητήσει την φωτογραφία που ψάχνει στην περιοχή του μέσω μίας μπάρας αναζήτησης που υπάρχει εάν γράψει εκεί το όνομα του αρχείου ή ακόμα και μια λέξη κλειδί.

Παρακάτω θα δούμε κάποιες φωτογραφίες από το μενού της ιστοσελίδας (screenshot) και μία περιγραφή που θα ακολουθεί κάθε μία από αυτές.

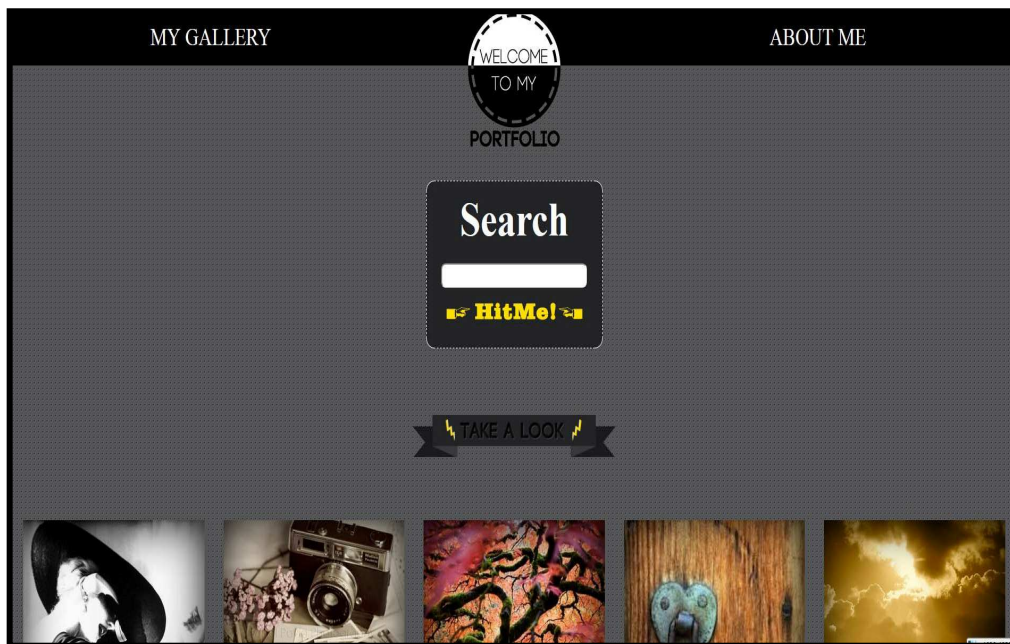
Καταρχήν, βλέπουμε στην εικόνα 1 τη σελίδα στην οποία ο χρήστης αφού έχει δημιουργήσει τον λογαριασμό του κάνει σύνδεση Log in για να εισέλθει στην περιοχή με τις φωτογραφίες του. Καταχωρεί το όνομα χρήστη και τον κωδικό του στα πεδία που του υποδεικνύονται και έτσι συνδέεται.



Εικόνα 1: Φόρμα εισόδου

Η εργασία μου είναι δομημένη υπό την μορφή μίας single page ιστοσελίδας, δηλαδή, μιας ιστοσελίδας της οποίας τα περιεχόμενα παρουσιάζονται σε μία και μοναδική σελίδα. Παρακάτω, βλέπουμε το μενού της ιστοσελίδας μας όπως φαίνεται σε μία οθόνη Η/Υ (εικ. 2) και όπως φαίνεται σε μία οθόνη μικρότερης ανάλυσης (εικ.3).

Στην εικόνα παρατηρούμε πως υπάρχει μία μπάρα με την επιγραφή NAVIGATION (= πλοήγηση) και πατώντας πάνω της βλέπουμε το μενού μας. Στην εικόνα εάν πατήσουμε τον σύνδεσμο πάνω αριστερά MY GALLERY θα μας μεταφέρει στο σύνολο των φωτογραφιών που έχουμε ανεβάσει. Εάν πατήσουμε στο σύνδεσμο πάνω δεξιά ABOUT ME ή αν κυλίσουμε προς τα κάτω τη σελίδα μας, μπορούμε να

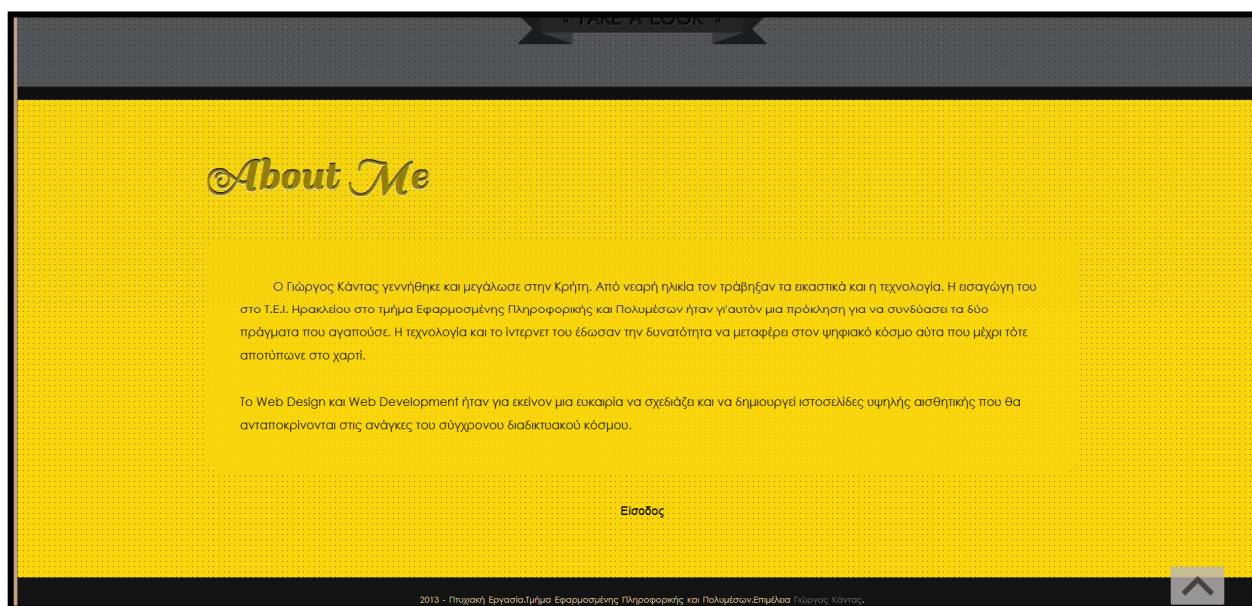


Εικόνα 3: Στιγμιότυπο από τη gallery



Εικόνα 2: Απεικόνιση του site σε οθόνη κινητού

διαβάσουμε ένα σύντομο βιογραφικό για τον δημιουργό της ιστοσελίδας. Βλέπουμε στην εικόνα 4 πως εμφανίζεται στον επισκέπτη. Όπως παρατηρούμε, υπάρχει μία μπάρα αναζήτησης στην οποία μπορούμε να πληκτρολογήσουμε το όνομα με το οποίο έχουμε αποθηκεύσει την φωτογραφία μας και έτσι να την βρούμε αμέσως.



Εικόνα 4: Βιογραφικό σημείωμα στο τέλος της σελίδας

Η συνολική παρουσίαση των φωτογραφιών μας εμφανίζεται με τη μορφή thumbnails δηλαδή μικρογραφιών (εικ. 5). Οι μικρογραφίες αυτές έχουν κοινό μέγεθος χωρίς αυτό να επηρεάζει την ποιότητα εμφάνισης της φωτογραφίας. Για παράδειγμα μία φωτογραφία σε μορφή landscape (τοπίου) και μια φωτογραφία σε μορφή portrait (πορτραίτου) θα μετατραπούν σε μικρογραφίες αναλογικώς χωρίς να υποστούν παραμόρφωση. Ο χρήστης έχει τη δυνατότητα να δει τις φωτογραφίες του στο πραγματικό τους μέγεθος απλά πατώντας πάνω στην αντίστοιχη μικρογραφία της φωτογραφίας. Όταν πλέον εισέλθουμε στην περιοχή μας, εικόνα 6, βλέπουμε τις φωτογραφίες μας και μπορούμε να τις επεξεργαστούμε αλλάζοντάς τους το όνομα αν πατήσουμε το κουμπάκι με τα εργαλεία ή διαγράφοντάς τις πατώντας το κουμπάκι x.



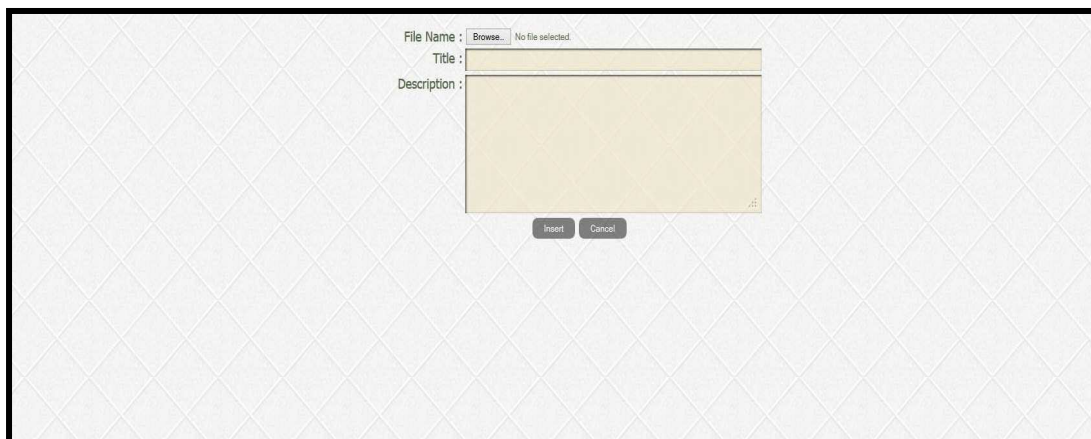
Εικόνα 5: Στιγμιότυπο από τις μικρογραφίες των φωτογραφιών

Picture	Title	Description		
	hgf	korina		
	jhg	jgh		
	hgf	hfg		
	zsddw	xczcz		
	fdshf	ghgjh		
	hjjg	ghfjhg		
	hdsv	ghfjds		
	εγεερ	ρρετες		
	htf	hfdhg		

Εικόνα 6: Φόρμα διαχείρισης φωτογραφιών

Εάν θέλουμε να προσθέσουμε κάποια φωτογραφία στο άλμπουμ μας απλά στην ίδια οθόνη στο κάτω μέρος πατάμε την επιλογή insert .

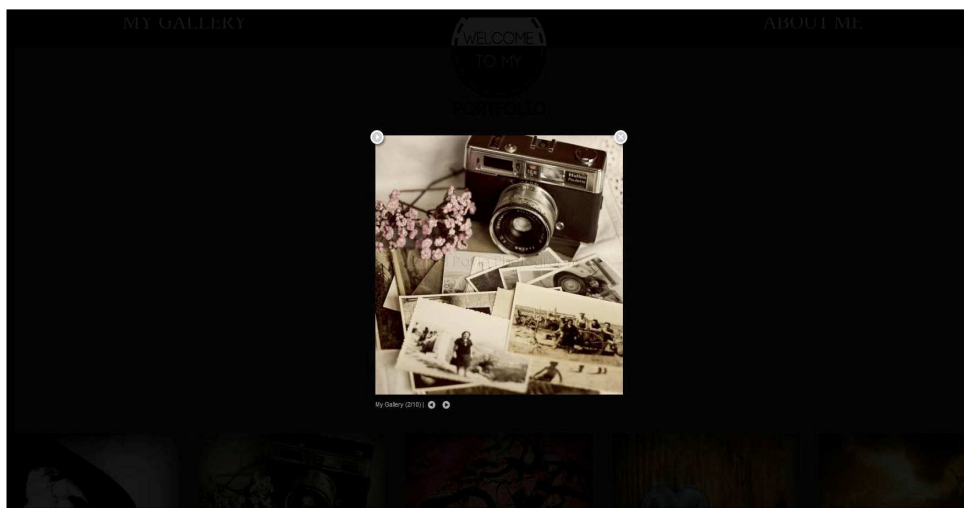
Μας ανοίγει αμέσως η φόρμα επιλογής φωτογραφιών που βλέπουμε στην εικόνα 7.



Εικόνα 7: Φόρμα επιλογής και επεξεργασίας φωτογραφιών

Αρχικά μας ζητάει να αναζητήσουμε (Browse) τον υπολογιστή μας ή στο κινητό μας κλπ. το αρχείο μας. Αφού το εντοπίσουμε και το ανεβάσουμε, μπορούμε να γράψουμε ένα τίτλο (Title) για αυτό το αρχείο π.χ. Γιώργος και μία σύντομη περιγραφή (Description) πχ. Καλοκαίρι στο Παλαίκαстро 2013. Μόλις είμαστε έτοιμη πατάμε Insert (εισαγωγή) αλλά αν θέλουμε να ακυρώσουμε την διαδικασία πατάμε Cancel (ακύρωση).

Η φωτογραφία μας φαίνεται όπως βλέπουμε στην παρακάτω εικόνα 8. Ο χρήστης έχει τη δυνατότητα πλοήγησης σε όλο το άλμπουμ μέσω των αντίστοιχων επιλογών πλοήγησης.



Εικόνα 8: Τρόπος απεικόνισης των φωτογραφιών στο φυσικό τους μέγεθος

Κεφάλαιο 2ο

2.1) Τί είναι οι γλώσσες προγραμματισμού;

Ως γλώσσα προγραμματισμού θα μπορούσαμε να ορίσουμε μία τεχνητή γλώσσα, η οποία περιλαμβάνει τους δικούς της κανόνες σύνταξης και εννοιολογίας και την οποία μπορεί να «καταλάβει» μία μηχανή όπως ένας ηλεκτρονικός υπολογιστής. Οι ανώτερου επιπέδου γλώσσες έουν σύνταξη και λεξιλόγιο που προσεγγίζουν την ανθρώπινη γλώσσα. Έπειτα ο πηγαίος κώδικα μεταφράζεται σε γλώσσα μηχανής από interpreters και compilers.

Ένας interpreter μεταφράζει σε γλώσσα μηχανής άμεσα την εντολή τη στιγμή της εισαγωγής. Της. Ένα compiler μεταφράζει όλο τν κώδικα σε μια ενδιάμεση κατάσταση δίνοντας αποτέλεσμα αρχείο object (*.obj) και στη συνέχεια ο linker μετατρέπο αυτό το αρχείο σε εκτελέσιμο αρχείο παράγοντας έτσι τον κώδικα μηχανής.

Οι γλώσσες αυτές έχουν τη δική τους δομή και τις δικές του προδιαγραφές τυποποίησης για να διατυπωθούν. Οι προγραμματιστές τις χρησιμοποιούν ώστε να οργανώνουν , να διαχειρίζονται πληροφορίες και να διατυπώνουν αλγόριθμους.

