



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ιστοσελίδα επαγγελματικής δικτύωσης και εύρεσης εργασίας



Του φοιτητή

Γκίρσα Παύλου

Αρ. Μητρώου: 08/3383

Επιβλέπων καθηγητής

Σιάκα Κ.

Θεσσαλονίκη 2013

ΠΡΟΛΟΓΟΣ

Η πτυχιακή αυτή αποσκοπεί στην μελέτη, σχεδίαση και ανάπτυξη μιας ιστοσελίδας για επαγγελματική δικτύωση και ενημέρωση, την διαχείριση του ανθρώπινου δυναμικού και πιο συγκεκριμένα για την ευκολότερη και γρηγορότερη κατασκευή βιογραφικού σημειώματος καθώς και αναζήτησης εργασίας και στοιχεία εργαζομένων. Για την επίτευξη του στόχου αυτού μελετήθηκαν τεχνολογίες ανάπτυξης δυναμικών ιστοσελίδων. Πιο συγκεκριμένα, χρησιμοποιήθηκε η HTML και CSS για την βασική ανάπτυξη της σελίδας, η PHP, για την σύνδεση με βάση δεδομένων καθώς επίσης και την σύνδεση των διαφορετικών σελίδων μεταξύ τους. Την JavaScript, JQuery και AJAX για διάφορες επιπρόσθετες λειτουργίες. Επίσης, χρησιμοποιήθηκε το σύστημα διαχείρισης βάσης δεδομένων MySQL (με phpMyAdmin), με το οποίο και έγινε η διαχείριση της βάσης δεδομένων.

ΠΕΡΙΛΗΨΗ

Οι ιστοσελίδες επαγγελματικής δικτύωσης έχουν αρχίσει να χρησιμοποιούνται ευρέως και στην Ελλάδα τόσο από τους οργανισμούς όσο και από τους εργαζόμενους/υποψήφιους. Σκοπός της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής επαγγελματικής δικτύωσης καθώς και διαχείρισης ανθρώπινου δυναμικού με σκοπό την αναζήτηση και εύρεση εργασίας και εργαζομένων. Τα εγγεγραμμένα μέλη του έχουν τη δυνατότητα να δημιουργήσουν το προσωπικό τους προφίλ (σελίδα), να συνδεθούν με άλλους χρήστες, να τους παρακολουθούν και κατ' επέκταση να αναζητήσουν εργασία. Μπορούν στην συνέχεια να προωθήσουν το προσωπικό τους link όπου επιθυμούν, ή να αναζητήσουν εργοδότες. Στους εργοδότες δίνεται επίσης η δυνατότητα να συντάξουν ένα προφίλ και να κάνουν μια αναζήτηση στην βάση δεδομένων με τα προφίλ για να βρουν τους υποψήφιους που μπορεί να τους ενδιαφέρουν.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν κατά κύριο λόγο, η γλώσσα προγραμματισμού διαδικτύου PHP, η JavaScript, το σύστημα διαχείρισης βάσεων δεδομένων MYSQL και ο Apache Server. Τα θεωρητικά θέματα που άπτονται του αντικειμένου της πτυχιακής όπως και τα εργαλεία που χρησιμοποιήθηκαν, αναπτύσσονται αναλυτικά καθώς επίσης και όλες οι φάσεις που ακολουθήθηκαν από την ανάλυση μέχρι την υλοποίηση του συστήματος.

ABSTRACT

Professional Networking is nowadays starting more and more to be used in Greece by both employees and employers. The purpose of this thesis is the design and implementation of a web application that provides professional networking and human resource management. Employers and employees can seek and find each others. Registered members (job seekers) can create their personal profiles (page) and connect with other users and thus share knowledge and seek employment. Employers will also be able to create a profile and search the database of suitable profiles and to find candidates who they may be interested in.

The main language which was used for the development of this application was PHP, JavaScript, database management system MYSQL and Apache Server. The theoretical issues related to the object of the thesis as well as the tools used are explained in detail as well as all phases followed by the analysis to the implementation of the system.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου στην επιβλέπουσα καθηγήτρια της εργασίας μου Κα. Κέρστιν Σιάκα, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία, για την καθοδήγησή της καθ' όλη τη διάρκειά της και κυρίως για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα ενδιαφέρον αντικείμενο.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΠΕΡΙΕΧΟΜΕΝΑ	6
Ευρετήριο σχημάτων	9
ΕΙΣΑΓΩΓΗ.....	10
Χρήση	10
Σκοπός.....	10
1. Ανάλυση των τεχνολογιών διαδικτύου.....	12
ΕΙΣΑΓΩΓΗ	12
1.1 Στατικές Ιστοσελίδες	12
1.2 Δυναμικές Ιστοσελίδες.....	13
1.3 HTML	14
1.3.1 Τι είναι η HTML	14
1.3.2 HTML Elements	15
1.3.3 HTML Forms	16
1.4 PHP	19
1.4.1 Τι είναι PHP	19
1.4.2 Σύνταξη και εντολές της PHP	20
1.4.3 Οι μεταβλητές στην PHP	20
1.4.4 Η εντολή if στην PHP	21
1.4.5 PHP Loops.....	21
1.4.6 Φόρμες PHP	23
1.5 JavaScript.....	26
1.5.1 Χαρακτηριστικά της JavaScript	26
1.5.2 Χρήση της JavaScript στις ιστοσελίδες.....	29
1.6 jQuery.....	31
1.6.1 Τα πλεονεκτήματα της JQuery.....	31
1.6.2 Χρήση	32
1.7 CSS	33
1.7.1 Ανάλυση των CSS	33
1.7.2 Σύνταξη των CSS.....	33

1.7.3 Χρήση των CSS.....	34
1.7.4 Πηγές πληροφοριών στα CSS.....	35
1.8 XML.....	37
1.8.1 Ανάλυση των χαρακτηριστικών της XML.....	37
1.8.2 Χαρακτήρες και διαφυγή	39
1.8.3 XMLHttpRequest	40
1.9 Java Servlets	44
1.9.1 Κύκλος ζωής ενός Java Servlet	45
1.9.2 Servlet Containers	46
1.10 JSON	48
1.11 Ajax.....	49
1.12 Αρχείο .htaccess.....	50
ΕΠΙΛΟΓΟΣ	54
2. Εργαλεία ανάπτυξης δυναμικών εφαρμογών	55
ΕΙΣΑΓΩΓΗ	55
2.1 AdobeDreamweaver	55
2.1.1 Τι είναι το Dreamweaver	55
2.1.2 Χαρακτηριστικά του Dreamweaver	56
2.2 XAMPP.....	57
2.3 Apache HTTP εξυπηρετητής.....	58
2.4 MySql.....	59
2.4.1 Πλεονεκτήματα της MySql.....	59
2.5 Λογισμικό PhpMyAdmin	60
2.5.1 Τι είναι το PhpMyAdmin	60
2.5.2 Χαρακτηριστικά του PhpMyAdmin	61
2.6 Adobe Photoshop.....	62
ΕΠΙΛΟΓΟΣ	63
ΚΕΦΑΛΑΙΟ 3.....	64
Σχεδιασμός Βάσεων Δεδομένων.....	64
ΕΙΣΑΓΩΓΗ	64
3.1 Βασικές Έννοιες.....	64
3.1.1 Τι είναι οι βάσεις δεδομένων	64
3.1.2 Τι είναι το Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) ή Database Management System (DBMS);.....	65

3.1.3 Τι είναι το Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή Database System(DBS);	65
3.1.4 Τι είναι ο DataBase Administrator (DBA) ή διαχειριστής μιας Βάσης δεδομένων (ΔΒΔ) και ποιος ο ρόλος του;	65
3.2 Αρχιτεκτονική τριών επιπέδων	65
3.3 Μορφές Βάσεων Δεδομένων	66
3.4 Στοιχεία Βάσεων Δεδομένων	69
3.5 Συνηθισμένα προβλήματα ή δυσκολίες που συναντώνται στις ΒΔ	70
3.6 Σύνδεση στην Βάση Δεδομένων με την PHP	72
3.6.1 Εκτέλεση SQL Ερωτημάτων (Queries) με την PHP	73
3.6.2 Εκτέλεση SQL Ερωτημάτων (Queries) με συναρτήσεις συ-γκεντρωτικών αποτελεσμάτων (Aggregate Functions) με την PHP	79
3.6.3 Εκτέλεση Προχωρημένων SQL Ερωτημάτων (Queries) με την PHP.....	82
ΕΠΙΛΟΓΟΣ	83
ΚΕΦΑΛΑΙΟ 4.....	84
Υλοποίηση Ιστοσελίδας	84
ΕΙΣΑΓΩΓΗ	84
4.1 Ανάλυση προβλήματος	84
4.1.1 Πλατφόρμα υλοποίησης	84
4.1.2 Θέματα ασφάλειας του χρήστη	84
4.1.3 Προσωπική σελίδα του χρηστη.....	85
4.2 Γενική διεύθυνση ιστοτόπου	86
4.3 Βάση Δεδομένων Εφαρμογής	95
4.3.1 Πίνακας Users	95
4.3.2 Πίνακας Fields	95
4.3.3 Πίνακας Fields	96
4.4 Ανάλυση Κώδικα Εφαρμογής.....	97
5. ΣΥΜΠΕΡΑΣΜΑΤΑ	105
5.1 Προτάσεις διαχείριση της σελίδας	105
5.2 Συμπεράσματα τεχνολογιών.....	105
5.3 Επεκτάσεις.....	105
5.4 Σύνοψη.....	106
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	107
Πηγές.....	107

Ευρετήριο σχημάτων

Εικόνα 1 «Κύκλος ζωής ενός Java Servlet»	45
Εικόνα 2 «Παράδειγμα κώδικα Json»	48
Εικόνα 3 «Αρχική οθόνη Dreamweaver»	55
Εικόνα 4 «Περιβαλλον διαχείρισης Dreamweaver»	56
Εικόνα 5 «Το Control Panel του XAMPP»	58
Εικόνα 6 «Σελίδα διαχείρισης της εφαρμογής phpMyAdmin»	62
Εικόνα 7 «Επίπεδο άποψης»	66
Εικόνα 8 «Ιεραρχικό Μοντέλο»	67
Εικόνα 9 «Δικτυωτές Βάσεις Δεδομένων».....	68
Εικόνα 10 «Σχεσιακές Βάσεις Δεδομένων»	69
Εικόνα 11 «Κεντρική Σελίδα».....	86
Εικόνα 12 «Σελίδα ανακτησης κωδικού πρόσβασης»	87
Εικόνα 13 «Κεντρική Σελίδα του χρήστη»	87
Εικόνα 14 «Tutorial»	88
Εικόνα 15 «Upload»	89
Εικόνα 16 «Βασικές πληροφορίες χρήστη»	89
Εικόνα 17 «Μέρος βιογραφικού».....	90
Εικόνα 18 «Επεξεργασία πεδίου».....	90
Εικόνα 19 «Κουμπιά προσθήκης κατηγορίας και πεδίου»	90
Εικόνα 20 «QR Code και Social Network Icons»	91
Εικόνα 21 «Follow»	91
Εικόνα 22 «Λίστα φίλων χρήστη»	92
Εικόνα 23 «Κεντρική μπάρα και αναζήτηση»	92
Εικόνα 24 «Σελίδα επεξεργασίας στοιχείων χρήστη»	93
Εικόνα 25 «Επεξεργασίας προσωπικών στοιχείων χρήστη»	93
Εικόνα 26 «Στοιχεία επικοινωνίας χρήστη»	93
Εικόνα 27 «Αναζήτηση»	94
Εικόνα 28 «Αναζήτηση με χάρτη»	94
Εικόνα 29 «Πίνακας Users»	95
Εικόνα 30 «Πίνακας Fields»	95

ΕΙΣΑΓΩΓΗ

Χρήση

Η χρήση της ιστοσελίδας λειτουργεί κυρίως ως ηλεκτρονική επαγγελματική κάρτα, με περαιτέρω επιλογές εξωστρέφειας και διαφάνειας. Δηλαδή, προσφέρει σε ηλεκτρονικό επίπεδο δυνατότητες δικτύωσης, ενημέρωσης, πρόσβασης και κατ' επέκταση, εύρεσης εργασίας. Αν κάποιος σας ζητήσει επαγγελματική κάρτα, μπορείτε να διευκρινίσετε την SeekMe ταυτότητά σας.