2.1.1) Πώς διαχωρίζονται;

Ο διαχωρισμός σε κατηγορίες δεν είναι απόλυτος, καθώς πολλές γλώσσες έχουν χαρακτηριστικά που εμπίπτουν σε περισσότερες από μία κατηγορίες, και γίνεται για οργανωτικούς λόγους. Οι γλώσσες προγραμματισμού που υπάρχουν είναι οι γλώσσες μηχανής, συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου και οι γλώσσες υψηλού επιπέδου.

Οι γλώσσες μηχανής δημιουργήθηκαν για να μπορέσει ο ηλεκτρονικός υπολογιστής να εκτελέσει μία λειτουργία κατευθείαν μέσω του κώδικα. Έπειτα, με τις συμβολικές γλώσσες ή τις γλώσσες χαμηλού επιπέδου ο υπολογιστής θα έπρεπε πριν εκτελέσει την λειτουργία να μεταφράση μέσω ενός συμβολομεταφραστή (assembler) τις εντολές. Οι γλώσσες υψηλού επιπέδου βασίζονται σε λέξεις τις καθομιλουμένης γλώσσας και ο υπολογιστής τις μεταφράζει μέσω ενός μεταγλωτιστή (compiler).

2.2.2) Από τί αποτελούνται;

Μία γλώσσα προγραμματισμού την προσδιορίζει το αλφάβητό της, το λεξιλόγιό της, η γραμματική της και η σημασιολογία της.

Αλφάβητο είναι το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα προγραμματισμού.

Το λεξιλόγιο αποτελείται από ένα σύνολο των στοιχείων του αλφαβήτου.

Η γραμματική αποτελείται από το συντακτικό της και την τυπολογία της.

Η σημασιολογία της είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και αποτελείται από τα παρακάτω στάδια:

- ανάλυση = κατανόηση των δεδομένων,
- σχεδιασμός κώδικα, συγγραφή προγράμματος,
- μεταγλώττιση του κώδικα σε γλώσσα μηχανής, (ο compiler μας εμφανίζει τυχόν λάθη),
- διασύνδεση,
- εκτέλεση προγράμματος και
- έλεγχος της λειτουργίας του.



2.2) Πώς δημιουργήθηκαν; - Ιστορική αναδρομή

Η ιστορία μας λέει πώς υπήρξε ανάγκη δημιουργίας γλώσσας προγραμματισμού πολύ πριν δημιουργηθεί ο πρώτος ηλεκτρονικός και προγραμματίσιμος υπολογιστής. Ένα παράδειγμα θα μπορούσε να ήταν και ο αργαλιός Jacquard το 1801, ο οποίος χρησιμοποιεί διάτρητες κάρτες για το αυτόματο πλέξιμο σχημάτων στα πλεκτά. Κατά τη διάρκεια του 2^{ου} παγκοσμίου πολέμου χρειάστηκε να επιλυθούν κάποια αριθμητικά προβλήματα όπως ο υπολογισμός καμπυλών βολών και εκεί

πρωτοεμφανίζονται οι γλώσσες προγραμματισμού. Αρχικά χρησιμοποιήθηκε η κατώτερη γλώσσα προγραμματισμού ή γλώσσα μηχανής (machine language). Οι εντολές της ήταν ακολουθίες από το 0 και το 1. Έπειτα υπήρξαν οι assemblers που απευθύνονταν στην CPU. Μεταξύ 1842-1843 γράφεται ο πρώτος κώδικας από την Ada Lovelace για μία μηχανή που ονομαζόταν Analytical Engine και την οποία σχεδίαζε και ο μαθηματικός Babbage.

Μόλις δημιουργήθηκε ο πρώτος ηλεκτρονικός υπολογιστής (ENIAC), προέκυψε και η ανάγκη της ύπαρξης ενός τρόπου, ενός κώδικα, για να μπορούν οι προγραμματιστές να γράψουν εφαρμογές γι' αυτόν. Η έλλειψη ταχύτητας και μνήμης, των τότε υπολογιστικών μοντέλων, επέβαλαν στους προγραμματιστές να γράφουν assembly. Πολύ σύντομα όμως, ανακαλύψανε ότι η χρήση της assembly ήταν επίσης προβληματική λόγω της μεγάλης προσπάθειας που επέβαλε η σύνταξη προγραμμάτων με αυτήν καθώς και η απασφαλμάτωσή τους.

Το 1948, ο Konrad Zuse δημοσίευσε ένα άρθρο για την γλώσσα προγραμματισμού που επινόησε, την Plankalkül. Η γλώσσα αυτή όμως δεν υλοποιήθηκε τότε, αλλά μετά από πολλά χρόνια, λόγω πολέμου. Εξίσου σημαντικές γλώσσες που αναπτύχθηκαν εκείνη την πρώτη περίοδο των υπολογιστών ήταν η C-10 και η ENIAC coding system. Όλες οι γλώσσες αυτές ήταν σχεδιασμένες για τον υπολογιστή ENIAC. Από την δεκαετία του 1950 δημιουργούνται τρεις μοντέρνες γλώσσες προγραμματισμού. Οι γλώσσες αυτές είναι: η COBOL (COmmon Business Oriented Language) η LISP (LIStProcessor) και η FORTRAN (FORmula TRANslation). Συγχρόνως δημιουργείται και η πρώτη έκδοση της ALGOL (ALGOrithmic Language), η ALGOL-60, η οποία εισήγαγε τα εξής δύο νέα στοιχεία: την ύπαρξη μπλόκ κώδικα και την ύπαρξη ξεχωριστής εμβέλεια μεταβλητών μέσα στα μπλόκ αυτά.

Επίσης άλλη μία καινοτομία που συνόδευε την ALGOL ήταν η δημιουργία ενός φορμαλισμού για την περιγραφή του συντακτικού των γλωσσών προγραμματισμού, η λεγόμενη BNF μορφή (Backus-Naur Form). Μέσα στις δύο αυτές δεκαετίες, και με τις βάσεις να έχουν ήδη οριστεί με τις υπολοιήσεις των παραπάνω γλωσσών, μέχρι το τέλος της δεκαετίας του 1960, ήδη αρχίζουν οι γλώσσες προγραμματισμού να πληθαίνουν και να εισάγει η καθεμία νέες δυνατότητες. Άλλες σημαντικές γλώσσες που αναπτύχθηκαν ήταν: η APL, η Simula και η BASIC.

Στη συνέχεια, ξεκίνησε και ο σχεδιασμός της γλώσσας Pascal. Την περίοδο 1967 με 1978 δημιουργούνται γλώσσες όπως η C, η ML και η Prolog, και δημιουργούνται

πλέον οικογένειες γλωσσών από τις οποίες κάθε σημερινή μοντέρνα γλώσσα έχει τουλάχιστον έναν πρόγονο. Λίγο αργότερα το δημιουργείται η πρώτη αντικειμενοστρεφής γλώσσα, η C++ βασισμένη σε τεράστιο βαθμό πάνω στην ήδη υπάρχουσα, C. Επιπροσθέτως, ένα νέο προγραμματιστικό στυλ, το να γράφει κανείς προγράμματα σε κομμάτια (modules), καταλήγει στην δημιουργία γλωσσών όπως η Modula. Άλλες γλώσσες που αναπτύχθηκαν είναι: η ADA, η Eiffel και η γνωστή Perl.

Η δεκαετία του '90 είναι η εποχή του Ίντερνετ και δημιουργούνται νέες ανάγκες στις γλώσσες προγραμματισμού. Γίνεται μεγάλη προσπάθεια να βοηθηθούν οι προγραμματιστές στο να γράφουν γρήγορα και εύκολα κώδικα, κάτι που οδηγεί στην δημιουργία των πρώτων IDE. Αναπτύσσονται οι γλώσσες C# και Visual BASIC της Microsoft.

Μέρα, με τη μέρα η τεχνολογία προχωράει και η εξέλιξη των γλωσσών αναπτύσσεται. Οι παλιές περνάνε στο παρασκήνιο και νέες με νέα χαρακτηριστικά ανάλογα με το τί ανάγκες προκύπτουν έρχονται στην επιφάνεια.

2.3) Γλώσσα προγραμματισμού XML

Για την εκπόνηση της εργασίας είναι απαραίτητη η χρήση γλωσσών προγραμματισμού. Μία από αυτές που χρησιμοποιήθηκαν είναι η XML. Είναι πολύ απλή αλλά χρήσιμη και όπως θα δούμε παρακάτω προσφέρει πολλά οφέλη στους προγραμματιστές.

Το ακρωνύμιο XML, συνθέτουν οι λέξεις Extensive Markup Language και η οποία είναι μία γλώσσα σήμανσης. Αναπτύχθηκε από μία ομάδα εργασίας XML υπό την επίβλεψη του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems με την ενεργή συμμετοχή μιας XML Ομάδας Ειδικού Ενδιαφέροντος (που οργανώθηκε από τον οργανισμό W3C).

Η σήμανση είναι η διαδικασία χρήσης κωδικών, οι οποίοι ονομάζονται ετικέτες για τον ορισμό της δομής, της οπτικής εμφάνισης και στην συγκεκριμένη γλώσσα του

νοήματος των δεδομένων. Στην XML περιέχεται ένα σύνολο κανόνων που αφορά την ηλεκτρονική κωδικοποίηση κειμένων. Οι ετικέτες στα έγγραφα της γλώσσας αυτής περικλείονται από τα σύμβολα μικρότερο από και μεγαλύτερο από (<, >). Είναι μία μορφοποίηση δεδομένων κειμένου, με την υποστήριξη του Unicode για όλες τις γλώσσες του κόσμου. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Σχεδιάστηκε με τέτοιο τρόπο έτσι ώστε να είναι απλή και χρήσιμη για το διαδίκτυο. Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Αποτελούνται από μονάδες αποθήκευσης που καλούνται entities (οντότητες), οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από χαρακτήρες (characters) οι οποίοι συνθέτουν character data και άλλοι οι οποίοι συνθέτουν markup. Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή. Οι χαρακτήρες που συμπεριλαμβάνονται σε ένα κείμενο XML αποτελούν τη σήμανση ή το περιεχόμενό του, τα οποία μπορούν ύστερα από την εφαρμογή κάποιων κανόνων να επισημανθούν και να διακριθούν.

Επιτρέπει στους προγραμματιστές να δημιουργήσουν οποιαδήποτε ετικέτα χρειάζονται για να περιγράψουν τα δεδομένα και τη δομή τους. Ο επεξεργαστής (XML parser) της γλώσσας XML είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML και το οποίο δουλεύει για μία εφαρμογή. Μερικές κωδικοποιήσεις βασισμένες στην XML τις συναντάμε σε αρκετές εφαρμογές γραφείου, όπως του Microsoft Office, του OpenOffice και του iWork της εταιρείας Apple.

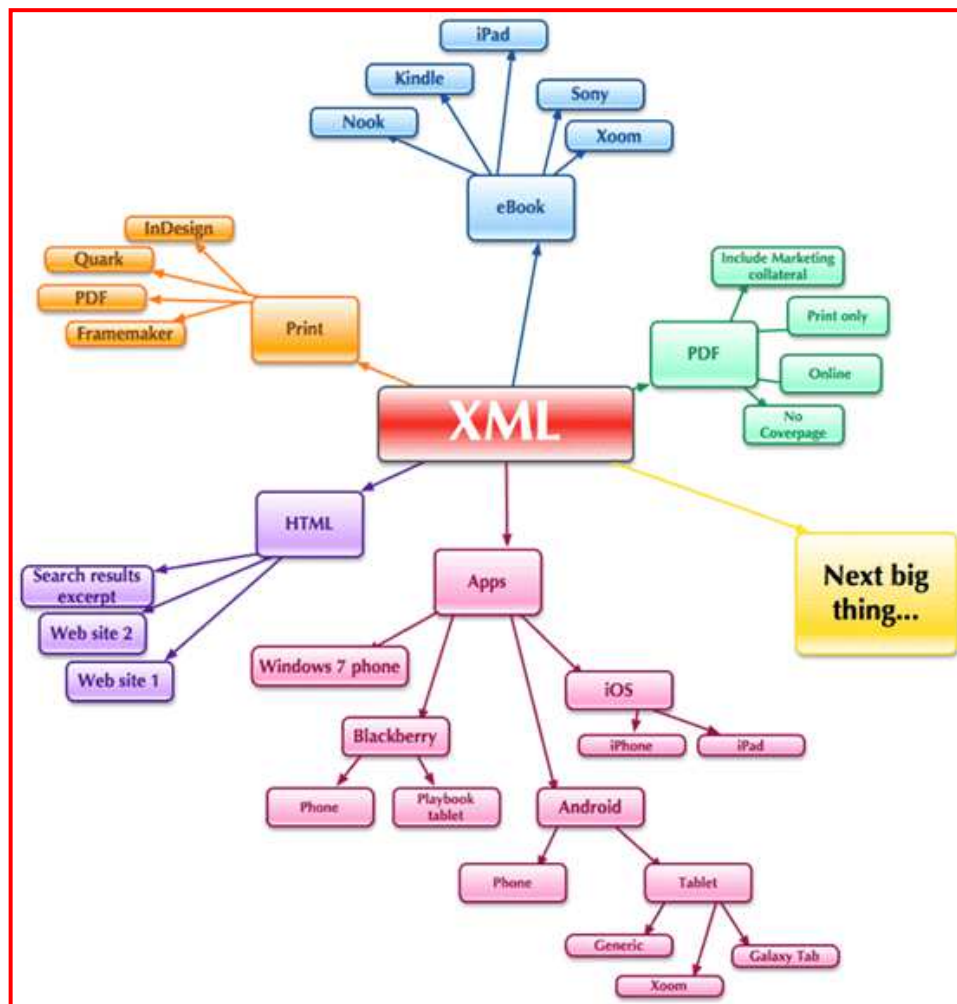
Έως τα τέλη της δεκαετίας του 2000 είχαν αναπτυχθεί πάρα πολλές γλώσσες που είχαν σαν βάση την XML. Κάποιες από αυτές ήταν και η RSS, η SOAP και η XHTML.

Για να συνοψίσουμε, οι στόχοι της XML είναι οι εξής:

1. Η XML πρέπει να είναι εύχρηστη στο Internet.
2. Η XML πρέπει να υποστηρίζει μεγάλη ποικιλία από εφαρμογές.
3. Η XML πρέπει να είναι συμβατή με την SGML.
4. Θα είναι εύκολο να γράφονται προγράμματα που επεξεργάζονται

XML έγγραφα.

5. Ο αριθμός των προαιρετικών χαρακτηριστικών στην XML θα είναι όσο το δυνατόν πιο μικρός, ιδανικό επίπεδο το μηδέν.
6. Τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα.
7. Ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα.
8. Ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός.
9. Τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα.
10. Η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας.



2. 4) Γλώσσα προγραμματισμού HTML

Οι λέξεις HyperText Markup Language (γλώσσα σήμανσης κειμένου) δημιουργούν εδώ, το ακρωνύμιο HTML, το οποίο είναι το όνομα της βασικής γλώσσας προγραμματισμού.

Δημιουργήθηκε στη δεκαετία του 1980 από τον φυσικό Τιμ Μπερνερς Λι. Ο Λι, ο οποίος εργαζόταν στο CERN επινόησε και κατασκεύασε το σύστημα χρήσης και διαμοιρασμού εγγράφων το λεγόμενο ENQUIRE. Στη συνέχεια δημιούργησε την προδιαγραφή της HTML και σύνθεσε τον browser και το λογισμικό του εξυπηρετητή. Έτσι, στα τέλη του 1991 δημοσιεύτηκε η πρώτη μορφή της γλώσσας και ήταν ένα έγγραφο το οποίο ονομαζόταν Ετικέτες HTML και περιέγραφε τα είκοσι στοιχεία που αποτελούσαν το σχεδιαστικό μέρος της. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4. Το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τύπογράφων, διαχωρίζει τη δομή από το περιεχόμενο και πάνω σε αυτήν στηρίζονται η HTML και η CSS. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, που και σε αυτή τη γλώσσα, περικλείονται μέσα στα σύμβολα μεγαλύτερο από και μικρότερο από (< , >), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης. Ανάμεσα σε αυτές, οι σχεδιαστές ιστοσελίδων και οι προγραμματιστές μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες. Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κάποιος να διαβάσει ή να ακούσει. Ο εκάστοτε browser «μεταφράζει» τις ετικέτες HTML, ερμηνεύσει το περιεχόμενο της ιστοσελίδας.

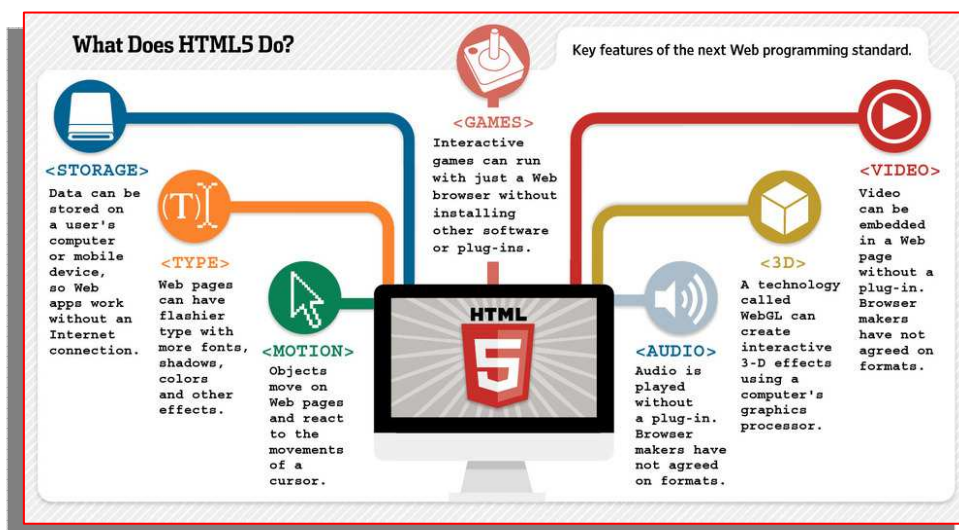
Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου, καθορίζοντας

δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα.

2.4.1) Διαφορές XML - HTML

Ανάμεσα στις δύο προαναφερθήςες γλώσσες προγραμματισμού υπάρχουν κάποιες ομοιότητες κυριώς όσον αφορά τη δομή του εγγράφου όμως σε καμία περίπτωση δεν είναι ίδιες. Παρακάτω αναφέρω μερικές σημαντικές διαφορές τους.

- Δεν μπορεί να γίνει αντικατάσταση XML με HTML, όμως υπάρχει η δυνατότητα να συμπεριληφθούν δεδομένα XML σε ετικέτες HTML και να προβληθούν σε μία ιστοσελίδα.
- Η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων, ενώ η XML αναπαριστά τη συναφή έννοια των δεδομένων.
- Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.
- Η HTML περιορίζεται σε ένα ορισμένο πακέτο ετικετών το οποίο μοιράζονται όλοι οι χρήστες.
- Η XML σχεδιάστηκε με τρόπο τέτοιο ώστε να ικανοποιήσει πολλές από τις ανάγκες των προγραμματιστών δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα.



2.4.2) HTML 5

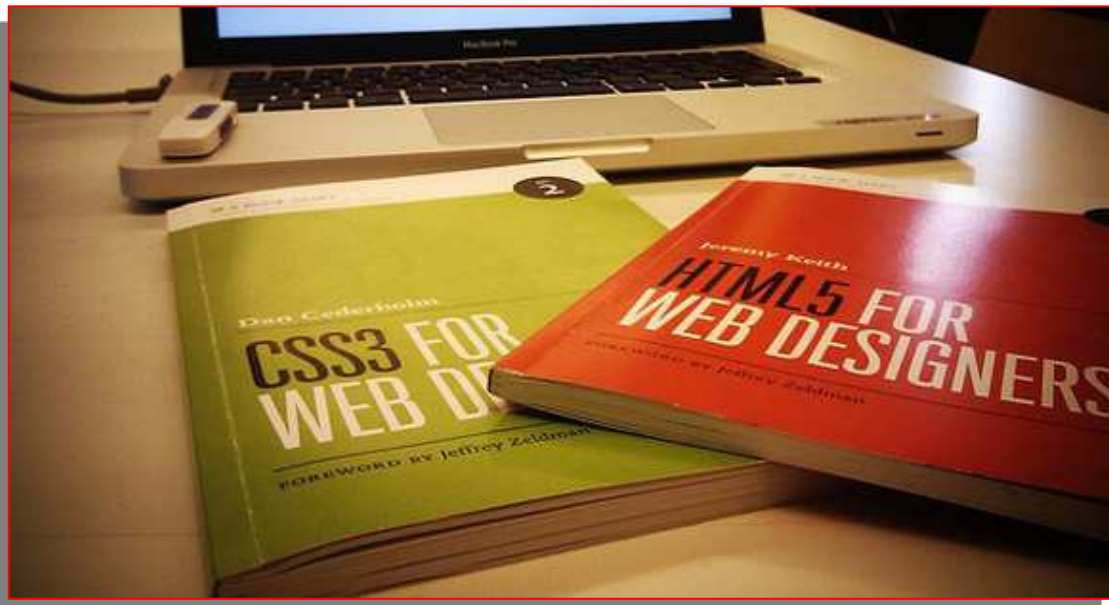
Στη συγκεκριμένη πτυχιακή εργασία χρησιμοποιήθηκε η πιο πρόσφατη έκδοση της γλώσσας προγραμματισμού HTML, η HTML 5. Το σημαντικό με αυτήν την έκδοση είναι ότι μας επιτρέπει να δημιουργήσουμε “mobile web sites” τα οποία να έχουν ταυτόχρονα πολλές από τις ιδιότητες των εφαρμογών που συναντάμε στις διάφορες πλατφόρμες κινητών, δίνοντας έτσι την ευκαιρία στην παρουσία του διαδικτύου και στα κινητά τηλέφωνα.

Η ανάγκη για standards και πλήρη συμβατότητα οδήγησε το World Wide Web Consortium (W3C) και το Web Hypertext Application Technology Working Group (WHATWG) να συνεργαστούν και να δημιουργήσουν την νέα αυτή γλώσσα. Οι εφαρμογές στα κινητά, παρέχουν μια πολύ πλουσιότερη εμπειρία από ένα τυπικό mobile web site. Ο λόγος για τον οποίο μια εφαρμογή το επιτελεί αυτό είναι γιατί εγκαθίσταται πάνω στο λειτουργικό του κινητού και αξιοποιεί στο έπακρο τις τεχνικές δυνατότητές του.

Ένα από τα χαρακτηριστικά του HTML 5 είναι ότι μπορεί να προσφέρει GPS, κάτι που θα φανεί ιδιαίτερα χρήσιμο ως εργαλείο για το ηλεκτρονικό εμπόριο και τις διαφημίσεις στο διαδίκτυο. Επίσης ένα άλλο μεγάλο χαρακτηριστικό γνώρισμα του HTML 5 είναι η δυνατότητά του να επιτρέψει την αναπαραγωγή βίντεο σε mobile web sites χωρίς τη χρήση του Flash.

Κάποιοι βασικοί κανόνες που έχουν οριστεί για την HTML5 είναι:

- Βάση για τα νέα χαρακτηριστικά να είναι οι HTML, CSS, DOM, και η JavaScript
- Ελαχιστοποίηση των plugins (όπως το Flash)
- Καλύτερη λειτουργία εντοπισμού λαθών
- Περισσότερο markup για να αντικατασταθεί το scripting
- Πλήρη συμβατότητα ανεξαρτήτως συσκευής



2.5) Γλώσσα προγραμματισμού CSS

Η γλώσσα προγραμματισμού CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Στυλ) ανήκει στην κατηγορία των γλωσσών φύλλων στυλ και χρησιμεύει στο να ελέγχει την εμφάνιση ενός εγγράφου το οποίο έχει γραφτεί με μία γλώσσα σήμανσης, HTML ή XHTML. Με την επισύναψη αρχείων δηλαδή, CSS σε δομημένα έγγραφα (HTML, XML) ο χρήστης έχει τη δυνατότητα να τροποποιήσει την παρουσίαση του εγγράφου χωρίς να κάνει αλλαγές στη δομή του, δηλαδή να προσθέσει κάποια ετικέτα.

Επίσης, η γλώσσα αυτή δίνει τη δυνατότητα να διαχωριστεί το περιεχόμενο από την εμφάνιση του εγγράφου κάνοντας έτσι πιο εύκολη τη δουλειά του προγραμματιστή σε θέματα που αφορούν την πρόσβαση στο περιεχόμενο, την καλύτερη δόμηση του κώδικα και θέματα βελτιστοποίησης. Χρησιμοποιώντας τη γλώσσα CSS στον προγραμματισμό ιστοσελίδων μπορούμε να εφαρμόσουμε την ίδια μορφοποίηση σε πολλές σελίδες, μπορούμε να μειώσουμε την πολυπλοκότητα του HTML κώδικα ορίζοντας τη μορφοποίηση μέσα στο CSS έγγραφο.

Ουσιαστικά, η συγκεκριμένη γλώσσα ελέγχει την εμφάνιση μίας ιστοσελίδας και ενός ιστότοπου από άποψη στυλ, δηλαδή χρώματα, στοίχιση και γενικά τα χαρακτηριστικά της και χρησιμοποιείται σε πολύ μεγάλο βαθμό στον σχεδιασμό διαδικτυακών εφαρμογών.



2.5.1) CSS3

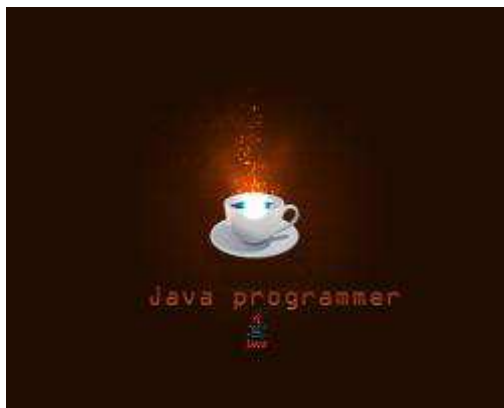
Στην συγκεκριμένη εργασία χρησιμοποιήθηκε η επόμενη έκδοση της γλώσσας CSS, η CSS3. Με τη CSS3 μπορούμε να δημιουργήσουμε οπτικά εφέ με διάφορα χαρακτηριστικά, όπως π.χ. εικόνες με στρογγυλεμένες άκρες, gradients, σκιές σε κείμενα, animations, διαφάνειες, πολλαπλές εικόνες φόντου, γραμματοσειρές και γενικά να διαμορφώσουμε ένα πολύ όμορφο και ενδιαφέρον design με πολύ ωραία γραφικά που μέχρι τώρα δημιουργόνταν μόνο στο Photoshop (= πρόγραμμα επεξεργασίας φωτογραφιών και εικόνων). Δίνονται οι δυνατότητες επίσης, στους προγραμματιστές να εξοικονομήσουν χρόνο για το επιθυμητό αποτέλεσμα με το να επιτρέπει λειτουργίες στυλ άλλων γλώσσων χωρίς αυτές να χρησιμοποιούνται.

Η CSS3 υποστηρίζεται από τους περισσότερους browsers (Safari, Firefox, Chrome κλπ.), στις νεότερες τις εκδόσεις .

Την επέλεξα διότι μου προσέφερε ένα ευρύ φάσμα δυνατοτήτων για να δημιουργήσω την κατάλληλη η εφαρμογή.

2.6) Γλώσσα προγραμματισμού Java

Μία γλώσσα που θα μπορούσε να φανεί χρήσιμη στο στήσιμο της συγκεκριμένης πρυχιακής είναι η Java. Η Java, αρχικά αναπτύχθηκε από την εταιρεία Sun και συγκεκριμένα από τον James Gosling, οποίος πειραματιζόταν πάνω στην γλώσσα C++ . Ο Gosling, πριν να οριστικοποιήσει την γλώσσα Java, είχε δημιουργήσει την πρόγονό της, μία γλώσσα με το όνομα Oak. Η Oak έμοιαζε αρκετά με την γλώσσα C++ αλλά ήταν πολύ πιο απλή. Η Oak μετονομάστηκε σε Java (ποικιλία καφέ,



αγαπημένη στους δημιουργούς της) διότι το πρώτο όνομα ήταν κατοχυρωμένο ήδη, και παρουσιάστηκε επίσημα στα μέσα του 1995.

Η γλώσσα προγραμματισμού Java, διαθέτει ένα μεγάλο πλεονέκτημα σε σχέση με τις άλλες γλώσσες, το λειτουργικό της σύστημα και η πλατφόρμας είναι ανεξάρτητα. Δηλαδή, τα προγράμματα που είναι γραμμένα σε Java «τρέχουνε» το ίδιο σε Linux, Unix, Macintosh και Windows, χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

2.6.1) Χαρακτηριστικά της Java

I. Απλή

Όπως προαναφέρθηκε, η βάση της γλώσσας Java είναι η C++, αλλά με λιγότερα στοιχεία, όπως η αφαίρεση της έννοιας των δεικτών και τα προβλήματα τα οποία έπρεπε να επιλυθούν εξαιτίας τους. Άρα, η Java είναι σίγουρα πιο απλή και ευκολονόητη διότι, οι προγραμματιστές πρώτα μαθαίνουν την C++ και έτσι στη συνέχεια τους είναι πολύ εύκολο να μάθουν την Java. Επίσης, περιλαμβάνει μόνο τρεις βασικούς τύπους δεδομένων, τους αριθμούς, τις λογικές μεταβλητές (Boolean) και τους πίνακες. Ένα από τα βασικά χαρακτηριστικά που έχει προσθέσει η Java είναι η αυτόματη διαχείριση της μνήμης, γνωστό και ως "μονάδα συλλογής σκουπιδιών" (garbage collector). Ο προγραμματιστής δεν χρειάζεται να απελευθερώνει την μνήμη που κατανέμει γιατί η εικονική μηχανή της Java (Virtual Machine ή VMJ) κάνει την δουλειά αυτή για λογαριασμό του προγραμματιστή. Τέλος, υποστηρίζει τη δημιουργία πολυνηματικών προγραμμάτων.