Έκαστο μέλος μπορεί να συνδεθεί και να επικοινωνήσει με επαγγελματίες της αγοράς, να ερευνήσει νέους πιθανούς ορίζοντες και να αποκτήσει πρόσβαση προς άτομα ή τμήματα εταιρειών που ασκούν σημαντική επιρροή στο εταιρικό γίνεσθαι κάθε οργανισμού.

Σκοπός

Ένα από τα προβλήματα που αντιμετωπίζουν οι χρήστες κατά την πλοήγηση τους στις διάφορες σελίδες του διαδικτύου, είναι η αναμονή επαναφόρτωσης όλης της ιστοσελίδας σε οποιαδήποτε αλλαγή του περιεχομένου αυτής. Θα λύσουμε το πρόβλημα αυτό με τις σύγχρονες τεχνολογίες και συγκεκριμένα με την τεχνολογία AJAX καθώς τα δεδομένα και οι πληροφορίες της ιστοσελίδας ανακτούνται και αποθηκεύονται από το διακομιστή ασύγχρονα στο παρασκήνιο ενώ παράλληλα θα εμφανίζονται στην σελίδα μας με την Javascript και την JQuery. Έτσι οι χρήστες μπορούν να πλοηγούνται και να αλλάζουν γρήγορα όλο το περιεχόμενο της ιστοσελίδας.

Σκοπός της πτυχιακής εργασίας αυτής είναι η εξοικείωση με τις βάσεις δεδομένων και ανάπτυξη ιστοσελίδων με νέες τεχνολογίες. Επίσης, η επέκταση και εμπάθυνση προηγούμενων γνώσεων σε γλώσσες σχεδίασης και προγραμματισμού (π.χ. html, php, JavaScript, JQuery κλπ), γλώσσες ανάπτυξης βάσεων δεδομένων (π.χ. MySQL), σύγχρονες τεχνολογίες διαδικτύου (π.χ. AJAX) και βοηθητικά εργαλεία (phpMyAdmin, Photoshop, Dreamweaver κ.α.).

Στην συνέχεια θα περιγραφούν αναλυτικά όλα τα βήματα που ακολουθήθηκαν για την πραγματοποίηση αυτής της εργασίας.

- Στο πρώτο κεφαλαίο γίνεται ανασκόπηση των τεχνολογιών διαδικτύου που χρησιμοποιήθηκαν για την διεκπεραίωση της πτυχιακής. Όπως επίσης και στις γλώσσες προγραμματισμού που αναπτύχθηκαν.
- Το δεύτερο παρουσιάζει περιληπτικά τις θεωρίες και τα εργαλεία που επιλέχθηκαν να χρησιμοποιηθούν για τον σχεδιασμό και την υλοποίηση αυτής της εφαρμογής.

- Το τρίτο κεφάλαιο περιέχει τον σχεδιασμό των βάσεων δεδομένων.
- Το τέταρτο κεφάλαιο παρουσιάζει την ίδια ιστοσελίδα καθώς και την διαδικασία υλοποίησης.
- Το πέμπτο κεφάλαιο αποτελεί μία σύνοψη της εργασίας και παρατίθενται κάποια συμπεράσματα καθώς και κάποιες πιθανές επεκτάσεις της εφαρμογής.
- Έπειτα, ακολουθεί η βιβλιογραφία και οι πηγές μελέτης.

1. Ανάλυση των τεχνολογιών διαδικτύου

ΕΙΣΑΓΩΓΗ

Εδώ γίνεται ανασκόπηση των τεχνολογιών διαδικτύου που χρησιμοποιήθηκαν για την διεκπεραίωση της πτυχιακής. Όπως επίσης και στις γλώσσες προγραμματισμού που αναπτύχθηκαν.

1.1 Στατικές Ιστοσελίδες

Οι δύο μεγάλες κατηγορίες ιστοσελίδων είναι οι λεγόμενες στατικές (static) και οι δυναμικές (dynamic). Όσον αφορά την όψη/εμφάνισή τους, δεν έχουν μεγάλες διαφορές μεταξύ τους, αλλά όσον αφορά την λειτουργία και τις δυνατότητές τους οι διαφορές είναι πολλές και ουσιαστικές.

Οι στατικές ιστοσελίδες είναι απλά HTML έγγραφα. Μπορούν να περιέχουν κείμενα, εικόνες, υπερσυνδέσμους (links) π.χ. μενού επιλογών και άλλα στοιχεία. Ο όρος "στατική" δεν αναφέρεται σε κάποιο είδος κίνησης, αλλά στην μονιμότητα του περιεχομένου της ιστοσελίδας. Δηλαδή, για μία στατική ιστοσελίδα το περιεχόμενο είναι δεδομένο κάθε φορά που εμφανίζεται. Για να αλλάξει το περιεχόμενό της στατικής ιστοσελίδας, θα πρέπει να γίνει επεξεργασία της με κάποιο πρόγραμμα κατασκευής ιστοσελίδων.

Το κύριο μειονέκτημά τους είναι ότι δεν είναι πρακτική/βολική η χρήση τους όταν χρειάζεται το περιεχόμενο της ιστοσελίδας να αλλάζει συχνά, π.χ. τουλάχιστον μία φορά την εβδομάδα να προστίθεται κάποιο άρθρο, ή να αλλάζουν τα προϊόντα κτλ. Αυτό συμβαίνει γιατί για να τροποποιηθεί μια στατική ιστοσελίδα, πρέπει να γίνουν επεμβάσεις απευθείας πάνω στην ιστοσελίδα, με κάποιο σχετικό πρόγραμμα (html editor είτε απλό είτε WYSIWYG), μια διαδικασία δηλαδή όπως το να σβήνεις και να γράφεις σε ένα τετράδιο, που όμως χρειάζεται μια σχετική εμπειρία. Μία στατική ιστοσελίδα, είναι κατάλληλη κυρίως για μικρές παρουσιάσεις στις οποίες σπάνια θα χρειαστεί κάποια μελλοντική αλλαγή στο περιεχόμενό της ή προσθήκη επιπλέον περιεχομένου. Σε αντίθετη περίπτωση, είναι πιο πρακτική η χρήση δυναμικών ιστοσελίδων. [8]

1.2 Δυναμικές Ιστοσελίδες

Οι δυναμικές ιστοσελίδες, σε αντίθεση με τις στατικές ιστοσελίδες, δεν είναι απλά HTML έγγραφα, αλλά συμπεριλαμβάνουν προγραμματισμό (σε μία γλώσσα προγραμματισμού κατάλληλη για το διαδίκτυο, όπως π.χ. είναι η php) και ουσιαστικά είναι web εφαρμογές.

Η κατασκευή δυναμικών ιστοσελίδων είναι προφανώς πιο πολύπλοκη από τις απλές στατικές ιστοσελίδες, και ο βαθμός δυσκολίας τους εξαρτάται φυσικά από τις λειτουργίες και τις δυνατότητες που περιλαμβάνει η εκάστοτε web εφαρμογή. Στις δυναμικές ιστοσελίδες, το περιεχόμενο της ιστοσελίδας, αποθηκεύεται και αντλείται δυναμικά από μία ή περισσότερες βάσεις δεδομένων (π.χ. MySQL).

Η χρήση των βάσεων δεδομένων, είναι αυτή που επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου στις δυναμικές ιστοσελίδες, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά την ίδια την ιστοσελίδα, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο στην βάση δεδομένων και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον "μηχανισμό" της ιστοσελίδας.

Γενικά, αυτό που προσφέρουν οι δυναμικές ιστοσελίδες, είναι μεγαλύτερη αλληλεπίδραση του χρήστη με την σελίδα π.χ. να προσθέτει τα σχόλια του στην σελίδα, αλλά και πολλές ευκολίες στον χειριστή του περιεχομένου της ιστοσελίδας π.χ. τον ιδιοκτήτη της σελίδας. Πάντως το γεγονός ότι μια δυναμική ιστοσελίδα προσφέρει περισσότερες δυνατότητες, δεν σημαίνει ότι αυτές είναι απαραίτητες σε όλους, δηλαδή σε αρκετές περιπτώσεις, μία στατική ιστοσελίδα μπορεί να καλύπτει πλήρως τις ανάγκες μιας συνοπτικής παρουσίασης.