II. Αντικειμενοστραφής

Στον κόσμο της πληροφορικής όταν λέμε αντικειμενοστρεφή προγραμματισμό (object-oriented programming ή ΑΠ) , ονομάζουμε ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του 1960 και καθιερώθηκε κατά τη δεκαετία του 1990, αντικαθιστώντας σε μεγάλο βαθμό το παραδοσιακό υπόδειγμα του δομημένου προγραμματισμού. Έχει να κάνει με μία μεθοδολογία ανάπτυξης προγραμμάτων, υποστηριζόμενη από κατάλληλες γλώσσες προγραμματισμού, όπως εδώ την Java, όπου ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μίας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί πραγματικό στιγμιότυπο στη μνήμη ενός σύνθετου, και πιθανώς οριζόμενου από τον χρήστη, τύπου δεδομένων που ονομάζεται κλάση. Οι κλάσεις στην Java αποτελούνται από τις μεθόδους και τις μεταβλητές. Μέθοδοι λέγονται οι συναρτήσεις στις οποίες

μπορεί να απευθύνεται το αντικείμενο μίας κλάσης. Μεταβλητές λέγονται τα δεδομένα που καθορίζουν την κατάσταση ενός αντικειμένου.

III. Υψηλή απόδοση

Η γλώσσα Java μπορεί να μην έχει την αποδοτικότητα της C++ αλλά όμως οι περισσότερες εφαρμογές που περιέχουν γραφικά αποδίδονται εξαιρετικά από την Java.

IV. Εύρωστη

Οι προγραμματιστές της Java, δημιούργησαν μία γλώσσα η οποία θα ήταν απλή αλλά δυνατή. Όπως είπαμε και προηγουμένως, δεν χρησιμοποιούνται οι δείκτες και έτσι είναι αδύνατον ένα πρόγραμμα της Java να καταστρέψει δεδομένα ή να γράφει πάνω σε άλλα. Επιπροσθέτως, η η Java ακολουθεί τις γλώσσες Lisp και SmallTalk οι οποίες επιτρέπουν την αυτόματη απελευθέρωση της μνήμης η οποία έχει καταμεριστεί αλλά δεν χρησιμοποιείται πια.

V. Πολυνηματική

Ένα πρόγραμμα λέγεται πολυνηματικό, όταν σχεδιάζεται έτσι ώστε να μπορεί να εκτελεί περισσότερες από μία δουλειές ταυτόχρονα. Ένα νήμα έχει τη δυνατότητα να εκτελείται ανεξάρτητα σε μία εφαρμογή ή πολλά νήματα να εκτελούνται ταυτόχρονα. Τα νήματα είναι μία ροή εκτέλεσης μέσα σε μία εφαρμογή.

VI. Ουδέτερη – Αρχιτεκτονική

Ο μεταγλωττιστής της γλώσσας Java εφαρμόζει ένα κώδικα εντολών γραμμένο σε byte ο οποίος μεταγλωττίζεται από τον μεταγλωττιστή της Java και έτσι δίνεται στην Java μία ουδέτερη αρχιτεκτονική. Δηλαδή τα προγράμματα της Java μπορούν να διαβαστούν και να εκτελεστούν από οποιαδήποτε ηλεκτρονικό υπολογιστή ανεξάρτητα από το λειτουργικό σύστημα που διαθέτει.

VII. Ασφάλεια

Η Java εκτός από απλή και δυνατή είναι και μία γλώσσα πολύ ασφαλής. Δηλαδή, ο μεταγλωτιστής (compiler) της εξασφαλίζει ότι ο πηγαίος κώδικας δεν παραβιάζει τους κανόνες ασφαλείας. Η μονάδα φόρτωσης κλάσεων διασφαλίζει ότι οι κλάσεις δεν παραβιάζουν τους ισχύοντες περιορισμούς όταν φορτώνονται στο σύστημα. Ακόμα, η ασφάλεια προς το API εμποδίζει τις μικροεφαρμογές να δημιουργούν προβλήματα στο σύστημα. Επίσης, ο κώδικας που εκτελείται από το περιβάλλον χρόνου εκτέλεσης, σε μορφή byte code, ελέγχεται για να διασφαλιστεί ότι και αυτός επίσης υπακούει στους κανόνες ασφαλείας.

VIII. Εύκολη στη μεταφορά

Στην Java όλοι οι τύποι δεδομένων έχουν προκαθορισμένο μέγεθος ανεξάρτητα από το μηχάνημα ή το λειτουργικό σύστημα στο οποίο το πρόγραμμα τρέχει.

2.7) Γλώσσα προγραμματισμού JavaScript

Στην παρούσα πτυχιακή επίσης, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Javascript. Η JavaScript αρχικά δημιουργήθηκε από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha, αργότερα LiveScript και τελικά JavaScript επειδή ήταν επηρεασμένη από την γλώσσα προγραμματισμού Java.

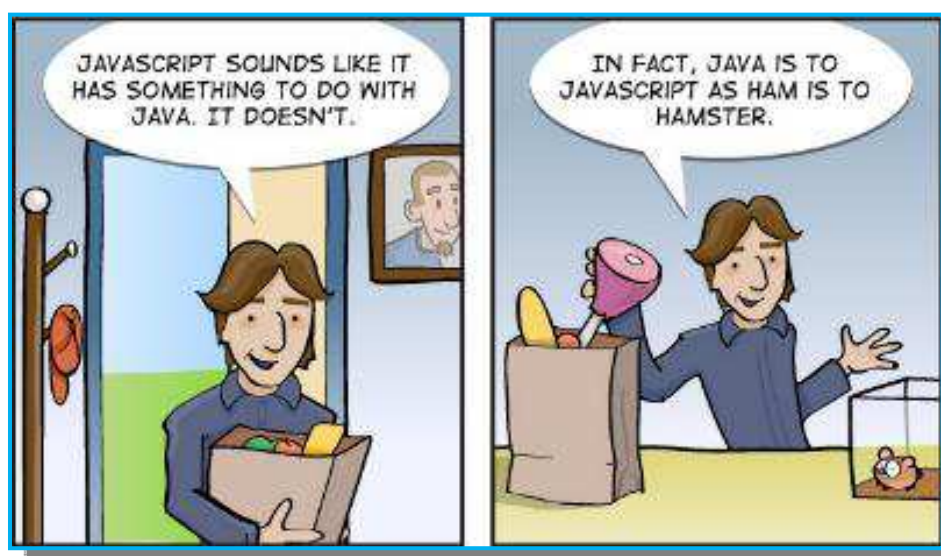
Στα τέλη του 1996 η εταιρεία Netscape υπέβαλε τη γλώσσα Javascript στο Ecma International η οποία είναι μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού προς εξέταση ως βιομηχανικό πρότυπο, και τελικά το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Αρχικά η Javascript ήταν βασισμένη στη γλώσσα προγραμματισμού C και χρησιμοποιούνταν για προγραμματισμό από την πλευρά του πελάτη (client) ο οποίος ήταν ο φυλλομετρητής (browser) του χρήστη (client – side). Δηλαδή, η επεξεργασία του κώδικα javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών.

Η συγκεκριμένη γλώσσα προγραμματισμού δεν χρειάζεται ιδιαίτερες δυνατότητες από τον sever για να εκτελεστεί αλλά βασίζεται στον browser των επισκεπτών. Επιπροσθέτω, μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Η JavaScript χρησιμοποιείται σε εφαρμογές έξω από ιστοσελίδες, όπως για παράδειγμα σε έγγραφα PDF, site-specific προγράμματα περιήγησης και desktop widgets, μοιάζει με την C++ αλλά σαν στόχο έχει την αξιοπιστία στις εφαρμογές του διαδικτύου. Πολλά ονόματα και συμβάσεις ονοματολογίας έχουν αντιγραφεί από τη Java, αλλά οι δύο γλώσσες είναι διαφορετικές και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού στο πλαίσιο της JavaScript, λαμβάνονται από Self και Scheme γλώσσες προγραμματισμού. Ο κώδικας Javascript μίας σελίδας περικλείεται από τις ετικέτες `<scripttype="text/javascript">` και `</script>`.

Συνοπτικά, η γλώσσα Javascript είναι μία γλώσσα scripting και σχεδιάστηκε για να προσθέσει αλληλεπίδραση στις ιστοσελίδες. Μπορούμε να φτιάξουμε σενάρια που να εκτελούν αυτόματες εργασίες (όταν μία σελίδα Web ή ένα παράθυρο ανοίγει ή κλείνει).

Στην επόμενη σελίδα βλέπουμε ένα συνοπτικό πίνακα με τις διαφορές ανάμεσα σε Java και Javascript.



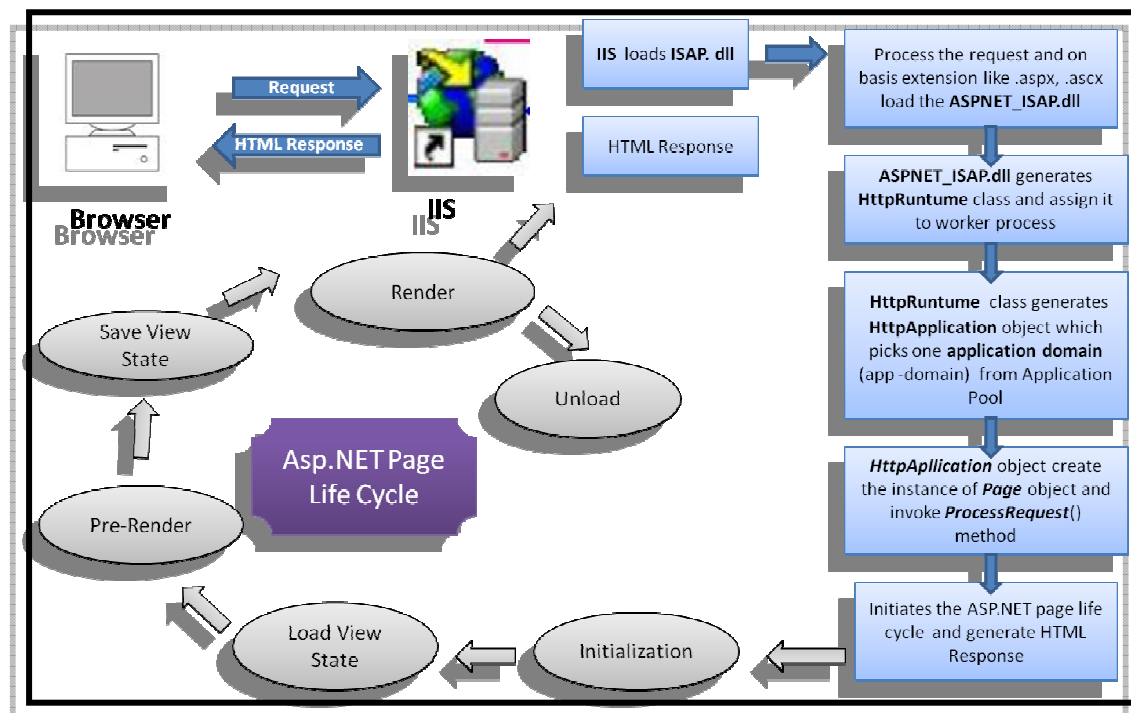
Διαφορές Java - Javascript

Η Java είναι περίπλοκη στην χρήση	Η Javascript είναι σχετικά εύκολη στην χρήση
Οι εφαρμογές της Java είναι λίγο αργές	Η Javascript εκτελείται πολύ γρήγορα.
Η Java χρειάζεται το JDK (JAVA DEVELOPER'S KIT)	Η Javascript χρειάζεται μόνο ένας φυλλομετρητή συμβατός με Javascript
Τα προγράμματα στην Java μεταγλωττίζονται σε εκτελέσιμα αρχεία και ενσωματώνονται στις σελίδες της html με ετικέτα <applet>. Οι φυλλομετρητές φορτώνουν και τρέχουν το αρχείο.	Στην Javascript τα προγράμματα ενσωματώνονται στις σελίδες της html με την μορφή σεναρίου και δεν χρειάζονται μεταγλώτιση ή διερμήνευση
Η Java είναι πλήρης γλώσσα προγραμματισμού με πολλές δυνατότητες	Η javascript είναι πιο κατάλληλη για απλές χρήσεις
Η Java είναι προσανατολισμένη στο αντικείμενο και είναι καθαρά αντικειμενοστρεφής γλώσσα προγραμματισμού	Η Javascript είναι βασισμένη στο αντικείμενο αλλά δεν είναι αληθινή αντικειμενοστρεφής γλώσσας προγραμματισμού.
Η Java έχει αυστηρούς κανόνες στη χρήση των μεταβλητών	Η Javascript χαρακτηρίζεται από χαλαρότητα ως προς τις μεταβλητές.

2.8) Γλώσσα προγραμματισμού asp.net

Το ASP.NET είναι ένα εργαλείο web εφαρμογής το οποίο επινοήθηκε και δημιουργήθηκε από την Microsoft για να διευκολύνει τους προγραμματιστές με το διαχωρισμό της παρουσίασης και του περιεχομένου και να δύναται να γράψει κώδικα. Κυκλοφορεί στο εμπόριο από τη Microsoft και επιτρέπει στους προγραμματιστές την δημιουργία δυναμικών ιστοσελίδων, εφαρμογών και υπηρεσιών web. Κυκλοφόρησε για πρώτη φορά τον Ιανουάριο του 2002 με την έκδοση 1.0 του .NET Framework, και είναι ο διάδοχος της τεχνολογίας Active Server Pages (ASP) της Microsoft. Το ASP.NET είναι «χτισμένο» με την γλώσσα Common Language Runtime (CLR), επιτρέποντας στους προγραμματιστές να γράψουν κώδικα ASP.NET χρησιμοποιώντας οποιαδήποτε γλώσσα υποστηρίζεται από το .NET.

Οι ASP.NET σελίδες, που είναι γνωστά ως «Web Forms», είναι τα κύρια δομικά στοιχεία για την ανάπτυξη εφαρμογών. Τα Web Forms περιέχονται σε αρχεία με επέκταση .aspx και περιέχουν την στατική (X)HTML σήμανση, όπως επίσης και σήμανση για τον καθορισμό στην πλευρά του διακομιστή, τα Web Controls και τα User Controls, όπου οι προγραμματιστές τοποθετούν όλα τα απαιτούμενα στατικά και δυναμικά περιεχόμενα για την ιστοσελίδα τους. Επίσης, ο δυναμικός κώδικας που τρέχει στον server μπορεί να τοποθετηθεί σε μια σελίδα μέσα σε ένα μπλοκ (<% - δυναμικός κώδικας -%>), και είναι παρόμοιος με άλλες τεχνολογίες ανάπτυξης ιστοσελίδων όπως PHP, JSP και ASP. Με το ASP.NET Framework 2.0, η Microsoft εισήγαγε ένα νέο «code-behind» μοντέλο το οποίο επιτρέπει στο στατικό κώδικα να παραμείνει στην .aspx σελίδα, ενώ ο δυναμικός κώδικας να παραμένει σε ένα .aspx.vb ή .aspx.cs ή .aspx.fs αρχείο (ανάλογα με τη γλώσσα προγραμματισμού που χρησιμοποιείται). Από τεχνικής απόψης το ASP διαθέτει αντικείμενα τα οποία καλούνται από την VBscript ή την Javascript για την εκτέλεση συγκεκριμένων, εξαιρετικά χρήσιμων λειτουργιών, όπως η συλλογή των δεδομένων που στέλνουν οι χρήστες, η ανταπόκριση στην είσοδο των χρηστών, η διαχείριση εφαρμογών και συνοδών εργασίας και ο χειρισμός του Server.



2.9) Γλώσσα προγραμματισμού PHP

Μία άλλη τεχνολογία η οποία θα μπορούσε να φανεί χρήσιμη στο στήσιμο της ιστοσελίδας που πραγματεύομαι είναι και η PHP.

Στα μέσα της δεκαετίας του '90 ο Rasmus Lerdorf δημιουργεί για προσωπική χρήση ένα script με το όνομα PHP.cgi χρησιμοποιώντας της γλώσσα προγραμματισμού Perl. Στη συνέχεια, εφόσον ήταν λειτουργικό και χρήσιμο πέρασε σε ευρεία χρήση και τελικά περίπου το 1997 η γλώσσα PHP/FI (Personal Home Page/Form Interpreter), βασιζόμενη πλέον στη γλώσσα C φτάνει στη έκδοση 2.0. Ακολούθησαν αρκετές εκδόσεις και μέχρι σήμερα έχουμε τις εκδόσεις 4 και 5 της οποίες και χρησιμοποιούν πιο πολύ οι προγραμματιστές.

Η PHP είναι μία γλώσσα προγραμματισμού η οποία χρησιμοποιείται για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο και είναι επισήμως γνωστή ως HyperText preprocessor. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη

επέκταση (π.χ. .php, .php4, .html κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα του πελάτη.



Μερικά από τα πλεονεκτήματα της γλώσσας PHP θα μπορούσαμε να πούμε ότι είναι τα εξής:

- μπορεί να χρησιμοποιηθεί για όλα τα σημαντικά λειτουργικά συστήματα (Linux, Unix, Solaris, OpenBSD, Microsoft Windows, Mac OS X, RISC OS),

- υποστηρίζει τους περισσότερους διακομιστές web (Apache, IIS) και αυτό περιλαμβάνει οποιοδήποτε web server ο οποίος μπορεί να χρησιμοποιήσει το FastCGI PHP binary, όπως lighttpd και Nginx,
- δουλεύει είτε ως module, είτε ως επεξεργαστής CGI,
- υπάρχει ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server,
- υπάρχει η δυνατότητα χρήσης συναρτησιακού (procedural programming) ή αντικειμενοστραφή προγραμματισμού (OOP), ή ένα μείγμα από τους δύο,
- δεν υπάρχει περιορισμός εξαγωγής HTML,
- οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash οι οποίες παράγονται αμέσως,
- γίνεται εύκολα οποιοδήποτε εξαγωγή κειμένου όπως XHTML και οποιοδήποτε άλλο αρχείο XML,
- δημιουργεί αυτόματα αρχεία και τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό μας περιεχόμενο,
- ένα από τα ισχυρότερα και πιο σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη από ένα ευρύ φάσμα βάσεων δεδομένων.

Η δημιουργία μίας ιστοσελίδας με υποστήριξη βάσης δεδομένων είναι πολύ απλή και μπορεί να γίνει με τη χρήση συγκεκριμένων επεκτάσεων (π.χ. για την MySQL), είτε χρησιμοποιώντας ένα abstract layer, όπως PDO, ή με τη σύνδεση σε οποιαδήποτε βάση δεδομένων που υποστηρίζει το πρότυπο Open Database Connection μέσω της επέκτασης ODBC. Άλλες βάσεις δεδομένων μπορούν να χρησιμοποιήσουν URL ή sockets, όπως η CouchDB.

3ο Κεφάλαιο

3.1) Τί είναι οι βιβλιοθήκες;

Η έννοια της βιβλιοθήκης στον κόσμο της πληροφορικής είναι αναπόσπαστο κομμάτι του δομημένου προγραμματισμού και αναπτύχθηκε παράλληλα με αυτόν. Είναι μία συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη ενός λογισμικού.

Οι βιβλιοθήκες περιλαμβάνουν υποβοηθητικό κώδικα και δεδομένα προσφέροντας με αυτόν τον τρόπο υπηρεσίες σε προγράμματα. Τα σύγχρονα λειτουργικά συστήματα παρέχουν βιβλιοθήκες που υλοποιούν την πλειονότητα των υπηρεσιών του συστήματος. Τα εκτελέσιμα αρχεία και οι βιβλιοθήκες αναφέρονται το ένα στον κώδικα και τα δεδομένα του άλλου μέσω μιας διαδικασίας που ονομάζεται σύνδεση και την πραγματοποιεί ο συνδέτης. Έτσι, ο περισσότερος κώδικας που χρησιμοποιούν οι σύγχρονες εφαρμογές παρέχεται από αυτές τις βιβλιοθήκες και δεν χρειάζεται να γραφεί από την αρχή για κάθε νέο πρόγραμμα.

3.2) Βιβλιοθήκη jQuery

Η βιβλιοθήκη που χρησιμοποίησα για την εκπόνηση της ιστοσελίδας είναι η jQuery. Η jQuery στις αρχές του 2006 από τον John Resig στο BarCamp. Η jQuery είναι μια βιβλιοθήκη με συναρτήσεις γραμμένες σε γλώσσα JavaScript με αδειοδότηση από την MIT LICENSE και την GNU GENERAL PUBLIC LICENSE. Χρησιμοποιείται από προγραμματιστές για να δημιουργήσουν ιστοσελίδες και διαδυκτιακές εφαρμογές που χρειάζονται ευχρηστία και διαδραστικότητα.

Ένα από τα πλεονεκτήματα της jQuery είναι ότι ακολουθεί την αρχή KISS (Keep It Simple Stupid) δηλαδή προσπαθεί να υπεραπλουστεύσει τον προγραμματισμό σε Javascript προσφέροντας απλούς μηχανισμούς και εντολές μέσω του framework της. Επίσης, παρέχει μία πλήρη τεκμηρίωση μέσω του Visual jQuery. Έχει ένα μικρό

μέγεθος της τάξης των 20kb το οποίο δίνει την δυνατότητα να γίνει εύκολα κατανοητή η αρχιτεκτονική της. Ο προγραμματιστής μπορεί να χρησιμοποιήσει όλα τα πλεονεκτήματα και τις δυνατότητες (εφέ, εμφάνιση-απόκρυψη κλπ.) που προσφέρει η Javascript. Η jQuery είναι σχεδιασμένη να επιλέγει στοιχεία HTML και να εφαρμόζει κάποιες ενέργειες στα επιλεγμένα στοιχεία.

Επέλεξα την συγκεκριμένη διότι διαθέτα τις απαραίτητες γνώσεις (HTML, CSS, JAVASCRIPT), και επίσης, αυτό που χαρακτηρίζει την γλώσσα jQuery είναι γενικά, ο συμπαγής κώδικα και η ευκολία στην συγγραφή.

jQuery API/1.2 http://jquery.com		EVENTS		CORE UI EFFECTS					
SELECTORS #id, tag, .class, * E[@attr] elm1, elm2, elmN E[@attr=val] ancestor descendant E[@attr^=val] (begins) parent > child E[@attr\$=val] (ends) parent/child E[@attr*=val] (contains) prev + next E[@attr=val][@attr=val] (both) prev ~ siblings :nth-child(index) :first :first-child :last :last-child :not(selector) :only-child :input :even :text :odd :enabled :password :eq(index) :disabled :radio :gt(index) :checked :checkbox :lt(index) :selected :submit :image :contains(text) :hidden :reset :empty :visible :button :has(selector) :header :file :parent :animated :hidden		HANDLERS .bind(type, data, fn) .one(type, data, fn) .trigger(type, data) .triggerHandler(type, data) .unbind(type, data) MOUSE .mousedown(fn) .mousemove(fn) .mouseout(fn) .mouseover(fn) .mouseup(fn) WINDOW .load(fn) .scroll(fn) .resize(fn) ERROR .error() .error(fn) KEYBOARD .keydown() .keydown(fn) .keypress() .keypress(fn) .keyup() .keyup(fn) PAGE .ready(fn)		INTERACTION .hover(fnIN, fnOUT) .toggle(fnIN, fnOUT) .blur() .blur(fn) .change() .change(fn) .click() .click(fn) .dblclick() .dblclick(fn) .focus() .focus(fn) .select() .select(fn) .submit() .submit(fn) .unload() .unload(fn) .unblur() .unblur(fn)		SHOW / HIDE .show() .show(speed, callback) .hide() .hide(speed, callback) .toggle() SLIDE (speed, callback) .slideDown(s, c) .slideUp(s, c) .slideToggle(s, c) ANIMATE .stop() .queue() .queue(callback) .queue(queue) .dequeue() .animate(params, duration, easing, callback) .animate(params, options) FADE .fadeIn(speed, callback) .fadeOut(speed, callback) .fadeTo(speed, opacity, callback)			
CSS .css(name, value) .css(properties) .height(value) .width(value) .addClass(class) .removeClass(class) .toggleClass(class) .offset() .attr(name) .attr(key, value) .attr(key, function) .removeAttr(name) ATTRIBUTES .attr(name) .attr(key, value) .attr(key, function) .removeAttr(name) HTML .html() .html(value) .text() .text(value) .val(value)		TRaversing FILTER .hasClass(class) .filter(expr) .filter(fn) is(expr) .map(callback) .not(expr) .slice(start, end) ACCESS .each(callback) .size() .length .get() .get(index) .index(subject) FIND (expr) .add(e) .children(e) .siblings(e) .contents() .find(e) .next(e) .nextAll(expr) .parent(e) .parents(e) .prev(e) .prevAll(e) CHAIN .andSelf() .end()		MANIPULATING INSIDE (content) .append(c) .appendTo(c) .prepend(c) .prependTo(c) AROUND .wrap(html) .wrap(element) .wrapAll(html) .wrapAll(element) .wrapInner(html) .wrapInner(element) OUTSIDE (content) .after(c) .before(c) .insertAfter(c) .insertBefore(c) REPLACE .replaceWith(c) .replaceAll(selector) CLEAR .empty() .remove(expression) CLONE .clone() .clone(true)		AJAX Request (url, data, callback) \$.ajax(options) .load(u, d, c) \$.get(u, d, c) \$.getJSON(u, d, c) \$.getScript(u, c) \$.post(u, d, c) .loadIfModified(u, d, c) Event Handler (callback) .ajaxComplete(c) .ajaxError(c) .ajaxSend(c) .ajaxStart(c) .ajaxStop(c) .ajaxSuccess(c) Serialize .serialize() .serializeArray() .ajaxSetup(options)			
USER AGENT \$.browser, \$.browser.version \$.boxModel		JavaScript \$.extend(obj1, ..., objN) \$.grep(array, callback, invert) \$.map(array, callback) \$.unique(array) \$.trim(string) \$.merge(1st, 2nd)		COLORCHARGE http://colorcharge.com jQuery 1.2 Cheat-sheet updated: December 23rd, 2007		EXTEND \$.fn.extend(obj) \$.extend(obj) \$.noConflict()		\$(); \$(expression, context), \$(html) \$(elements), \$(callback)	

4ο Κεφάλαιο

4.1) Τί είναι η τεχνολογία AJAX;

Το ακρονύμιο Asynchronous Javascript and XML σχηματίζουν την τεχνολογία AJAX. Η AJAX είναι μία τεχνολογία η οποία είναι η πιο σύγχρονη σε ότι αφορά τον προγραμματισμό στο κόσμο του διαδικτύου. Εντυπωσιακές εμφανίσεις, ευφάνταστα scrolling και δυναμικά ξεφυλίσματα σε photo album είναι αποτελέσματα της συγκεκριμένης τεχνολογίας η οποία προσφέρει διαδραστικές δυνατότητες σε ένα δυναμικό site, μετατρέποντας μία απλή ιστοσελίδα σε μία διαδικτυακή εφαρμογή.

Η AJAX προσφέρει τη δυνατότητα εμφάνισης νέων στοιχείων στο site, χωρίς τη φόρτωση νέας σελίδα. Μπορεί δηλαδή ο web developer να δημιουργήσει ένα site με μία μόνο σελίδα, στην οποία θα φορτώνονται διαφορετικά δεδομένα ανάλογα με τις επιλογές του χρήστη. Με αυτόν τον τρόπο καταργεί τους ατελείωτους φακέλους με τα html αρχεία, στα οποία επαναλαμβάνεται το ίδιο κομμάτι κώδικα, βελτιώνοντας παράλληλα και την ασφάλεια του site καθώς καταργεί την αλλαγή του url στη μπάρα διευθύνσεων.