Από πλευράς κόστους, η στατική ιστοσελίδα είθισται να είναι η φτηνή επιλογή, καθώς είναι πιο απλή η κατασκευή της ιστοσελίδας, ενώ οι δυναμικές ιστοσελίδες λόγω της πολυπλοκότητας τους κοστίζουν ακριβότερα και αυτό είναι λογικό τουλάχιστον όταν γίνονται κατά παραγγελία. Υπάρχει βέβαια και η περίπτωση υλοποίησης μιας δυναμικής ιστοσελίδας με την χρήση κάποιας open source εφαρμογής (CMS) η οποία διατίθεται δωρεάν μέσω του internet και σε αυτήν την περίπτωση η κατασκευή της ιστοσελίδας μπορεί να έχει μηδενικό κόστος (αν γίνει self-service) ή να υπάρξει κάποια χρέωση (η τελική τιμή μιας ιστοσελίδας είναι υποκειμενική υπόθεση) αν η εγκατάσταση και η τυχόν παραμετροποίηση της δωρεάν εφαρμογής ανατεθεί σε κάποια εταιρία. [8]

1.3 HTML

1.3.1 Τι είναι η HTML

Η HTML ή HyperText Markup Language έχει σχεδιαστεί για να καθορίζει τη λογική οργάνωση ενός εγγράφου, με σημαντικές προεκτάσεις στο hypertext. Δεν έχει σχεδιαστεί για να είναι η γλώσσα ενός επεξεργαστή, όπως το Word. Η επιλογή αυτή έγινε γιατί το ίδιο έγγραφο HTML μπορεί να προβληθεί από πολλούς διαφορετικούς “browsers”, με διαφορετικές ιδιότητες. Έτσι, για παράδειγμα, HTML μας επιτρέπει να επισημάνουμε τις επιλογές του κειμένου ως τίτλοι ή παράγραφοι, και στη συνέχεια να αφήνει την ερμηνεία αυτών σημειώνοντας τα στοιχεία μέχρι και το πρόγραμμα περιήγησης. Για παράδειγμα, ένα πρόγραμμα περιήγησης μπορεί κόψει την αρχή μιας παραγράφου, ενώ κάποιος άλλος μπορεί να αφήσει μόνο μια κενή γραμμή.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σε γλώσσες όπως η, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου. [9]

1.3.2 HTML Elements

Οι ιδιότητες της HTML μπορούν να διαιρούν το κείμενο ενός εγγράφου σε πακέτα που ονομάζονται elements. Τα πακέτα αυτά μπορούν να χωριστούν σε δύο ευρείες κατηγορίες - εκείνα που καθορίζουν το πώς το “body” του εγγράφου θα εμφανίζεται από το πρόγραμμα περιήγησης, καθώς και τις διατάξεις που ορίζουν τις πληροφορίες για το έγγραφο, όπως ο τίτλος ή οι σχέσεις με άλλα έγγραφα. Οι λεπτομερείς κανόνες για την HTML (τα ονόματα των ετικετών / στοιχείων, το πώς μπορούν να χρησιμοποιηθούν), ορίζεται με άλλη γλώσσα, γνωστή ως η πρότυπη γλώσσα γενικευμένης σήμανσης, ή SGML.

Τα στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος απόετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» (<) και «μικρότερο από» (>), δηλαδή < και >.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: <tag attribute1="value1" attribute2="value2">content</tag>. Μερικά στοιχεία HTML περιγράφονται ως άδεια στοιχεία, έχουν τη μορφή <tag attribute1="value1" attribute2="value2" >, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο «/», η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

Παράδειγμα:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

(Το κείμενο ανάμεσα στο <html> και το </html> περιγράφει την ιστοσελίδα, και το κείμενο μεταξύ του <body> και του </body> είναι το ορατό μέρος της. Το σημασμένο κείμενο '<title>Hello HTML</title>' καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser). Το Document Type Declaration στον

πιο πάνω κώδικα είναι για την HTML5. Αν δεν συμπεριλαμβάνεται η δήλωση `<!doctype html>`, τότε μερικοί browser θα καταφύγουν στην λειτουργία quirks για την εμφάνιση της σελίδας.

1.3.3 HTML Forms

Οι φόρμες HTML χρησιμοποιούνται για να περάσουμε δεδομένα σε ένα server. Μια φόρμα μπορεί να περιέχει στοιχεία εισόδου, όπως text fields, checkboxes, radiobuttons, submit buttons και πολλά άλλα. Μια φόρμα μπορεί να περιέχει επίσης λίστες για να επιλέγουμε ανάμεσα σε καταλόγους. Το πιο σημαντικό στοιχείο της φόρμας είναι το στοιχείο εισόδου. Το στοιχείο εισόδου χρησιμοποιείται για να επιλέξουμε πληροφορίες του χρήστη. Ένα στοιχείο εισόδου μπορεί να διαφέρει με πολλούς τρόπους, ανάλογα με τον τύπο της μεταβλητής. Ένα στοιχείο εισόδου μπορεί να είναι πεδίο κειμένου, πεδίο κωδικού πρόσβασης, κουμπιά και πολλά άλλα. [6]

1.3.3.1 Πεδία κειμένου

`<input type="text" />` ορίζει ένα one-line πεδίο εισόδου όπου ένας χρήστης μπορεί να εισάγει κείμενο

Παράδειγμα:

```
<form>
First name: <input type="text" name="firstname" /><br />
Last name: <input type="text" name="lastname" />
</form>
```

Το συγκεκριμένο παράδειγμα στον browser εμφανίζεται ως εξής:

First name:
Last name:

1.3.3.2 Πεδίο κωδικού πρόσβασης

`<input type="password" />` ορίζει ένα πεδίο κωδικού πρόσβασης

Παράδειγμα:

```
<form>
Password: <input type="password" name="pwd" />
</form>
```


Το συγκεκριμένο παράδειγμα στον browser εμφανίζεται ως εξής:

Password:

1.3.3.2 Κουτιά Επιλογής Checkboxes

`<input type="checkbox" />` ορίζει ένα πλαίσιο ελέγχου. Το checkbox επιτρέπει σε έναν χρήστη να επιλέξει μία ή περισσότερες επιλογές από έναν περιορισμένο αριθμό επιλογών.

Παράδειγμα:

```
<form>
<input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
<input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

Το συγκεκριμένο παράδειγμα στον browser εμφανίζεται ως εξής:

I have a bike
 I have a car

1.3.3.3 Κουτιά Υποβολής (Submit Buttons)

`<input type="submit" />` ορίζει ένα κουμπί υποβολής. Ένα κουμπί υποβολής χρησιμοποιείται για την αποστολή δεδομένων φόρμας σε ένα server. Τα δεδομένα που αποστέλλονται στη σελίδα καθορίζονται από τη μεταβλητή "action" της φόρμας. Το αρχείο που ορίζεται στη μεταβλητή "action" συνήθως επεξεργάζεται το δεδομένο που λαμβάνει στην είσοδο. [6]

Παράδειγμα:

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

Το συγκεκριμένο παράδειγμα στον browser εμφανίζεται ως εξής:

Username:

Εάν πληκτρολογήσουμε ορισμένους χαρακτήρες στο πεδίο κειμένου παραπάνω, και κάνουμε κλικ στο κουμπί "Submit", το πρόγραμμα περιήγησης θα στείλει το αποτέλεσμα μας σε μια σελίδα που ονομάζεται "html_form_action.asp". Η σελίδα θα μας δείξει το δεδομένο που λάβαμε στην είσοδο.

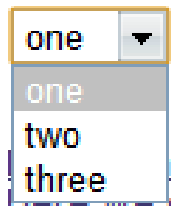
1.3.3.4 Λίστες με προεπιλεγμένη τιμή (Drop down lists with a pre-selected value)

Ορίζουμε τη drop down list χρησιμοποιώντας "select" και "option" ετικέτες. Το πλαίσιο επιλογής καθορίζεται με βάση την ετικέτα "select". Ο αριθμός των επιλογών ορίζεται χρησιμοποιώντας την ετικέτα "option". Το όνομα που θα εμφανίζεται στην επιλογή δίνεται ανάμεσα στις ετικέτες "option".

Παράδειγμα:

```
<html>
<body>
  <form name=myform>
    <select name=mytextarea>
      <option name=one value=one> one </option>
      <option name=two value=two> two </option>
      <option name=three value=three> three </option>
    </select>
  </form>
</body>
</html>
```

Το συγκεκριμένο παράδειγμα στον browser εμφανίζεται ως εξής:



1.4 PHP



1.4.1 Τι είναι PHP

Η γλώσσα PHP (Hypertext PreProcessor) είναι μια γλώσσα προγραμματισμού που μπορούμε να χρησιμοποιήσουμε για τη δημιουργία εφαρμογών Web. Μια εφαρμογή Web μπορεί να είναι οτιδήποτε από μία απλή εφαρμογή "contact us" μέσα από ένα πλήρες σύστημα blogging, ηλεκτρονικό κατάστημα, ή και σύστημα forum. Η γλώσσα Php είναι γνωστή ως μια γλώσσα προγραμματισμού server-side. Αυτό σημαίνει ότι λειτουργεί με το Web server. Οι περισσότερες γλώσσες προγραμματισμού Web είναι server-side, αλλά μερικές, όπως ή Javascript που θα δούμε παρακάτω είναι client-side, πράγμα που σημαίνει ότι λειτουργούν με το πρόγραμμα περιήγησης.

Μία server-side γλώσσα μας δίνει περισσότερη ευελιξία, δεδομένου ότι μπορούμε να κάνουμε πολλά πράγματα που είναι δύσκολο να κάνουμε με JavaScript - παραδείγματος χάριν, δουλεύοντας σε συνεργασία με αρχεία και βάσεις δεδομένων, ή επεξεργασία εικόνων. Η γλώσσα προγραμματισμού Php έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα.

Με την τεχνολογία της Php μπορούμε να δημιουργήσουμε δυναμικές ιστοσελίδες (dynamic web pages). Επίσης ο Server-side κώδικας είναι πιθανών πιο ασφαλής από ό, τι κώδικα JavaScript. Δεδομένου ότι ο JavaScript κώδικας αποστέλλεται στο πρόγραμμα περιήγησης μέσω Web είναι εύκολο για έναν επισκέπτη να δει και να επεξεργαστεί τον κώδικα. Όμως ο Server-side κώδικας, από την άλλη, παραμένει στο Web server και δεν είναι προσβάσιμος για τους επισκέπτες του site. [2][14]

1.4.2 Σύνταξη και εντολές της PHP

Έχουμε τον παρακάτω κώδικα της Php

```
<?php  
echo "<html><body>"; echo "Hello";  
echo "</body></html>";  
?>
```

Με αυτό το κομμάτι κώδικα θα δούμε το μήνυμα Hello στην οθόνη μας, ενώ αν επιχειρήσουμε να δούμε τον πηγαίο κώδικα (source code) θα υπάρχουν μόνο τα tags της HTML, χωρίς τις σημάνσεις της Php.

Οι ετικέτες (tags) <?php και ?> χρησιμοποιούνται για να δηλώσουμε ένα μπλοκ κώδικα Php. Για να είναι σε θέση ο web server να επεξεργαστεί τον κώδικα της Php, πρέπει να του πούμε πού αρχίζει και πού τελειώνει ένα μπλοκ κώδικα Php. Μπορούμε να τοποθετήσουμε κώδικα της HTML μέσα στον κώδικα της Php, ο οποίος θα εκτελεσθεί κανονικά.

Το παραπάνω παράδειγμα θα μπορούσε να είχε γραφεί και ως εξής :

```
<html><body>  
<?php  
echo "Hello"; ?>  
</body></html>
```

Η εντολή echo χρησιμοποιείται για να στείλουμε ένα κείμενο (string) στον φυλλομετρητή (browser). Όλες οι εντολές της Php πρέπει να τελειώνουν με τον χαρακτήρα ; και μια εντολή μπορεί να επεκταθεί και σε περισσότερες από μία γραμμές. Όταν ο φυλλομετρητής ενός χρήστη ζητήσει μια σελίδα Php σαν την παραπάνω, ο server θα την επεξεργαστεί, θα μετατρέψει τον κώδικα Php σε καθαρή HTML μορφή και έτσι ο χρήστης δεν θα μπορέσει να δει τον αρχικό κώδικα Php.

1.4.3 Οι μεταβλητές στην PHP

Στην Php μπορούμε να χρησιμοποιήσουμε και μεταβλητές (variables) για να αποθηκεύσουμε και να ανακτήσουμε δεδομένα που χρησιμοποιούμε συχνά. Όλες οι μεταβλητές πρέπει να αρχίζουν με τον χαρακτήρα \$ και ένα απλό παράδειγμα κώδικα Php που χρησιμοποιεί μεταβλητές και εμφανίζει το ίδιο αποτέλεσμα με το πρώτο παράδειγμα είναι το εξής :

```
<?php $string="Hello";  
echo "<html><body>";  
echo "$string";  
echo "</body></html>";  
?>
```

Το όνομα μιας μεταβλητής μπορεί να αποτελείται από γράμματα και ψηφία, αλλά πρέπει να ξεκινά οπωσδήποτε με γράμμα. Μπορεί να περιέχει και τον ειδικό χαρακτήρα _ (underscore). Οι μεταβλητές της Php μπορούν να περιέχουν κείμενο ή και αριθμούς :

```
$string1="Florina";  
$number1=2;
```

1.4.4 Η εντολή if στην PHP

Στο παρακάτω παράδειγμα, ο έλεγχος μεταβιβάζεται σε μία από τρεις εντολές ανάλογα με την τρέχουσα ώρα.

```
<?php  
$date=date("H");  
echo "<html><body>"  
if ($date<12) echo "Καλημέρα";  
else if ($date<18) echo "Καλό απόγευμα";  
else echo "Καληνύχτα";  
echo "</body></html>";  
?>
```

Η συνάρτηση date("H") της Php επιστρέφει την ώρα του υπολογιστή σε 24ωρη μορφή και έτσι η μεταβλητή \$date θα περιέχει ένα string με τιμές από 0 έως 23, το οποίο μπορούμε να χειριστούμε σαν έναν κανονικό αριθμό.

1.4.5 PHP Loops

Συχνά θέλουμε τον ίδιο κομμάτι κώδικα να τρέξει ξανά και ξανά σε μια σειρά. Αντί για την προσθήκη διαφόρων σχεδόν ίδιων γραμμών κώδικα σε ένα σενάριο, μπορούμε να χρησιμοποιούμε βρόχους για να εκτελέσει μια εργασία.

Στην PHP, έχουμε τις ακόλουθες looping δηλώσεις:

- while - βρόχος με ένα κομμάτι του κώδικα, ενώ μια συγκεκριμένη συνθήκη είναι αληθής.
- do ... while - βρόχος με ένα κομμάτι κώδικα για μία φορά, και στη συνέχεια να επαναλαμβάνεται ο βρόχος εφ' όσον ένας προκαθορισμένος όρος είναι αληθής.
- for - βρόχος με ένα κομμάτι κώδικα ένα συγκεκριμένο αριθμό επαναλήψεων.
- foreach - βρόχος με ένα κομμάτι κώδικα για κάθε στοιχείο σε μια συστοιχία.

1.4.5.1 Η εντολή *for* στην PHP

Το παρακάτω παράδειγμα δείχνει τη χρήση της εντολής *for* στην Php.

```
for (init; condition; increment) {  
    code to be executed;  
}>
```

Παράμετροι:

- *init*: Κυρίως χρησιμοποιείται για να ορίσουμε έναν μετρητή (αλλά μπορεί να είναι και οποιοδήποτε κομμάτι κώδικα που θα εκτελεστεί μία φορά στην αρχή του βρόχου)
- *condition*: Αξιολογείται η αληθής κατάσταση για κάθε επανάληψη στο βρόχο. Αν πάρει την τιμή TRUE, το loop συνεχίζει. Αν πάρει την τιμή FALSE, ο βρόχος τελειώνει.
- *increment*: Κυρίως χρησιμοποιείται για την αύξηση έναν μετρητή (αλλά μπορεί να είναι ο- ποιοδήποτε κομμάτι κώδικα που θα εκτελεστεί στο τέλος του βρόχου)

1.4.5.2 Η εντολή *foreach* στην PHP

Το παρακάτω παράδειγμα δείχνει τη χρήση της εντολής *foreach* στην Php

```
foreach ($array as $value) {  
    code to be executed;  
}
```

Η *foreach* χρησιμοποιείται για βρόχους μέσα από συστοιχίες (arrays). Για κάθε επανάληψη του βρόχου, η αξία του τρέχοντος στοιχείου του πίνακα που έχει ανατεθεί στο *\$value* (και ο δείκτης του πίνακα αυξανετε κατα ένα) - έτσι για την επόμενη επανάληψη του βρόχου, θα μπορούμε να δούμε την επόμενη συστοιχία.

1.4.5.3 Η εντολή *while* στην PHP

Ο βρόχος *while* εκτελεί ένα κωμάτι κώδικα, με προϋπόθεση να είναι αλήθης η συνθήκη μέσα στο *while*.

```
while ( condition ) {  
    code to be executed
```

```
;}
```

1.4.6 Φόρμες PHP

Το πιο σημαντικό πράγμα που πρέπει να παρατηρήσουμε όταν ασχολούμαστε με φόρμες HTML και PHP είναι ότι οποιοδήποτε στοιχείο της φόρμας σε μια σελίδα HTML θα γίνει δι-αθέσιμο στο PHP script μας.

Το παράδειγμα που ακολουθεί περιέχει μία HTML φόρμα με δύο πεδία εισαγωγής και ένα κουμπί.

Παράδειγμα:

```
<html>
<body>
  <form action="welcome.php" method="post">
    Name: <input type="text" name="fname" />
    Age: <input type="text" name="age" />
    <input type="submit" />
  </form>
</body>
</html>
```

Όταν ένας χρήστης συμπληρώνει τη φόρμα και κάνει κλικ στο κουμπί "Submit", τα δεδομένα της φόρμας αποστέλλονται σε ένα αρχείο PHP, που ονομάζονται "welcome.php": Το "welcome.php" είναι κάπως έτσι :

```
<html>
<body>
Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

Και το αποτέλεσμα θα ήταν κάπως έτσι:

```
Welcome Nikos!
You are 25 years old.
```

Επικύρωση Φόρμας

Το input του χρήστη θα πρέπει να επικυρώνετε από το πρόγραμμα περιήγησης όποτε είναι αυτο δυνατόν. Η επικύρωση από το browser είναι πιο γρήγορη και μειώνει το φόρτο εργασίας του server. Θα πρέπει να εξεταστεί η επικύρωση του διακομιστή, εάν το input του χρήστη θα ενσωματώνεται σε βάση δεδομένων. Ένας

καλός τρόπος για την επικύρωση μιας φόρμας στο διακομιστή είναι να έχουμε την ίδια τη φόρμα στον, αντί την αλλαγή σε διαφορετική σελίδα. Ο χρήστης θα λάβει το μήνυμα λάθους στην ίδια σελίδα με τη φόρμα. Αυτό το καθιστά ευκολότερο να ανακαλύψει το σφάλμα.

1.4.6.1 Η μεταβλητή \$_GET στην PHP

Η \$_GET μεταβλητή χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα με τιμή: method="get". Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο GET είναι ορατές σε όλους (θα εμφανίζεται στη γραμμή διευθύνσεων του browser) και οριοθετεί την

ποσότητα των πληροφοριών για την αποστολή.

Παράδειγμα:

```
<form action="welcome.php" method="get">
  Name: <input type="text" name="fname" />
  Age: <input type="text" name="age" />
  <input type="submit" />
</form>
```

Όταν ο χρήστης χρησιμοποιήσει το κουμπί "Submit", η διεύθυνση URL που αποστέλλεται στο διακομιστή θα μπορούσε να είναι κάπως έτσι: <http://welcome.php?fname=Giannis&age=25>

Το "welcome.php" αρχείο μπορεί πλέον να χρησιμοποιεί τη \$_GET μεταβλητή για τη συλλογή δεδομένων φόρμας (τα ονόματα των πεδίων της φόρμας θα γίνονται αυτόματα τα κλειδιά στη συστοιχία \$_GET):

```
Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
```

Πότε να χρησιμοποιούμε τη μέθοδο = "get"; Όταν χρησιμοποιούμε τη μέθοδο = "get" στις HTML φόρμες, όλα τα ονόματα των μεταβλητών και των τιμών θα εμφανίζονται στη διεύθυνση URL.

Σημείωση: Η μέθοδος αυτή δεν πρέπει να χρησιμοποιείται κατά την αποστολή κωδικών πρόσβασης ή άλλων προσωπικών στοιχείων!

1.4.6.2 Η μεταβλητή \$_POST στην PHP

Η \$_POST μεταβλητή αυτή χρησιμοποιείται για τη συλλογή τιμών από μια φόρμα στέλνεται με τιμή: method = "post". Οι πληροφορίες που αποστέλλονται από τη φόρμα με τη μέθοδο POST δεν είναι ορατές για τους άλλους και δεν υπάρχουν όρια για την ποσότητα των πληροφοριών κατά την αποστολή. Σημείωση: Ωστόσο, υπάρχει ένα μέγιστο μέγεθος 8 Mb για τη μέθοδο POST, από προεπιλογή (μπορεί να αλλάξει με τον καθορισμό των post_max_size στο αρχείο php.ini).

Παράδειγμα:

```
<form action="welcome.php" method="post">
  Όνομα: <input type="text" name="fname" />
  Ηλικία: <input type="text" name="age" />
  <input type="submit" />
</ form>
```

Όταν ο χρήστης χρησιμοποιήσει το κουμπί "Submit", η διεύθυνση URL που αποστέλλεται στο διακομιστή θα είναι κάπως έτσι:

http:// welcome.php

Το "welcome.php" αρχείο μπορεί πλέον να χρησιμοποιεί τη \$_POST μεταβλητή για τη συλλογή δεδομένων στη φόρμα (τα ονόματα των πεδίων της φόρμας θα γίνονται αυτόματα τα κλειδιά στη συστοιχία \$_POST)

```
Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.
```

1.5 JavaScript

Η JavaScript είναι μια γλώσσα σε μια μορφή script που χρησιμοποιείται για να επιτρέψει την πρόσβαση μέσω προγραμματισμού σε αντικείμενα τόσο στην εφαρμογή-πελάτη όσο και άλλες εφαρμογές. Χρησιμοποιείται κυρίως με τη μορφή client-side JavaScript, ενσωματωμένη ως ένα ολοκληρωμένο στοιχείο του web browser, επιτρέποντας έτσι την ανάπτυξη ενισχυμένων διεπαφών χρήστη και δυναμικών ιστοσελίδων. Η JavaScript είναι μια διάλεκτος του προτύπου ECMAScript 10 και χαρακτηρίζεται ως δυναμική και ασθενώς δακτυλογραφημένη. Η JavaScript επηρεάστηκε από πολλές γλώσσες και έχει σχεδιαστεί να μοιάζει με τη Java, αλλά να είναι ευκολότερο για τους μη προγραμματιστές να εργαστούν με αυτή. [11]

1.5.1 Χαρακτηριστικά της JavaScript

Τα ακόλουθα χαρακτηριστικά είναι κοινά σε όλες τις εφαρμογές σύμφωνα με το ECMAScript, εκτός και αν ορίζεται διαφορετικά.