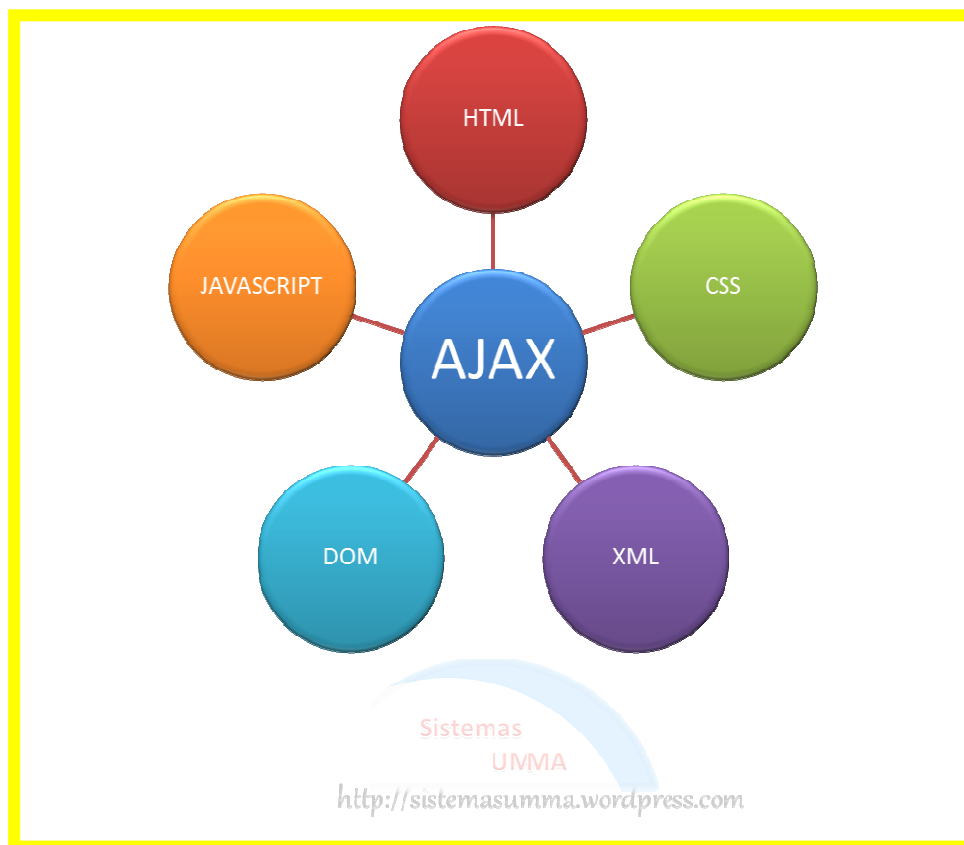
Η κύρια γλώσσα με την οποία εφαρμόζεται η AJAX είναι η JavaScript. Όμως τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους και κάποιες πρόσθετες βιβλιοθήκες οι οποίες δίνουν τη δυνατότητα στον προγραμματιστή να γράψει κώδικα σε JavaScript με ποιο εύκολο, σύντομο και κατανοητό τρόπο. Τέτοιες βιβλιοθήκες είναι οι:

- jQuery
- Dojo
- YUI
- MooTools
- Prototype

Η δημοφιλέστερη και η ποιο εύχρηστη από αυτές είναι η jQuery. Το μόνο που χρειάζεται για την εγκατάστασή της, είναι το κατέβασμα του αρχείου της βιβλιοθήκης και η αποθήκευσή του στο φάκελο του site που θα χρησιμοποιηθεί η

jQuery. Ένας άλλος εναλλακτικός τρόπος είναι η δήλωση του url του αρχείου της βιβλιοθήκης απευθείας μέσα στο κώδικα του site αποφεύγοντας έτσι το κατέβασμα της βιβλιοθήκης.

Είναι μία τεχνολογία που θα μπορούσε να χρησιμοποιηθεί στην δημιουργία της ιστοσελίδας που δημιούργησα στην εκπόνηση της συγκεκριμένης πτυχιακής εργασίας.



5ο Κεφάλαιο

5.1) Υλοποίηση

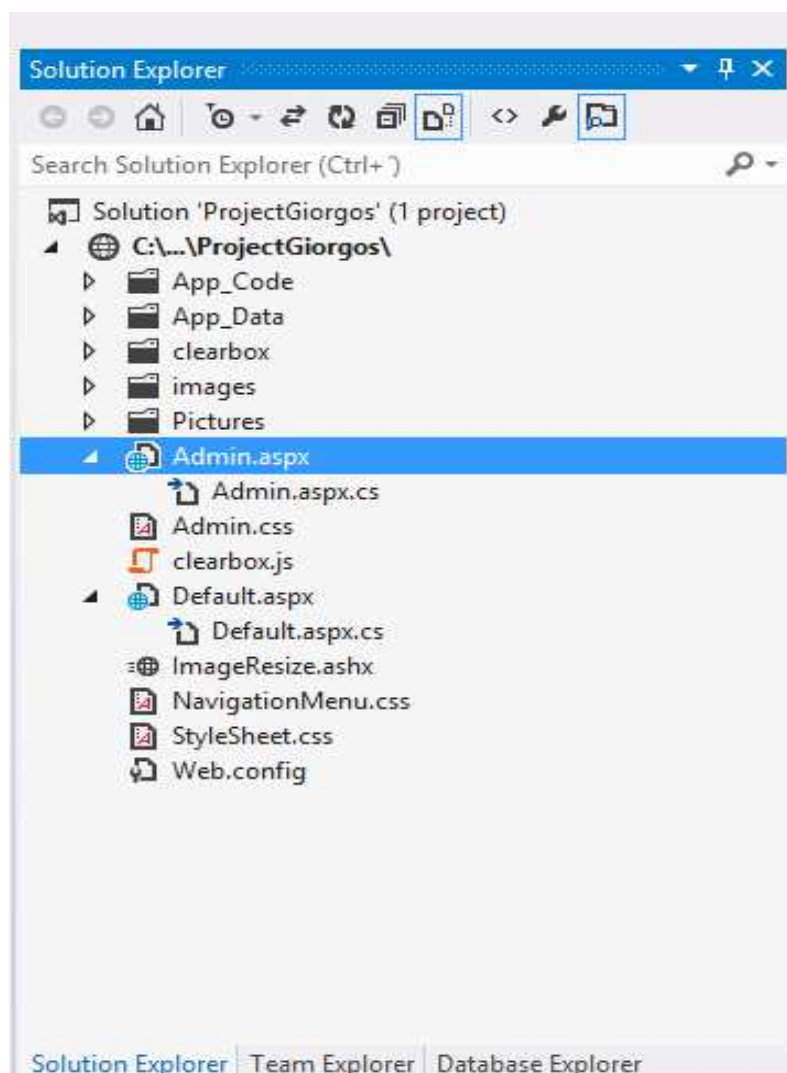
Για την υλοποίηση της πτυχιακής μου εργασίας χρησιμοποίησα το Microsoft Visual Studio Express 2012 for Web καθώς και τον λογισμικό επεξεργασίας εικόνων και δημιουργίας γραφικών Adobe Photoshop CS4.

Στην συνέχεια παραθέτω στιγμιότυπα του περιβάλλοντος όπου υλοποιήθηκε η πτυχιακή μου εργασία , τα απαραίτητα αρχεία που την απαρτίζουν καθώς και μια σύντομη περιγραφή – επεξήγηση κάποιων σημαντικών σημείων του κώδικα.

Στο φάκελο *App_Code* υπάρχει το αρχείο *Pictures.cs* το οποίο περιέχει την κλάση για το data layer - δηλαδή ο κώδικας για τη διαχείριση των δεδομένων - της εφαρμογής.

Ο φάκελος *App_data* περιέχει το αρχείο *Pictures.xml* το οποίο αποθηκεύει τα δεδομένα μας, αποτελεί στην ουσία τη βάση δεδομένων μας.

Ο φάκελος *clearbox* περιέχει όλα εκείνα τα αρχεία που είναι απαραίτητα για την απεικόνιση των thumbnails μας –πατώντας πάνω σε αυτά- σε φωτογραφίες, μέσω μιας pop-up λειτουργίας, στο πραγματικό τους μέγεθος. Δεν αποτελεί δική μου



Εικόνα 9: Φάκελοι και αρχεία του project

υλοποίηση. Διατίθεται δωρεάν στο διαδίκτυο.

Στο φάκελο *images* είναι αποθηκευμένες όλες οι εικόνες και τα γραφικά που χρησιμοποιήθηκαν για την υλοποίηση του οπτικού μέρους του ιστότοπου.

Στο φάκελο *Pictures* αποθηκεύονται οι εικόνες εκείνες που ο χρήστης ανεβάζει στην ιστοσελίδα και αποτελούν στην ουσία το περιεχόμενο του ψηφιακού μας άλμπουμ.

Το αρχείο *Admin.aspx* αποτελεί το γραφικό περιβάλλον της σελίδας όπου ο χρήστης καλείται να συνδεθεί με τη βάση πληκτρολογώντας το «όνομα χρήστη» και τον «κωδικό» καθώς επίσης και το οπτικό μέρος των σελιδών όπου γίνεται η διαχείριση των φωτογραφιών. Το αρχείο *Admin.aspx.cs* αποτελεί την προγραμματιστική διαδικασία για την επίτευξη της παραπάνω λειτουργίας, δηλαδή της εισόδου του χρήστη στο σύστημα και της δυνατότητας του να τις διαχειριστεί.

Το αρχείο *Default.aspx* αποτελεί και αυτό το γραφικό περιβάλλον του ψηφιακού μας άλμπουμ, της σελίδας δηλαδή που έχουν πρόσβαση οι επισκέπτες. Στο αρχείο *Default.aspx.cs* βρίσκεται ο κώδικας που είναι υπεύθυνος για την εμφάνιση των φωτογραφιών μας καθώς και για την αναζήτηση τους βάσει της ονομασίας ή της περιγραφής που τους έχει δοθεί, δυνατότητα που δίνεται στον επισκέπτη.

Το αρχείο *ImageResize.ashx* είναι μια έτοιμη βιβλιοθήκη που μετατρέπει τις φωτογραφίες σε μικρογραφίες.

Τα αρχεία *NavigationMenu.css* και *Stylesheet.css* αποτελούν την μορφοποίηση της σελίδας μας *Default.aspx*, είναι δηλαδή εκείνα που είναι υπεύθυνα για το οπτικό αποτέλεσμα της σελίδας καθώς και της δυνατότητας προσαρμογής της σελίδας σε οποιαδήποτε ανάλυση οθόνης.

Στο αρχείο *Web.config* είναι αποθηκευμένα ο κωδικός πρόσβασης του χρήστη καθώς και το *username*.

Στη συνέχεια θα δώσουμε μια σύντομη περιγραφή κομματιών κώδικα που εκτελούν λειτουργίες απαραίτητες για την υλοποίηση της εφαρμογής.

Αρχικά ορίζουμε μια μεταβλητή τύπου boolean, την isAdmin η οποία δείχνει αν έχει κάνει login κάποιος διαχειριστής. Σε περίπτωση που έχει κάνει login τότε δίνουμε

```
public partial class Admin : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        bool isAdmin;
        if (Session["isAdmin"] == null)
            isAdmin = false;
        else
            isAdmin = (bool)Session["isAdmin"];

        if (isAdmin)
        {
            pnlAdmin.Visible = true;
            pnlLogin.Visible = false;
        }
        else
        {
            pnlAdmin.Visible = false;
            pnlLogin.Visible = true;
        }

        if (!Page.IsPostBack)
        {
            BindGrid();
        }
    }
}
```

Εικόνα 10:Έλεγχος εισόδου του χρήστη

εντολή να εμφανιστεί το πάνελ της διαχείρισης και ταυτόχρονα να αποκρύψουμε τη φόρμα εισόδου. Με το IsPostBack ελέγχουμε αν έχουμε ξαναφορτώσει τη σελίδα έτσι ώστε σε περίπτωση που το έχουμε κάνει να μην χρειαστεί να φορτώσουμε τις φωτογραφίες από το xml ξανά.

Στη συνέχεια ορίζουμε μια μεταβλητή τύπου Pictures η οποία διαχειρίζεται τη βάση δεδομένων σε xml. Με την εντολή `DataTable Pictures = MyPictures.Select();` αποθηκεύουμε τις φωτογραφίες από το xml σε ένα πίνακα στη μνήμη. Στη συνέχεια για κάθε εγγραφή διορθώνουμε την εικόνα ώστε να περιέχει το url της εικόνας όπως το βλέπει ο χρήστης και φορτώνουμε τον πίνακα με τα δεδομένα στο gridview.

Το gridview είναι μια διεπαφή χρήστη που διαχειρίζεται τα περιεχόμενα ενός πίνακα και με μια μορφοποίηση που του έχουμε ορίσει εμείς τα εμφανίζει στην οθόνη.

```
private void BindGrid()
{
    Pictures MyPictures = new Pictures();
    DataTable Pictures = MyPictures.Select();

    foreach (DataRow Picture in Pictures.Rows)
    {
        Picture["FileName"] = ResolveUrl("~/Pictures/" + Picture["FileName"]);
    }

    grvPictures.DataSource = Pictures;
    grvPictures.DataBind();
}
```

Εικόνα 11: Στιγμιότυπο κώδικα

Στη συνέχεια διαχειριζόμαστε το event για το κουμπί Insert. Σε αυτό το event γίνεται έλεγχος για την ύπαρξη ή όχι τόσο του φακέλου όπου θα αποθηκεύουμε τις φωτογραφίες μας όσο για των ίδιων των φωτογραφιών. Συγκεκριμένα, σε περίπτωση που δεν έχει δημιουργηθεί φάκελος για την αποθήκευση των φωτογραφιών, δηλαδή είναι η πρώτη φορά που εισάγουμε κάποια εικόνα, τότε τον δημιουργούμε. Ελέγχουμε έπειτα αν ο χρήστης έχει επιλέξει μια φωτογραφία. Αν η επιλεγμένη φωτογραφία δεν υπάρχει στο φάκελο που δημιουργήσαμε τότε την αποθηκεύουμε, διαφορετικά εμφανίζουμε κατάλληλο μήνυμα προς το χρήστη. Σε περίπτωση που ο χρήστης δεν έχει επιλέξει καμία εικόνα εμφανίζεται κατάλληλο μήνυμα για αυτό το σκοπό.

Στη συνέχεια διαχειριζόμαστε το event ανανέωσης (update button) αποθηκεύοντας τις πληροφορίες της κάθε εικόνας όπως τίτλο και περιγραφή και ανανεώνοντας το gridView. Τέλος κρύβουμε τη φόρμα επιλογής φωτογραφιών εμφανίζουμε το

```
protected void btnInsert_Click(object sender, EventArgs e)
{
    if (!Directory.Exists(MapPath("~/Pictures")))
    {
        Directory.CreateDirectory(MapPath("~/Pictures"));
    }
    if (fulFileUploadControl.HasFile)
    {
        if (!File.Exists(MapPath("~/Pictures/" + fulFileUploadControl.FileName)))
        {
            fulFileUploadControl.SaveAs(MapPath("~/Pictures/" + fulFileUploadControl.FileName));
        }
        else
        {
            lblErrorMessage.Text = "<br />The file name already exists.";
            return;
        }
    }
    else
    {
        lblErrorMessage.Text = "<br />You have to select a picture.";
        return;
    }

    Pictures.Insert(fulFileUploadControl.FileName, txtTitle.Text, txtDescription.Text);
    BindGrid();

    tblForm.Visible = false;
    grvPictures.Visible = true;

    btnInsertRecord.Visible = true;
}
protected void btnUpdate_Click(object sender, EventArgs e)
{
    Pictures.Update(hdfPictureID.Value, txtTitle.Text, txtDescription.Text);
    BindGrid();

    tblForm.Visible = false;
    grvPictures.Visible = true;

    btnInsertRecord.Visible = true;
}
```

Εικόνα 12: Διαχείριση επιλογής εισαγωγής φωτογραφιών

gridview και εμφανίζουμε πάλι το κουμπί εισαγωγής φωτογραφιών.

Το επόμενο event που θα διαχειριστούμε είναι της επιλογής Cancel με την οποία κρύβουμε τη φόρμα επιλογής φωτογραφιών, εμφανίζουμε το gridview καθώς και το κουμπί εισαγωγής φωτογραφιών (Insert).

Το επόμενο βήμα είναι η διαδικασία επεξεργασίας των εικόνων. Ελέγχουμε αν ο χρήστης έχει επιλέξει να επεξεργαστεί τα στοιχεία μιας εγγραφής (τίτλο, περιγραφή) ή αν έχει επιλέξει να την διαγράψει. Στην πρώτη περίπτωση αποθηκεύουμε το id της φωτογραφίας που επιθυμούμε να επεξεργαστούμε και την εμφανίζουμε τα στοιχεία της (όνομα φωτογραφίας, τίτλο, περιγραφή). Στην περίπτωση που θελήσουμε να διαγράψουμε μια φωτογραφία επιλέγουμε πάλι το id της και παίρνουμε την εγγραφή που αντιστοιχεί σε αυτό το id και την διαγράφουμε μαζί με όλες τις πληροφορίες που η εγγραφή φέρει.

```
protected void btnLogin_Click(object sender, EventArgs e)
{
    string UserName = System.Configuration.ConfigurationManager.AppSettings["Username"];
    string Password = System.Configuration.ConfigurationManager.AppSettings["Password"];
    if (txtUsername.Text.Equals(UserName) && txtPassword.Text.Equals(Password))
    {
        Session["IsAdmin"] = true;
        pnlAdmin.Visible = true;
        pnlLogin.Visible = false;
    }
    else
    {
        lblErrorMessage.Text = "<center>Username or password are not valid.</center>";
    }
}
```

Εικόνα 13: Διαχείριση επιβεβαίωσης στοιχείων χρήστη για την εισοδό του στο σύστημα

Τέλος δημιουργούμε μια string μεταβλητή τύπου UserName όπου αποθηκεύουμε το username που έχουμε ορίσει στο Web Config. Ακριβώς το ίδιο κάνουμε και για τον κωδικό. Στη συνέχεια ελέγχουμε αν το username και ο κωδικός που έχουμε ορίσει είναι τα ίδια με αυτά που έχει πληκτρολογήσει ο χρήστης στην φόρμα εισόδου. Αν

είναι σωστά τα στοιχεία εμφανίζεται το panel διαχείρισης των φωτογραφιών. Διαφορετικά εμφανίζεται κατάλληλο μήνυμα στην οθόνη μας.

Στη συνέχεια παρουσιάζεται το αρχείο xml. Σε αυτό το αρχείο είναι αποθηκευμένες οι φωτογραφίες που έχουμε επιλέξει με τα χαρακτηριστικά που οι ίδιοι έχουμε δώσει όπως τον τίτλο της φωτογραφίας και την περιγραφή καθώς και κάποια άλλα γνωρίσματα τα οποία το σύστημα μας προσδίδει αυτόματα όπως την ονομασία της φωτογραφίας και ένα μοναδικό ID.

```
Web.config Admin.aspx.cs Admin.aspx
<?xml version="1.0"?>
<Pictures>
  <Picture>
    <PictureID>5f9db6be-2f7c-4251-a5ea-a7a062ec7f99</PictureID>
    <FileName>lillianbassmanvintagefashion11-1.jpg</FileName>
    <Title>hgf</Title>
    <Description>korina</Description>
  </Picture>
  <Picture>
    <PictureID>89d612b0-cdfa-44a1-8de0-3934296327ba</PictureID>
    <FileName>Semana-10.-Fotografía-Antigua-Week-10.-Vintage-Photography.jpg</FileName>
    <Title>jhg</Title>
    <Description>jgh</Description>
  </Picture>
  <Picture>
    <PictureID>165b491f-ed61-4b94-a058-13a53a1985a0</PictureID>
    <FileName>photo-1-640x426.jpg</FileName>
    <Title>hgf</Title>
    <Description>hfg</Description>
  </Picture>
  <Picture>
    <PictureID>ed3b59c1-7458-44f3-9e88-a62eba47be9e</PictureID>
    <FileName>images.jpg</FileName>
    <Title>zsddw</Title>
    <Description>xczcz</Description>
  </Picture>
  <Picture>
    <PictureID>b330b1eb-e107-4311-9f00-f9c48f38f17e</PictureID>
    <FileName>img1.jpg</FileName>
    <Title>fdshf</Title>
    <Description>ghgjhg</Description>
  </Picture>
  <Picture>
    <PictureID>b3e06a99-e2a0-48b9-a2f2-c22e12abec05</PictureID>
    <FileName>img2.jpg</FileName>
    <Title>hggjg</Title>
    <Description>ghfjhg</Description>
  </Picture>
  <Picture>
    <PictureID>12b0926e-81c6-4061-9245-08555b0b378e</PictureID>
    <FileName>img3.jpg</FileName>
    <Title>hdsv</Title>
    <Description>ghfjsg</Description>
  </Picture>
</Pictures>
```

Εικόνα 14: Αρχείο xml

```

Default.aspx*  X Pictures.xml  Web.config  Admin.aspx.cs  Admin.aspx
</script>
<!-- Back to Top plugin -->

<p id="back-top">
  <a href="#top"><span></span></a>
</p>

<table border="0" width="100%" cellpadding="0" cellspacing="0" class="header_table">
  <tr align="center">
    <td class="header_black" id="my_gallery"><a href="#gallery">MY GALLERY</a></td>
    <td valign="bottom" class="fisthalf_logo">
      </td>
    <td class="header_black" id="about_me"><a href="#bio">ABOUT ME</a></td>
  </tr>
  <tr align="center">
    <td></td>
    <td valign="top" width="20%">
      </td>
    <td></td>
  </tr>
</table>

<div class="nav">
  <ul>
    <li class="current"><a >./NAVIGATION/./</a></li>
    <li><a href="#gallery">My Gallery</a></li>
    <li><a href="#bio">About Me</a></li>
  </ul>
</div>

<div class="search" align="center">
  <h2>Search</h2>
  <asp:TextBox ID="txtSearch" runat="server" CssClass="search_box" ></asp:TextBox><br />

  <asp:ImageButton ID="btnSearch" CssClass="search_button" Text="Search" OnClick="btnSearch_Click" ImageUrl="images/hitme.png" runat="server" />
</div>

<div align="center">
  
</div>

<div class=" wrap" >
  <a id="gallery"></a>
  <asp:Repeater ID="MyGallery" runat="server">
    <ItemTemplate>
      <div class="box">
        <a href="<#Eval("FileName") %>" rel="clearbox[gallery=My Gallery]">
          <div class="boxInner">
            " class="images" /></div>
          </div>
        </a>
      </div>
    </ItemTemplate>
  </asp:Repeater>
</div>
<!--

```

Εικόνα 15: Στιγμιότυπο από τον κώδικα του αλμπουμ

Στο παρακάτω τμήμα του κώδικα (Εικόνα 15) γίνεται η διαχείριση της αναζήτησης, δηλαδή της δυνατότητας που έχει ο χρήστης να αναζητήσει μία φωτογραφία απλά πληκτρολογώντας στο αντίστοιχο πεδίο μια λέξη κλειδί, την ονομασία της φωτογραφίας ή την περιγραφή της.

```
protected void btnSearch_Click(object sender, EventArgs e)
{
    Pictures MyPictures = new Pictures();
    DataTable Pictures = MyPictures.Select();

    foreach (DataRow Picture in Pictures.Rows)
    {
        Picture["FileName"] = ResolveUrl("~/Pictures/" + Picture["FileName"]);
    }

    if (Pictures.Select("(Title Like '%" + txtSearch.Text + "%') OR (Description Like '%" + txtSearch.Text + "%')").Count() == 0)
    {
        lblErrorMessage.Visible = true;
        MyGallery.Visible = false;
    }
    else
    {
        lblErrorMessage.Visible = false;
        MyGallery.Visible = true;

        DataTable Filtered = Pictures.Select("(Title Like '%" + txtSearch.Text + "%') OR (Description Like '%" + txtSearch.Text + "%')").CopyToDataTable();
        MyGallery.DataSource = Filtered;
        MyGallery.DataBind();
    }
}
```

Εικόνα 16: Λειτουργικότητα της αναζήτησης φωτογραφιών

Στο παρακάτω στιγμιότυπο απεικονίζεται η χρήση των media queries που προσδίδουν στο site την δυνατότητα να είναι λειτουργικό και ευπαρουσίαστο σε όλες τις αναλύσεις οθονών.

```

fault.aspx Pictures.xml Web.config Admin.aspx.cs Admin.aspx
-webkit-transition: all 0.3s ease-out;
-moz-transition: all 0.3s ease-out;
-o-transition: all 0.3s ease-out;
transition: all 0.3s ease-out;
}
body.no-touch .boxInner:hover .titleBox, body.touch .boxInner.touchFocus .titleBox {
margin-bottom: 0;
}
@media only screen and (max-width : 480px) {
/* Smartphone view: 1 tile */
.box {
width: 100%;
padding-bottom: 100%;
}
}
@media only screen and (max-width : 650px) and (min-width : 481px) {
/* Tablet view: 2 tiles */
.box {
width: 50%;
padding-bottom: 50%;
}
}
@media only screen and (max-width : 1050px) and (min-width : 651px) {
/* Small desktop / ipad view: 3 tiles */
.box {
width: 33.3%;
padding-bottom: 33.3%;
}
}
@media only screen and (max-width : 1290px) and (min-width : 1051px) {
/* Medium desktop: 4 tiles */
.box {
width: 25%;
padding-bottom: 25%;
}
}
/*THUMBNAILS*/
/*@media screen and (max-width: 600px) {
.container {
flex-direction:column;
-webkit-flex-direction:column;
-moz-flex-direction:column;
}
}*/
@media screen and (max-width: 1400px) {
.takealook {
width: auto;
margin: 60px 0;
}
}

```

Εικόνα 17: Χρήση των media queries

Στο αρχείο Picture.cs γίνεται η διαχείριση του αρχείου xml στο οποίο φορτώνουμε τις φωτογραφίες. Αρχικά όπως φαίνεται στην παρακάτω εικόνα δημιουργούμε μια μεταβλητή τύπου string με την οποία διαβάζουμε τη θέση στην οποία βρίσκεται το xml. Ελέγχουμε αν υπάρχει το αρχείο αυτό και αν δεν υπάρχει επιστρέφουμε ένα κενό πίνακα, διαφορετικά διαβάζουμε το xml αρχείο και το αποθηκεύουμε σε μια DataSet μεταβλητή η οποία αντιπροσωπεύει στην ουσία μια βάση δεδομένων στην μνήμη. Στην συνέχεια ελέγχουμε αν στο xml βρέθηκε κάποιος πίνακας και αν δεν υπάρχει, επιστρέφουμε έναν κενό πίνακα, διαφορετικά επιστρέφουμε τον πρώτο πίνακα με τις εγγραφές που βρέθηκε μέσα στο xml.

```
public static string FilePath = HttpContext.Current.Server.MapPath("~/App_Data/Pictures.xml");
public DataTable Select()
{
    // if the file does not exists return an empty DataTable
    if (!System.IO.File.Exists(FilePath))
        return new DataTable();

    // Read the xml
    DataSet MyDataSet = new DataSet();
    MyDataSet.ReadXml(FilePath);

    if (MyDataSet.Tables.Count == 0)
        return new DataTable();

    return MyDataSet.Tables[0];
}
```

Εικόνα 18: Ελέγχουμε αν υπάρχει το xml αρχείο

Στο επόμενο κομμάτι κώδικα συνεχίζονται οι ίδιοι έλεγχοι , αυτή τη φορά για μια μόνο γραμμή του πίνακα. Μετά για την πρώτη γραμμή το πίνακα MyDataSet ζητάμε να βρούμε την φωτογραφία με PictureID το ίδιο με αυτό που του έχουμε περάσει ως παράμετρο στην αρχή της μεθόδου της κλάσης μας.

```
public static DataRow SelectByPictureID(string PictureID)
{
    // if the file does not exists return an empty DataTable
    if (!System.IO.File.Exists(FilePath))
        return null;

    // Read the xml
    DataSet MyDataSet = new DataSet();
    MyDataSet.ReadXml(FilePath);

    if (MyDataSet.Tables.Count == 0)
        return null;

    DataRow[] Filtered = MyDataSet.Tables[0].Select("PictureID = '" + PictureID + "'");

    if (Filtered.Length == 0)
        return null;
    else
        return Filtered[0];
}
```

Εικόνα 19: Μέθοδος της κλάσης DataRow SelectByPictureID

Στη συνέχεια φτιάχνουμε μια συνάρτηση με παραμέτρους τα στοιχεία μιας εγγραφής δηλαδή μιας φωτογραφίας, όνομα αρχείου, τίτλος και περιγραφή. Δημιουργούμε ένα μοναδικό ID για την καινούρια εγγραφή και ελέγχουμε αν το xml υπάρχει. Σε περίπτωση που υπάρχει το φορτώνουμε διαφορετικά το δημιουργούμε. Μετά επιλέγουμε ένα πίνακα με όνομα Pictures και αν δεν υπάρχει τον δημιουργούμε. Κατόπιν φτιάχνουμε τα πεδία και τα βάζουμε στον πίνακα και τέλος αποθηκεύουμε τον πίνακα.

```
public static void Insert(string FileName, string Title, string Description)
{
    string PictureID = Guid.NewGuid().ToString();
    XmlDocument MyXmlDocument = new XmlDocument();

    if (System.IO.File.Exists(FilePath))
    {
        MyXmlDocument.Load(FilePath);
    }
    else
    {
        XmlDeclaration MyXmlDeclaration = MyXmlDocument.CreateXmlDeclaration("1.0", null, null);
        MyXmlDocument.AppendChild(MyXmlDeclaration); // Create the root element
        XmlElement root = MyXmlDocument.CreateElement("Pictures");
        MyXmlDocument.AppendChild(root);
    }

    XmlNode PicturesXmlNode = MyXmlDocument.SelectSingleNode("Pictures");

    XmlElement PictureXmlElement = MyXmlDocument.CreateElement("Picture");

    XmlElement PictureIDXmlElement = MyXmlDocument.CreateElement("PictureID");
    PictureIDXmlElement.InnerText = PictureID;
    PictureXmlElement.AppendChild(PictureIDXmlElement);

    XmlElement FileNameXmlElement = MyXmlDocument.CreateElement("FileName");
    FileNameXmlElement.InnerText = FileName;
    PictureXmlElement.AppendChild(FileNameXmlElement);

    XmlElement TitleXmlElement = MyXmlDocument.CreateElement("Title");
    TitleXmlElement.InnerText = Title;
    PictureXmlElement.AppendChild(TitleXmlElement);

    XmlElement DescriptionXmlElement = MyXmlDocument.CreateElement("Description");
    DescriptionXmlElement.InnerText = Description;
    PictureXmlElement.AppendChild(DescriptionXmlElement);

    PicturesXmlNode.AppendChild(PictureXmlElement);

    MyXmlDocument.Save(FilePath);
}
```