1.5.1.1 Αναγκαστική και δομημένη

Η JavaScript υποστηρίζει όλη τη δομημένη σύνταξη προγραμματισμού σε C (π.χ., δηλώσεις if, while βρόχοι, δηλώσεις switch, κλπ.), με μία μερική εξαίρεση την οριοθέτηση των πεδίων: πχ. το C-block style-scoring επίπεδο δεν υποστηρίζεται (αντ' αυτού, η JavaScript έχει ένα functional-scoring επίπεδο). Η JavaScript 1.7, ωστόσο, υποστηρίζει το block-επίπεδο scoring χρησιμοποιώντας την λέξη let. Όπως και η C, έτσι και η JavaScript κάνει διάκριση μεταξύ των εκφράσεων(expressions) και των δηλώσεων(statements).

1.5.1.2 Δυναμική

Δυναμική δακτυλογράφηση

Όπως στις περισσότερες γλώσσες προγραμματισμού, οι τύποι σχετίζονται με τις αξίες, και όχι με τις μεταβλητές αυτές καθ' αυτές. Για παράδειγμα, μια μεταβλητή x θα μπορούσε να δεσμεύεται σε έναν αριθμό, και αργότερα σε μια λέξη. Η JavaScript υποστηρίζει διάφορους τρόπους για τη δοκιμή του τύπου ενός αντικειμένου.

Βασισμένη στο αντικείμενο

Η JavaScript είναι σχεδόν εξ ολοκλήρου βασισμένη στο αντικείμενο. Τα αντικείμενα της JavaScript είναι συστοιχίες, επαυξημένες με τα πρωτότυπα. Τα ονόματα των ιδιοτήτων των αντικειμένων είναι λέξεις κλειδιά : Για παράδειγμα το `obj.x = 10` και `obj ["x"] = 10` είναι ισοδύναμα, με την τελεία στη σύνταξη να χρησιμοποιείται για να διαβάζεται ή να εκφράζεται ευκολότερα το αντικείμενο. Οι ιδιότητες και οι τιμές των αντικειμένων, μπορούν να προστεθούν, μεταβληθούν ή και να διαγραφούν κατά το χρόνο εκτέλεσης καθώς οι περισσότερες ιδιότητες ενός αντικειμένου (και εκείνων που αφορούν την πρωτότυπη αλυσίδα κληρονομίας) μπορούν να καταμετρηθούν με τη χρήση ενός βρόχου `for`. Η JavaScript έχει ένα μικρό αριθμό ενσωματωμένων αντικειμένων όπως το `Function` και το `Date`.

Αξιολόγηση του χρόνου εκτέλεσης

Η JavaScript περιλαμβάνει τη λειτουργία `eval` που μπορεί να εκτελέσει τα `statements` που έχουν δοθεί σαν `strings` κατά το χρόνο εκτέλεσης.

1.5.1.3 Λειτουργίες

Πρώτης κλάσης λειτουργίες

Τα `functions` ανήκουν στην πρώτη κλάση και υπάγονται στη κατηγορία των αντικειμένων και ως εκ τούτου, έχουν ιδιότητες και μπορούν να χρησιμοποιηθούν όπως κάθε άλλο αντικείμενο.

Εσωτερικές λειτουργίες(functions) και closures

Τα εσωτερικά `functions` (`functions` που ορίζονται σε άλλα `functions`) δημιουργούνται κάθε φορά που γίνεται επίκληση στην εξωτερική `function`, και οι μεταβλητές των εξωτερικών `functions` εξακολουθούν να υφίστανται στο βαθμό που οι εσωτερικές `functions` εξακολουθούν να υπάρχουν, ακόμη και μετά την ολοκλήρωση της επίκλησης (π.χ. αν η εσωτερική `function` επιστράφηκε, εξακολουθεί να έχει πρόσβαση στις μεταβλητές της εξωτερικής `function`). Αυτός είναι ο μηχανισμός πίσω από τα `closures` εντός της JavaScript.

1.5.1.4 Βασισμένη στα πρωτότυπα

Πρωτότυπα

Η JavaScript χρησιμοποιεί πρωτότυπα αντί των κλάσεων κληρονομικότητας. Είναι δυνατό να προσομοιωθούν πολλά χαρακτηριστικά που χρησιμοποιούν ως βάση τις κλάσεις χρησιμοποιώντας τα πρωτότυπα στη JavaScript.

Λειτουργίες(functions) ως κατασκευαστές αντικειμένων

Οι λειτουργίες διπλασιάζονται σαν κατασκευαστές αντικειμένων μαζί με το τυπικό ρόλο τους. Προκαθορίζοντας μια κλήση συνάρτησης με τη λέξη `new` δημιουργείτε ένα νέο αντικείμενο και οι κλήσεις του λειτουργούν με την τοπική λέξη-κλειδί `this` δεσμεύοντας έτσι αυτό το αντικείμενο για την εν λόγω επίκληση. Η ιδιότητα `prototype` του κατασκευαστή καθορίζει το αντικείμενο που χρησιμοποιείται για το εσωτερικό πρωτότυπο ενός νέου αντικειμένου. Οι ενσωματωμένοι κατασκευαστές που υπάρχουν στην JavaScript, όπως για παράδειγμα το `Array`, έχουν επίσης πρωτότυπα τα οποία μπορούν να τροποποιηθούν.

Λειτουργίες ως μέθοδοι

Σε αντίθεση με πολλές αντικειμενοστραφείς γλώσσες, δεν υπάρχει διάκριση μεταξύ του ορισμού της λειτουργίας και του ορισμού της μεθόδου. Αντίθετα, η διάκριση γίνεται κατά τη διάρκεια της λειτουργίας, έτσι μια `function` μπορεί να κληθεί σαν να ήταν μια μέθοδος. Όταν μια `function` καλείται σαν μέθοδος ενός αντικειμένου, η λειτουργία της τοπικής λέξης `this` είναι υποχρεωτική για την εν λόγω επίκληση στο αντικείμενο.

1.5.1.5 Διάφορα

Περιβάλλον run-time

Η JavaScript συνήθως βασίζεται στο χρόνο εκτέλεσης του περιβάλλοντος (π.χ. σε ένα `web browser`) για να παρέχει αντικείμενα και μεθόδους μέσω των οποίων τα οποία `scripts` μπορούν να αλληλεπιδράσουν με "τον έξω κόσμο". Στην πραγματικότητα, εναπόκειται από το περιβάλλον, έτσι ώστε να παρέχετε η δυνατότητα να συμπεριλάβει / `scripts` εισαγωγής (π.χ. HTML `<script>` στοιχεία).

Λειτουργίες variadic

Ένας αόριστος αριθμός παραμέτρων μπορεί να περάσει σε μια λειτουργία. Η λειτουργία αυτή μπορεί να έχει πρόσβαση σε αυτά μέσω των επίσημων παραμέτρων και των αντικειμένων τοπικών επιχειρημάτων.

Πίνακες και αντικείμενα literals

Όπως πολλές γλώσσες προγραμματισμού, έτσι και εδώ ,οι πίνακες και τα αντικείμενα (`associative arrays` σε άλλες γλώσσες), μπορούν να δημιουργηθούν το καθένα με μια συνοπτική συντόμευση της σύνταξης. Στην πραγματικότητα, αυτές οι `literals` αποτελούν τη βάση για τη μορφή των δεδομένων JSON.

Σύνηθες εκφράσεις

Η JavaScript υποστηρίζει επίσης, σύνηθες εκφράσεις με παρόμοιο τρόπο με τη Perl, οι οποίες παρέχουν μια πιο συνοπτική και ισχυρή σύνταξη για τη χειραγώγηση του κειμένου που είναι πιο εξελιγμένη από ό, τι οι ενσωματωμένες συναρτήσεις συμβολοσειράς.

1.5.2 Χρήση της JavaScript στις ιστοσελίδες

Η κύρια χρήση της JavaScript είναι να γράφει λειτουργίες που είναι ενσωματωμένες ή περιλαμβάνονται στις HTML σελίδες και αλληλεπιδρούν με το Document Object Model (DOM) της σελίδας. Μερικά απλά παραδείγματα αυτής της χρήσης είναι:

- Το άνοιγμα ενός νέου παραθύρου με προγραμματιστικό έλεγχο του μεγέθους, της θέσης και των χαρακτηριστικών του νέου παραθύρου (δηλαδή εάν το μενού, η γραμμών εργαλείων, κλπ. θα είναι ορατά).
- Επικύρωση των τιμών μιας διαδικτυακής φόρμας για τη βεβαίωση ότι οι τιμές αυτές θα γίνουν δεκτές προτού αυτές υποβληθούν στο διακομιστή.
- Αλλαγή εικόνων καθώς ο κέρσορας του ποντικιού κινείται από πάνω τους: Η τεχνική αυτή χρησιμοποιείται συχνά για να επιστήσει την προσοχή του χρήστη σε σημαντικές συνδέσεις οι οποίες εμφανίζονται ως γραφικά στοιχεία.

Επειδή ο κώδικας της JavaScript μπορεί να λειτουργεί τοπικά σε ένα πρόγραμμα περιήγησης του χρήστη (και όχι σε ένα απομακρυσμένο server) μπορεί να ανταποκρίνεται άμεσα στις ενέργειες χρηστών, κάνοντας τις εφαρμογές πιο διαδραστικές. Επιπλέον, ο κώδικας της JavaScript μπορεί να ανιχνεύσει ενέργειες χρηστών που η HTML δεν μπορεί να ανιχνεύσει από μόνη της, όπως παραδείγματος χάριν ατομικές πληκτρολογήσεις. Εφαρμογές, όπως το Gmail επωφελούνται από αυτό αφού ένα μεγάλο μέρος της λογικής της διεπαφής του χρήστη είναι γραμμένο σε JavaScript, και η JavaScript αποστέλλει αιτήσεις για παροχή πληροφοριών (όπως το περιεχόμενο ενός μηνύματος ηλεκτρονικού ταχυδρομείου) στο διακομιστή. Η ευρύτερη τάση του προγραμματισμού Ajax εκμεταλλεύεται με τον ίδιο τρόπο αυτή την δύναμη.