Εικόνα 20: Δημιουργία του πίνακα στο xml αρχείο

Στη παρακάτω εικόνα βλέπουμε ένα στιγμιότυπο από το αρχείο cb_config.js. Πρόκειται για ένα αρχείο javascript του φακέλου clearbox μιας βιβλιοθήκης που έχω χρησιμοποιήσει για την παρουσίαση των φωτογραφιών σε πλήρες μέγεθος. Σε αυτό αρχείο βρίσκονται κάποιες ρυθμίσεις τις οποίες μπορούμε να αλλάξουμε σύμφωνα με τις δικές μας επιθυμίες και τα χαρακτηριστικά που θέλουμε να δώσουμε στη σελίδα.

```

js  X Default.aspx  Pictures.xml  Web.config  Admin.aspx.cs  Admin.aspx
CB layout:

CB_WindowColor='transparent',           // color of the CB window (note: you have to change the rounded-corner PNGs too!), transparent is also working
CB_MinWidth=200,                        // minimum (only at images) or initial width of CB window
CB_MinHeight=200,                       // initial height of CB window
CB_WinPadd=15,                          // padding of the CB window from sides of the browser
CB_RoundPix=0,                          // change this setting only if you are generating new PNGs for rounded corners
CB_ImgBorder=0,                         // border size around the picture in CB window
CB_ImgBorderColor='#ddd',               // border color around the picture in CB window
CB_Padd=0,                              // padding around inside the CB window

CB_BodyMarginLeft=0,                   //
CB_BodyMarginRight=0,                  // if you set margin to <body>,
CB_BodyMarginTop=0,                    // please change these settings!
CB_BodyMarginBottom=0,                 //

CB_ShowThumbnails='off',               // it tells CB how to show the thumbnails ('auto', 'always' or 'off') thumbnails are only in picture-mode!
CB_ThumbsBGColor='#000',               // color of the thumbnail layer
CB_ThumbsBGOpacity=.35,                // opacity of the thumbnail layer
CB_ActThumbOpacity=.65,                // thumbnail opacity of the current viewed image

CB_SlideShowBarColor='#fff',           // color of the slideshow bar
CB_SlideShowBarOpacity=.5,              // opacity of the slideshow bar
CB_SlideShowBarPadd=17,                 // padding of the slideshow bar (left and right)
CB_SlideShowBarTop=2,                  // position of the slideshow bar from the top of the picture

CB_SimpleDesign='off',                  // if it's 'on', CB doesn't show the frame but only the content - really nice ;)

CB_CloseBtnTop=-10,                    // vertical position of the close button in picture mode
CB_CloseBtnRight=-14,                  // horizontal position of the close button in picture mode
CB_CloseBtn2Top=-20,                   // vertical position of the close button in content mode
CB_CloseBtn2Right=-30,                 // horizontal position of the close button in content mode

CB_OSD='off',                           // turns on OSD
CB_OSDShowReady='off',                 // when clearbox is loaded and ready, it shows an OSD message

CB font, text and navigation (at the bottom of CB window) settings:

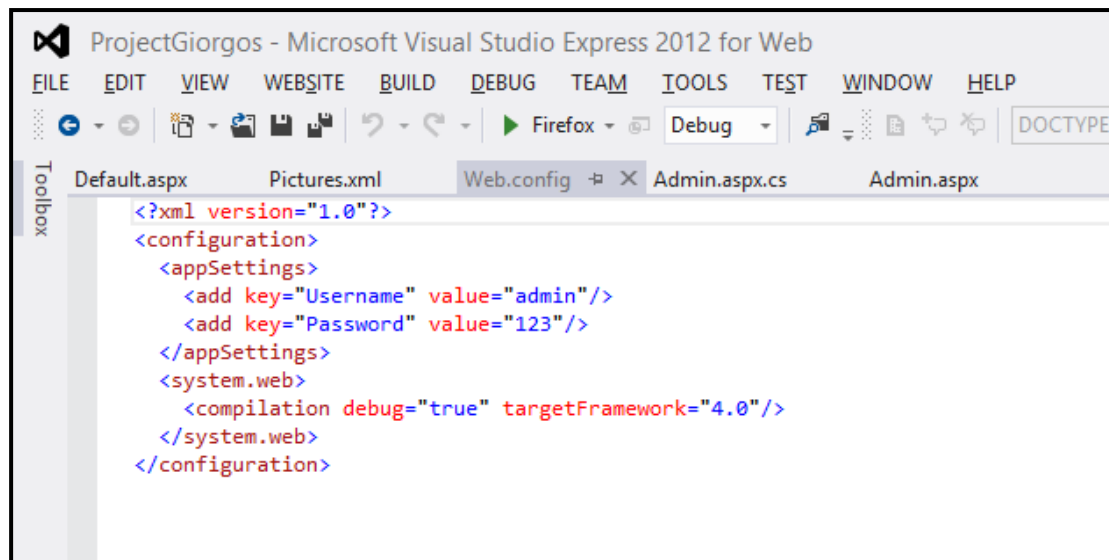
CB_FontT='helvetica, arial, sans-serif', //
CB_FontSizeT=13,                       // these variables are referring to the title or caption line
CB_FontColorT='#777',                   //
CB_FontWeightT='normal',                //

CB_FontC='helvetica, arial, sans-serif', //
CB_FontSizeC=11,                       // these variables are referring to
CB_FontColorC='#999',                   // comment lines under the title
CB_FontWeightC='normal',                //
CB_TextAlignC='justify',                 //
CB_txtHMax=120,                         // maximum height of the comment box in pixels

CB_FontG='helvetica, arial, sans-serif', //
CB_FontSizeG=11,                       // these variables are referring to the gallery name
CB_FontColorG='#aaa',                   //
CB_FontWeightG='normal',                //
    
```

Εικόνα 21: Αρχείο cb_config.js από το plugin του clearbox

Τέλος το αρχείο web.config περιλαμβάνει τις κύριες ρυθμίσεις για το website.



The screenshot shows the Microsoft Visual Studio Express 2012 for Web interface. The title bar reads "ProjectGiorgos - Microsoft Visual Studio Express 2012 for Web". The menu bar includes FILE, EDIT, VIEW, WEBSITE, BUILD, DEBUG, TEAM, TOOLS, TEST, WINDOW, and HELP. The toolbar contains various icons for navigation and development. The Solution Explorer on the left shows a project structure with files: Default.aspx, Pictures.xml, Web.config, Admin.aspx.cs, and Admin.aspx. The main editor window displays the content of the Web.config file, which is an XML configuration file. The code is as follows:

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="Username" value="admin"/>
    <add key="Password" value="123"/>
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.0"/>
  </system.web>
</configuration>
```

Εικόνα 22: Web.config αρχείο

Κεφάλαιο 6ο

6.1) Συμπεράσματα και μελλοντικές βλέψεις

Από τη σύλληψη της ιδέας της συγκεκριμένης πτυχιακής εργασίας έως και τώρα στην ολοκλήρωσή της, το κύριο συμπέρασμα είναι πως η τεχνολογία έχει μπει για τα καλά στη ζωή μας. Υπάρχουν πολλές επιλογές από εφαρμογές, γλώσσες προγραμματισμού και ένας web developer μπορεί να επιλέξει από μια μεγάλη ποικιλία. Ανάμεσα στις γλώσσες προγραμματισμού όπως διαπιστώνουμε δεν υπάρχουν πολλές διαφορές. Όλες έχουν σαν κοινό αποτέλεσμα την διευκόλυνση του χρήστη.

Σε μία εποχή πολύ μεταβαλλόμενη ο άνθρωπος προσπαθεί πολύ να βρει τις λύσεις εκείνες, που θα κάνουν τη ζωή του και καταστάσεις της ζωής του, μικρές ή μεγάλες, πιο εύκολες και πιο απλές. Σε μία τέτοια άποψη στηρίζεται και η δημιουργία ιστοσελίδων οι οποίες όχι άπλα θα καλύπτουν τις ανάγκες του κάθε χρήστη αλλά θα αποτελούν ένα μέσο διαφήμισης και προώθησης της δουλειάς του ιδιοκτήτη της σελίδας, σε ένα κοινό το οποίο πλέον έχει γίνει πιο απαιτητικό και πιο επικριτικό. Άλλωστε με την ραγδαία εξέλιξη της τεχνολογίας ιδιαίτερα στον κόσμο των υπολογιστών και μικροπολογιστών, η ανάγκη να κρατήσουμε τους χρήστες των παραπάνω ικανοποιημένους αποτελεί και το κύριο ζήτημα. Γι' αυτόν ακριβώς το λόγο και ο σχεδιασμός και η δημιουργία-ανάπτυξη ιστοτόπων πρέπει να προσαρμόζεται και να ανταποκρίνεται σε αυτές τις ανάγκες. Έτσι με βάσει αυτές τις ανάγκες δημιουργήθηκαν, εξελίχθηκαν και συνεχίζουν να εξελίσσονται οι γλώσσες εκείνες οι οποίες στα χέρια των web developers αποτελούν εργαλεία απαραίτητα για την κατασκευή προτότυπων, διαδραστικών και εύελικτων σελίδων. Γιατί αν κάποτε η εμφάνιση ενός animation ή ενός scrollable background φτιαγμένα καθαρά και μονό προγραμματιστικά, σε μια ιστοσελίδα, αποτελούσε όνειρο για κάποιους προγραμματιστές, σήμερα θεωρείται απόλυτα εφαρμόσιμο και απαραίτητο για μια σελίδα που θέλουμε να ξεχωρίσει.

Στο μέλλον θα μπορούσαν να προστεθούν στην υπάρχουσα ιστοσελίδα και κάποιες άλλες υπηρεσίες για την καλύτερη εξυπηρέτηση των χρηστών.

Μερικές από τις ιδέες παραθέτονται παρακάτω:

- ✓ θα μπορούσε ο χρήστης να οργανώσει τις φωτογραφίες του σε φακέλους,
- ✓ θα μπορούσε ο χρήστης αφού ανεβάσει την φωτογραφία του στο λογαριασμό του να του παρέχονται κάποια εργαλεία μορφοποίησης,
- ✓ θα μπορούσε ο χρήστης μέσω συνδέσμου να ανεβάσει την φωτογραφία του απευθείας από τη σελίδα στα μέσα κοινωνικής δικτύωσης (facebook, twitter, instagram κλπ.) με ένα μικρό λογότυπο που θα εντυπωνόταν αυτόματα στην φωτογραφία και έτσι θα παρείχε και διαφήμιση στο site,
- ✓ θα μπορούσε ο χρήστης μέσω από μία ομάδα συνδέσμων φωτογραφείων της περιοχής του να στέλνει το αρχείο του αμέσως για εκτύπωση στο κατάστημα της επιλογής του.

Βιβλιογραφία

Έντυπη

1. Rogers CAdenhead, Laura Lemay – Πλήρες Εγχειρίδιο της Java, εκδ.Μ-Γκιούρδας
2. Luke Welling, Laura Thomson (2002) “Ανάπτυξη Web Εφαρμογών με PHP και MySQL”, Εκδόσεις Μ. Γκιούρδας.
3. Julie C. Meloni (2009) “Μάθε PHP, MySQL και Apache, Όλα σε ένα”, 4^η Έκδοση, Εκδόσεις Γκιούρδας
4. Φώτης Λαζαρίνης – Τεχνολογίες Πολυμέσων, εκδ. Κλειδάριθμος
5. Ullman, Larry – Εισαγωγή στην PHP για τον παγκόσμιο ιστό, εκδ. Κλειδάριθμος
6. Stephen R. Davis, 1996 ‘Δουλέψτε με την Jav’
7. A.Silberschatz, H.Korth, S.Sudarsham, 2002 ,Η πλήρης θεωρία των βάσεων δεδομένων, Συστήματα βάσεων δεδομένων
8. H. M. Deitel, P. J. Deitel – C++ Προγραμματισμός, εκδόσεις Μ. Γκιούρδας
9. Sharon Zakhour, Shott Hommel, Jacob Royal, Isaac Rabinovite, Tom Risser, Mark Hoeber – Ο επίσημος οδηγός της Java, εκδ. Μ. Γκιούρδας
10. Phil Ballard, Michael Moncur – Μάθετε Ajax, Javascript & PHP, εκδ. Μ Γκιούρδας
11. G. Andrew Duthie – Microsoft ASP.NET προγραμματισμός με τη Microsoft Visual Basic .NET βήμα βήμα, εκδ. Κλειδάριθμος
12. Fast ASP.NET Websites, by Dean Alan Hume, publ. Manning

13. HTML5 Cookbook Solutions & Examples for HTML5 Developers ,
By Christopher ,Schmitt, Kyle Simpson, Publisher: O'Reilly Media
14. HTML5: The Missing Manual ,The Book That Should Have Been in the Box
By Matthew MacDonald, Publisher: O'Reilly Media
15. Dr. Andreas Mauthe; Dr. Peter Thomas (2004). Professional Content
Management Systems: Handling Digital Media Assets. John Wiley & Sons
16. J.R. Okin (2005). The Information Revolution. Ed. Ironbound Press, 350 pp.
17. ASP.NET Step by Step, Microsoft Press, Εκδόσεις Microsoft Press
18. Beginning ASP.Net 1.1 E-Commerce Cristian Darie, Karli Watson, Εκδόσεις
APress
19. David Flanagan, 2006, "Javacript, The definitive guide", 5th edition
20. Learning JQuery – Jonathan Chaffer, Karl Swedberg , Packt Publishing
21. Beginning ASP.NET 4.5 Databases, 3rd Edition – Sandeep Chanda, Damien
Foggon - Apress

Ηλεκτρονική

1. <http://en.wikipedia.org/wiki/ASP.NET>
2. <https://support.papaki.gr/>
3. <http://www.tizag.com/>
4. PHP Homepage. php.net
5. Static web page - Definition of website by the Free Online Dictionary,
Thesaurus and Encyclopedia". [Thefreedictionary.com](http://www.thefreedictionary.com)
6. [wikipedia.org/wiki/CSS](http://en.wikipedia.org/wiki/CSS)
7. <http://www.lib.teiher.gr/>

8. <http://www.webstyleguide.com/>
9. <http://www.ebusinessforum.gr/>
10. <http://nefeli.lib.teicrete.gr/>
11. http://www.w3schools.com/html/html5_intro.asp
12. <http://www.css3.info>
13. <http://www.webpagethatsuck.com/>
14. <http://www.1stwebdesigner.com/tutorials/fluid-grids-in-responsive-design/>
15. [.https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries)
16. http://en.wikipedia.org/wiki/W3C_recommendation#Specification_Maturation
17. <http://www.eeei.gr/>
18. <http://www.wpdfd.com/>
19. <http://www.techopedia.com/>
20. <http://www.yourwebskills.com/headingshtml.php>
21. <http://www.ebooks4greeks.gr/%cf%83%ce%b7%ce%bc%ce%b5%ce%b9%cf%8e%cf%83%ce%b5%ce%b9%cf%82-java>
22. <http://despreneur.com/can-a-web-developer-also-be-a-web-designer/>
23. https://foss.ntua.gr/wiki/index.php/%CE%93%CE%BB%CF%8E%CF%83%CF%83%CE%B5%CF%82_%CE%A0%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D
24. <http://eureka.lib.teithe.gr>
25. <http://goldentut.wordpress.com/2012/12/23/%CF%84%CE%AF-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-%CF%84%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1-ajax/>

Copyright © Κάντας Γεώργιος 2013

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Κρήτης.