Ένας μηχανισμός της JavaScript (επίσης γνωστός και ως διερμηνέας της JavaScript ή JavaScript implementation) είναι ένας διερμηνέας που μεταφράζει τον JavaScript κώδικα και τον εκτελεί αναλόγως. Η πρώτη μηχανή JavaScript δημιουργήθηκε από Brendan Eich στο Netscape Communications Corporation, για την εφαρμογή περιήγησης Netscape Navigator web.

Ένα πρόγραμμα περιήγησης στο διαδίκτυο είναι μακράν το πιο κοινό περιβάλλον υποδοχής για τη JavaScript. Τα προγράμματα περιήγησης στο διαδίκτυο χρησιμοποιούν συνήθως API για τη δημιουργία "αντικειμένων υποδοχής" τα οποία είναι υπεύθυνα για την αντανάκλαση του DOM σε JavaScript. Ο web server είναι μια άλλη κοινή εφαρμογή του μηχανισμού της JavaScript. Ένας JavaScript web Server εκθέτει τα αντικείμενα εκείνα που αντιπροσωπεύουν μια αίτηση HTTP καθώς και αντικείμενα απάντησης, τα οποία ένα πρόγραμμα JavaScript στη συνέχεια χειραγωγεί, δημιουργώντας μια δυναμική ιστοσελίδα. [9]

Ένα παράδειγμα μιας ιστοσελίδας που περιέχει JavaScript είναι το παρακάτω:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head><title>simple page</title></head>
<body>
  <script type="text/javascript">
    document.write('Hello World!');
  </script>
  <noscript>
    <p>Your browser either does not support JavaScript, or you have JavaScript turned off.</p>
  </noscript>
</body>
</html>
```

1.6 jQuery



Η jQuery είναι μια ελαφριά, cross-browser JavaScript βιβλιοθήκη που δίνει έμφαση στην αλληλεπίδραση μεταξύ JavaScript και HTML. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον John Resig και πλέον χρησιμοποιείται σε περισσότερο από το 27% των 10.000 δημοφιλέστερων διαδικτυακών τόπων – καθιστώντας την ως την πιο δημοφιλή βιβλιοθήκη JavaScript σήμερα.

Η βιβλιοθήκη jQuery είναι δωρεάν, ανοιχτού κώδικα λογισμικό υπό διπλή άδεια χρήσης MIT License και GNU General Public License, Version 2. Η σύνταξη της έχει ως στόχο να την εύκολη πλοήγηση κόμβων ενός εγγράφου, επιλέγοντας στοιχεία του DOM, δημιουργώντας animations, διαχειρίζοντας γεγονότα (events), και αναπτύσσοντας διαδικτυακές εφαρμογές AJAX. Η jQuery παρέχει επίσης την δυνατότητα στους προγραμματιστές να δημιουργήσουν plugins στην ίδια την βιβλιοθήκη, με τέτοιο τρόπο ώστε να την επεκτείνουν. Με την παροχή αυτής της δυνατότητας, οι προγραμματιστές μπορούν να δημιουργήσουν χαμηλού επιπέδου αλληλεπιδράσεις και animations, προχωρημένα εφέ και υψηλού επιπέδου widgets. Το παραπάνω συμβάλει δραστικά στην δημιουργία ισχυρών και δυναμικών ιστοσελίδων στα πρότυπα του Web 2.0. [24]

1.6.1 Τα πλεονεκτήματα της JQuery

- Ακολουθεί την αρχή KISS (Keep It Simple Stupid): Η βιβλιοθήκη JQuery προσπαθεί να υπεραπλουστεύσει τον προγραμματισμό σε Javascript προσφέροντας πραγματικά απλούς μηχανισμούς και εντολές μέσω του framework της.
- Παρέχει πλήρη και αναλυτικότερη τεκμηρίωση που συμπληρώνεται από την εκτεταμένη παρουσία ηλεκτρονικών βοηθημάτων. Εκτός από την πολύ καλοδουλεμένη τεκμηρίωσή του JQuery, οι ενδιαφερόμενοι μπορούν να ανατρέξουν και στην σελίδα του Visual JQuery όπου μπορούν να βρουν μία εναλλακτική αλλά πολύ βολική, από άποψη δομής, τεκμηρίωση.
- Υποστηρίζεται από μία πάρα πολύ ενεργή κοινότητα: Όπως για τα περισσότερα open source έργα λογισμικού, έτσι και για το JQuery η ύπαρξη μιας κατά το μέγιστο δυνατό ενεργής κοινότητας αποτελεί τον ακρογωνιαίο λίθο για την ανάπτυξη και ευημερία του.
- Μικρό μέγεθος: Το γεγονός ότι το βασικό πακέτο της JQuery είναι μόλις 20Kb αφενός επιβεβαιώνει την πρώτη παρατήρηση, ότι δηλαδή η φιλοσοφία της έγκειται στην απλότητα και αφετέρου κάνει πολύ εύκολη την κατανόηση της αρχιτεκτονικής της.

- Ποικιλία χαρακτηριστικών: Η JQuery δίνει τη δυνατότητα στον χρήστη να χρησιμοποιήσει σχεδόν το σύνολο των δυνατοτήτων που προσφέρει η γλώσσα JavaScript. Από απλά χαρακτηριστικά που σχετίζονται με βασικές λειτουργίες εμφάνισης / απόκρυψης, ως Ajax κλήσεις και σύνθετα εφέ.
- Επεκτασιμότητα: Η λογική με την οποία είναι φτιαγμένη η JQuery είναι απλή πράγμα που αντικατοπτρίζεται και στον ίδιο της τον κώδικα. Αυτό κάνει πολύ εύκολη την επέκτασή / τροποποίησή της.

1.6.2 Χρήση

Για να ενσωματώσει κανείς την βιβλιοθήκη αυτή στην ιστοσελίδα του, αρκεί μόνο να κατεβάσει την βιβλιοθήκη και να συνδέσει το κεντρικό script με το έγγραφο HTML, γράφοντας την παρακάτω γραμμή στον κώδικα του εγγράφου HTML:

```
<script type="text/javascript" src="jQuery.js"></script>
```

Το παραπάνω μπορεί επίσης να ενσωματωθεί και μέσω του Google AJAX Libraries API:

```
<script type="text/javascript" src="http://www.google.com/jsapi"></script>  
<script>google.load("jquery", "1.3.2");</script>
```

Μόλις η βιβλιοθήκη jQuery ενσωματωθεί στο HTML έγγραφο, μπορούμε να χρησιμοποιήσουμε τις λειτουργίες της. Η jQuery έχει δύο είδη βασικών αλληλεπιδράσεων. Το πρώτο είδος είναι μέσω της ειδικής συνάρτησης \$, η οποία είναι ο κατασκευαστής ενός αντικείμενου τύπου jQuery. Αυτές οι συναρτήσεις, συχνά αποκαλούνται “εντολές”, και είναι chainable – επιστρέφουν πάντα αντικείμενο jQuery. Το δεύτερο είδος είναι μέσω των συναρτήσεων του στυλ \$.-πρόθεμα. Αυτές οι συναρτήσεις αποκαλούνται βοηθητικές λειτουργίες και δεν λειτουργούν με το jQuery αντικείμενο καθεαυτό.

Μια τυπική ροή εργασίας για επεξεργασία πολλαπλών κόμβων του DOM, αρχίζει με την συνάρτηση \$ η οποία καλείται με κάποιον CSS selector, που έχει ως αποτέλεσμα ένα jQuery αντικείμενο το οποίο παραπέμπει σε κανένα ή και πολλά αντικείμενα DOM του HTML εγγράφου. Αυτά τα αντικείμενα ή κόμβοι, μπορούν να επεξεργαστούν και να παραποιηθούν, μέσω των συναρτήσεων και των μεθόδων της βιβλιοθήκης. Για παράδειγμα:

```
$("div.test").add("p.quote").addClass("blue").slideDown("slow");
```

... βρίσκει την ένωση όλων των div με κλάση test και όλες της παραγράφους με κλάση quote, και τους προσθέτει στην κλάση τους την κλάση blue, όπου στην συνέχεια τα εμφανίζει την παραπάνω ένωση με ειδικό εφέ τύπου slide-down.

1.7 CSS

1.7.1 Ανάλυση των CSS

Τα Cascading Style Sheets ή αλλιώς εν συντομογραφία CSS είναι μια γλώσσα μορφοποίησης των φύλλων, που χρησιμοποιείται για να περιγράψει τη σημασιολογική παρουσίαση (δηλαδή, την εμφάνιση και μορφοποίηση) από ένα έγγραφο γραμμένο σε μια markup language. Η πιο κοινή εφαρμογή της είναι η μορφοποίηση ιστοσελίδων γραμμένες σε HTML και XHTML, αλλά η γλώσσα μπορεί να εφαρμοστεί σε οποιοδήποτε είδος εγγράφου XML.

Η CSS είναι κυρίως σχεδιασμένη ώστε να επιτρέπει το διαχωρισμό του περιεχομένου του εγγράφου (γραμμένο σε HTML ή παρόμοια markup language) από την παρουσίαση του εγγράφου, συμπεριλαμβανομένων των στοιχείων όπως, το χρώμα, τις γραμματοσειρές, και τη διάταξη. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να παρέχει περισσότερη ευελιξία και έλεγχο στη συγγραφή των χαρακτηριστικών παρουσίασης έτσι ώστε πολλές σελίδες να μπορούν να μοιράζονται μια ενιαία μορφοποίηση, και να μειώσει την πολυπλοκότητα και την επανάληψη του διαρθρωτικού περιεχομένου. Η CSS μπορεί επίσης να επιτρέψει στην ίδια σελίδα σήμανσης να παρουσιάζεται σε διαφορετικά στυλ για τις διαφορετικές μεθόδους επεξεργασίας, όπως στην οθόνη, στην εκτύπωση, με φωνή (όταν διαβάζεται από μια ομιλία με βάση το πρόγραμμα περιήγησης ή ανάγνωσης οθόνης). Αν και ο συντάκτης του εγγράφου συνδέει τυπικά το έγγραφο με ένα φύλλο στυλ CSS, ο αναγνώστης μπορεί να χρησιμοποιήσει ένα διαφορετικό στυλ, για παράδειγμα στον υπολογιστή του, παρακάμπτοντας έτσι το στυλ που ο συγγραφέας έχει καθορίσει.

Η CSS έχει προκαθορίσει ένα σύστημα προτεραιοτήτων, που καθορίζει πιο στυλ υπερισχύει εάν περισσότερα από ένα ανταγωνίζονται ένα συγκεκριμένο στοιχείο. Σε αυτό το λεγόμενο cascade, οι προτεραιότητες και οι βαρύτητες υπολογίζονται και αποδίδονται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα.

Οι προδιαγραφές των CSS τηρούνται από το World Wide Web Consortium (W3C) ενώ ο τύπος των μέσων ενημέρωσης (τύπος MIME) text / css έχει καταχωρηθεί για χρήση με τα CSS από το RFC 2318 (Μάρτιος 1998). [6] [9]

1.7.2 Σύνταξη των CSS

Η CSS έχει απλή σύνταξη και χρησιμοποιεί μια σειρά αγγλικών λέξεων-κλειδιά για να καθορίσει τις ονομασίες των διαφόρων ιδιοτήτων των στυλ.

Ένα φύλλο στυλ αποτελείται από μια λίστα των κανόνων, όπου κάθε κανόνας ή σεντ κανόνων αποτελείται από έναν ή περισσότερους επιλογείς και μια μπλοκ δήλωση. Μια μπλοκ δήλωση αποτελείται από μια λίστα των δηλώσεων και κάθε

δήλωση με τη σειρά της αποτελείται από ένα property, μια άνω και κάτω τελεία (:), μια τιμή, και μετά μια άνω τελεία (;).

Στα CSS, οι επιλογείς (selectors) χρησιμοποιούνται για να δηλώσουν σε ποια από τα στοιχεία markup εφαρμόζεται ένα στυλ, σαν ένα είδος έκφρασης συσχέτισης. Οι επιλογείς μπορούν να ισχύουν για όλα τα στοιχεία ενός συγκεκριμένου τύπου, ή μόνο για εκείνα τα στοιχεία που εμφανίζουν ένα συγκεκριμένο χαρακτηριστικό. Τα στοιχεία μπορούν να συνοδεύονται, ανάλογα με τον τρόπο που τοποθετούνται το ένα σε σχέση με το άλλο στον κώδικα σήμανσης, ή ανάλογα τον τρόπο που βρίσκονται εντός του μοντέλου αντικειμένου εγγράφου (document object model).

Οι ψευδο-κλάσεις είναι μια άλλη μορφή προδιαγραφών που χρησιμοποιούνται στα CSS για να προσδιοριστούν τα στοιχεία σήμανσης, καθώς και σε ορισμένες περιπτώσεις, συγκεκριμένες ενέργειες του χρήστη, ισχύει ένα συγκεκριμένο μπλοκ δήλωσης. Ένα συχνό παράδειγμα που χρησιμοποιείται η ψευδο-κλάση είναι η `:hover` η οποία εφαρμόζει ένα στυλ μόνο όταν ο χρήστης «δείχνει» στο ορατό στοιχείο, συνήθως κρατώντας τον κέρσορα του ποντικιού πάνω από αυτό. Άλλες ψευδο-κλάσεις και ψευδο-στοιχεία είναι, για παράδειγμα, το `:first-line`, το `:visited`, το `:before` κτλ.

Μια ψευδο-κλάση επιλέγει ολόκληρα στοιχεία, όπως το `:link` ή το `:visited`, λαμβάνοντας υπόψιν της ότι ένα ψευδο-στοιχείο κάνει μια επιλογή που μπορεί να αποτελείται από επιμέρους στοιχεία, όπως `:first-line` ή `:first-letter`.

Οι επιλογείς μπορούν να συνδυάζονται και με άλλους τρόπους, ιδίως στη CSS 2.1, για να επιτευχθεί μεγαλύτερη ακρίβεια και ευελιξία.

Παρακάτω φαίνεται ένα παράδειγμα το οποίο αθροίζει όλους τους κανόνες που αναφέραμε παραπάνω:

```
selector [, selector2, ...]:pseudo-class {  
  
property: value;  
  
}  
  
/* comment*/
```

1.7.3 Χρήση των CSS

Πριν από τα CSS, σχεδόν όλα τα χαρακτηριστικά παρουσίασης των HTML εγγράφων περιλαμβάνονταν μέσα στο έγγραφο σήμανσης HTML, όπως τα χρώματα του φόντου, τα στυλ του φόντου, οι ευθυγραμμίσεις των στοιχείων, ενώ τα σύνορα και τα μεγέθη έπρεπε να περιγράφονται ρητά, και συχνά κατ'επανάληψη. Με τη χρήση των CSS επιτρέπετε στους συγγραφείς να μεταφέρουν

πολλές από αυτές τις πληροφορίες σε ένα χωριστό φύλλο στυλ με αποτέλεσμα η HTML σήμανση να γίνεται πολύ απλούστερη.

Πριν από τα CSS, οι συγγραφείς που ήθελαν να δώσουν τέτοια τυπογραφικά χαρακτηριστικά στο έγγραφο, για παράδειγμα, όλα τα headings h2 έπρεπε να χρησιμοποιήσουν τη γραμματοσειρά HTML και άλλα στοιχεία παρουσίασης, έπρεπε να το κάνουν για κάθε εμφάνιση του heading h2. Έτσι η πρόσθετη σήμανση παρουσίασης στα έγγραφα HTML, τα έκανε πιο πολύπλοκα, και γενικά πιο δύσκολο να διατηρηθούν. Στα CSS, η παρουσίαση χωρίζεται από τη δομή επιτρέποντας έτσι, στην εκτύπωση, το CSS να μπορεί να ορίσει το χρώμα, τη γραμματοσειρά, την στοίχιση του κειμένου, το μέγεθος, τα σύνορα, τις αποστάσεις, τη διάταξη και πολλά άλλα τυπογραφικά χαρακτηριστικά,. Ενώ μπορούν να το κάνουν αυτό ανεξάρτητα για την οθόνη και ανεξάρτητα για τα έντυπα. Το W3C θεωρεί ότι τα πλεονεκτήματα των CSS για τον καθορισμό όλων των πτυχών της παρουσίασης των σελίδων HTML είναι ανώτερη από άλλες μεθόδους παρουσίασης.

1.7.4 Πηγές πληροφοριών στα CSS

Οι πληροφορίες των CSS μπορούν να παρέχονται από διάφορες πηγές και μπορούν να συνδεθούν με το έγγραφο HTML είτε ως χωριστό έγγραφο είτε να ενσωματωθούν σε αυτό. Διαφορετικά στυλ μπορούν να εφαρμοστούν σε ένα έγγραφο HTML ανάλογα με τη συσκευή εξόδου που χρησιμοποιείται. Για παράδειγμα, η έκδοση για την οθόνη μπορεί να είναι αρκετά

διαφορετική από την έντυπη έκδοση, έτσι ώστε οι συντάκτες να μπορούν να προσαρμόσουν την παρουσίαση κατάλληλα για κάθε μέσο.

Οι προτεραιότητες για τις πηγές CSS (από την υψηλότερη στη χαμηλότερη) είναι οι ακόλουθες :

Author styles (είναι πληροφορίες που παρέχονται από το συγγραφέα της ιστοσελίδας), με τη μορφή

- Inline στυλ, δηλαδή πληροφορίες μέσα στο έγγραφο HTML, για το ύψος σε ένα μόνο στοιχείο, που καθορίζονται με το χαρακτηριστικό "style".
- Ενσωματωμένα στυλ, δηλαδή μπλοκ πληροφοριών για τα CSS στο εσωτερικό του εγγράφου HTML
- εξωτερικά φύλλα στυλ, δηλαδή ένα ξεχωριστό CSS-αρχείο που αναφέρεται στο έγγραφο HTML.

User Styles

- Αυτού του είδους τα στυλ είναι ένα τοπικό αρχείο CSS, το οποίο καθορίζεται από τον χρήστη χρησιμοποιώντας τις επιλογές στο web

browser, και ενεργεί ως παράκαμψη, που πρέπει να εφαρμόζεται σε όλα τα έγγραφα.

User Agent styles

- Είναι το φύλλο στυλ που ισχύει από τον χρήστη, π.χ. η προεπιλεγμένη παρουσίαση των στοιχείων του browser του χρήστη.

Το φύλλο στυλ με την υψηλότερη προτεραιότητα χρησιμοποιείται ώστε να εμφανίσει το περιεχόμενο. Δηλώσεις που δεν έχουν δηλωθεί ως πηγές υψηλής προτεραιότητας, θα παρακάμπτονται από πηγές χαμηλότερης προτεραιότητας, όπως π.χ. το στυλ πράκτορα χρήστη. Αυτή η διαδικασία ονομάζεται cascading.

Ένας από τους στόχους των CSS είναι, να επιτρέπει στους χρήστες να έχουν μεγαλύτερο βαθμό ελέγχου στην παρουσίαση. Για παράδειγμα εκείνοι που βρίσκουν τους κόκκινους πλάγιους τίτλους δύσκολο να διαβαστούν, μπορούν να εφαρμόσουν άλλα φύλλα στυλ στο έγγραφο. Έτσι ανάλογα με τον browser και το web site, ο χρήστης μπορεί να επιλέξει διάφορα φύλλα στυλ που παρέχονται από τους σχεδιαστές, ώστε να αποσύρει όλα τα προστιθέμενα στυλ και να προβάλει την ιστοσελίδα χρησιμοποιώντας το προεπιλεγμένο στυλ του browser ή μπορεί να εξουδετερώσει μόνο τους κόκκινους πλάγιους τίτλους της κλάσης χωρίς να τροποποιήσει τα άλλα χαρακτηριστικά της σελίδας.

Το αρχείο highlightheaders.css περιέχει:

```
h1 { color: white; background: orange !important; }  
h2 { color: white; background: green !important; }
```

Ένα τέτοιο αρχείο είναι αποθηκευμένο σε τοπικό επίπεδο και μπορεί να εφαρμοστεί σε περίπτωση που έχει οριστεί στις επιλογές περιήγησης. Η λέξη "important" στο τέλος σημαίνει ότι το στυλ αυτό υπερισχύει από τα ορίσματα του συγγραφέα.

1.8 XML

Η XML (eXtensible Markup Language) είναι ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων με ηλεκτρονικά μέσα. Ορίζεται στην προδιαγραφή XML 1.0 που παράγεται από το W3C και πολλές άλλες σχετικές προδιαγραφές.

Οι σχεδιαστικοί στόχοι της XML είναι να τονίσει την απλότητα, τη γενικότητα, και την ευχρηστία της μέσω του διαδικτύου. Πρόκειται για μια μορφή δεδομένων κειμένου, με ισχυρή υποστήριξη μέσω του Unicode για τις διάφορες γλώσσες ανά τον κόσμο. Αν και ο σχεδιασμός της XML επικεντρώνεται σε έγγραφα, χρησιμοποιείται ευρέως για την αναπαράσταση των αυθαίρετων δομών δεδομένων, όπως για παράδειγμα στον τομέα των υπηρεσιών web.

Υπάρχει μια ποικιλία προγραμμάτων που χρησιμοποιούν οι προγραμματιστές λογισμικού ώστε να μπορούν να χρησιμοποιήσουν την XML για πρόσβαση στα δεδομένα, και πολλά συστήματα σχημάτων με στόχο να βοηθήσουν στον ορισμό των γλωσσών που βασίζονται στη XML.

Από το 2009 και μετά, έχουν αναπτυχθεί πολλές γλώσσες που βασίζονται σε XML, συμπεριλαμβανομένου σε αυτές την RSS, την Atom, την SOAP, και την XHTML. Η XML έχει γίνει η προεπιλεγμένη μορφή αρχείου για τα περισσότερα εργαλεία γραφείου παραγωγικότητας, συμπεριλαμβανομένου του Microsoft Office, OpenOffice.org, AbiWord, και iWork της Apple. [6]

1.8.1 Ανάλυση των χαρακτηριστικών της XML

Τα παρακάτω χαρακτηριστικά βασίζονται στις προδιαγραφές της XML, χωρίς όμως να αποτελούν έναν πλήρη κατάλογο όλων των χαρακτηριστικών που εμφανίζονται στην XML αλλά παρέχετε μια εισαγωγή στα βασικά χαρακτηριστικά που συναντάμε συχνότερα.

Χαρακτήρες (Unicode)

Εξ ορισμού, ένα έγγραφο XML είναι μια συμβολοσειρά χαρακτήρων, στον οποίο σχεδόν κάθε νόμιμος χαρακτήρας Unicode, μπορεί να εμφανιστεί .

Επεξεργαστής και εφαρμογές

Αναμένεται ότι ένας επεξεργαστής λειτουργεί στην υπηρεσία της εφαρμογής που χειρίζεται XML. Υπάρχουν ορισμένες πολύ ειδικές απαιτήσεις σχετικά με το τι πρέπει να κάνει και το τι δεν πρέπει να κάνει ο επεξεργαστής XML, αλλά καμία ως το προς τη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως οι προδιαγραφές τον ονομάζουν) αναφέρεται συχνά στην καθομιλουμένη ως ένας XML parser.

Markup και περιεχόμενο

Οι χαρακτήρες που συνθέτουν ένα έγγραφο XML χωρίζονται στη σήμανση και στο περιεχόμενο. Η σήμανση και το περιεχόμενο μπορούν να διακρίνονται από την εφαρμογή μέσω απλών συντακτικών κανόνων. Όλες οι συμβολοσειρές που αποτελούν σήμανση αρχίζουν με το χαρακτήρα "<" και τελειώνουν με το χαρακτήρα ">", ή αρχίζουν με το χαρακτήρα "&" και τελειώνουν με τον χαρακτήρα ";" . Οι συμβολοσειρές χαρακτήρων που δεν αποτελούν μέρος του markup αποτελούν μέρος του περιεχόμενου.

Tag (Ετικέτες)

Είναι μια markup κατασκευή που ξεκινά με "<" και τελειώνει με ">". Οι ετικέτες εμφανίζονται σε τρεις μορφές: Ετικέτες έναρξης, για παράδειγμα <section>, Ετικέτες τέλους, για παράδειγμα </section>, και σε ετικέτες κενών στοιχείων, για παράδειγμα <line-break/>.

Στοιχεία

Είναι μια λογική συνιστώσα του εγγράφου η οποία αρχίζει είτε με την ίδρυση μιας ετικέτας και τελειώνει με μια ετικέτα τέλους, ή αποτελείται μόνο από μια ετικέτα κενών στοιχείων. Οι χαρακτήρες μεταξύ των σημείων των ετικετών έναρξης και λήξης, εάν υπάρχουν, είναι τα περιεχόμενα του στοιχείου, και μπορεί να περιέχουν σήμανση, συμπεριλαμβανομένων και των άλλων στοιχείων, τα οποία ονομάζονται child elements. Ένα παράδειγμα είναι ένα στοιχείο <Greeting> Hello, world. </Greeting>. Ένα άλλο είναι το <line-break/>.

Ιδιότητα

Μια markup κατασκευή αποτελείται από ένα ζεύγος όνομα / τιμή που υπάρχει μέσα σε μία ετικέτα έναρξης ή σε μια ετικέτα κενού στοιχείου. Σε αυτό το παράδειγμα, το όνομα της ιδιότητας είναι "number" και η τιμή της είναι "3": <step number="3"> Connect A to B. </ step> Το ακόλουθο στοιχείο έχει δύο ιδιότητες, src και alt: . Ένα στοιχείο δεν πρέπει να έχει δύο ιδιότητες με το ίδιο όνομα.

Δηλώσεις XML

Τα XML έγγραφα ξεκινούνε δηλώνοντας ορισμένες πληροφορίες για τον εαυτό τους, όπως στο ακόλουθο παράδειγμα.

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

Παράδειγμα

Εδώ είναι ένα μικρό, πλήρες έγγραφο XML, το οποίο χρησιμοποιεί όλες τις δομές και τις έννοιες.

```
<? xml version = "1.0" encoding = "utf-8"?>
<painting>
  
  <caption> This is Raphael's "Foligno" Madonna, painted in
  <date> 1511 </ date>-<date> 1512 </ date>. </caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το έγγραφο: painting, img, title, και δύο date. Τα στοιχεία date είναι παιδιά του caption, το οποίο είναι ένα παιδί του στοιχείου painting. Το img έχει δύο ιδιότητες, src και alt.

1.8.2 Χαρακτήρες και διαφυγή

Τα XML έγγραφα που αποτελούνται εξ ολοκλήρου από τους χαρακτήρες του Unicode. Εκτός από ένα μικρό αριθμό, ρητά αποκλεισμένων, χαρακτήρων ελέγχου κάθε χαρακτήρα που ορίζεται από το Unicode μπορεί να εμφανιστεί στο περιεχόμενο ενός εγγράφου XML. Η επιλογή των χαρακτήρων που μπορούν να εμφανιστούν εντός σήμανσης είναι κάπως πιο περιορισμένη αλλά εξακολουθεί να είναι μεγάλη.

Η XML περιλαμβάνει ιδιότητες για τον προσδιορισμό της κωδικοποίησης των χαρακτήρων Unicode που συνθέτουν το έγγραφο, και για την έκφραση των χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν άμεσα.

Κωδικοποίηση ανίχνευσης

Το σύνολο χαρακτήρων Unicode μπορούν να κωδικοποιηθούν σε bytes για την αποθήκευση ή τη μετάδοση με μια ποικιλία διαφορετικών τρόπων, που ονομάζεται "κωδικοποίηση". Το Unicode από μόνο του ορίζει κωδικοποιήσεις που καλύπτουν όλο το σύνολο, ποιο γνωστές από τις οποίες είναι το UTF-8 και το UTF-16. Υπάρχουν πολλές άλλες κωδικοποιήσεις κείμενου που υπήρχαν πριν από το Unicode, όπως ASCII και ISO / IEC 8859. Τα χαρακτηριστικά τους σχεδόν σε κάθε περίπτωση είναι υποσύνολα των χαρακτήρων Unicode.

Η XML επιτρέπει τη χρήση οποιασδήποτε από τις καθορισμένες Unicode κωδικοποιήσεις, καθώς και κάθε άλλη μορφή κωδικοποίησης χαρακτήρων των οποίων οι χαρακτήρες εμφανίζονται στο Unicode. Η XML επίσης παρέχει ένα μηχανισμό με τον οποίο ένας XML επεξεργαστής μπορεί με αξιοπιστία και χωρίς καμία προηγούμενη γνώση, να καθορίσει τη κωδικοποίηση που χρησιμοποιείται, χωρίς να είναι απαραίτητο ότι κωδικοποιήσεις πέραν της UTF-8 και της UTF-16 θα αναγνωριστούν από κάθε parser.

Χαρακτήρες Διαφυγής

Υπάρχουν διάφοροι λόγοι για τους οποίους μπορεί να είναι δύσκολο ή αδύνατο να συμπεριληφθούν ορισμένοι χαρακτήρες απευθείας σε ένα έγγραφο XML. Για παράδειγμα, οι χαρακτήρες "<" και "&" αποτελούν βασικούς δείκτες σύνταξης και δεν μπορούν να εμφανιστούν στο περιεχόμενο, (π.χ. μπορεί να κωδικοποιηθεί ένα έγγραφο XML σε ASCII, αλλά δεν η ASCII δεν μπορεί να υποστηρίξει πολλούς χαρακτήρες Unicode, όπως για παράδειγμα, το "έ"). Για τους λόγους αυτούς, η XML παρέχει χαρακτηριστικά διαφυγής για τις περιπτώσεις που υπάρχουν προβληματικοί ή ανύπαρκτοι χαρακτήρες. Υπάρχουν πέντε προκαθορισμένες οντότητες: το < το οποίο αναπαριστά τον χαρακτήρα "<", το > το οποίο αναπαριστά τον χαρακτήρα ">", το & το οποίο αναπαριστά τον χαρακτήρα "&", το ' το οποίο αναπαριστά τον χαρακτήρα "'", και το " το οποίο αναπαριστά τον χαρακτήρα "\". Επίσης, κάθε χαρακτήρας Unicode μπορεί να εκπροσωπείται με έναν αριθμητικό χαρακτήρα αναφοράς. Σκεφτείτε το κινεζικό χαρακτήρα "中", αριθμητικός κωδικός του οποίου σε μορφή Unicode είναι ο 4E2D σε δεκαεξαδική μορφή ή σε δεκαδική ο 20.013. Ένας χρήστης του οποίου το πληκτρολόγιο δεν προσφέρει καμία μέθοδο για την εισαγωγή αυτού χαρακτήρα, θα μπορούσε να τον εισάγει παρόλα αυτά σε ένα έγγραφο XML με την κωδικοποίηση του είτε ως 中 ή 中. Ομοίως, η συμβολοσειρά "I <3 Jörg" θα μπορούσε να είναι κωδικοποιημένη για συμπεριλαμβάνεται σε ένα έγγραφο XML ως "I <3 Jörg".

1.8.3 XMLHttpRequest

Το XMLHttpRequest (XHR) είναι ένα DOM API που μπορεί να χρησιμοποιηθεί μέσα σε μια web browser scripting language, όπως JavaScript, για να στείλει μία αίτηση HTTP ή HTTPS απευθείας σε έναν web server και να φορτώσει τα δεδομένα της απάντησης από τον server απευθείας πίσω σε scripting language. Από την στιγμή που τα δεδομένα θα βρίσκονται εντός του scripting language, είναι διαθέσιμα και ως ένα έγγραφο XML, αν η απάντηση ήταν έγκυρης σήμανσης της XML, αλλά και ως απλό κείμενο. Έτσι τα δεδομένα της XML μπορούν να χρησιμοποιηθούν ώστε να χειραγωγήσουν το ενεργό έγγραφο στο παράθυρο του προγράμματος περιήγησης χωρίς να είναι απαραίτητο να φορτωθεί εκ νέου το έγγραφο της ιστοσελίδας από τον χρήστη. Το απλό κείμενο δεδομένων επίσης μπορεί να αξιολογηθεί στο πλαίσιο του scripting language ώστε να χειραγωγήθει το έγγραφο. Στο παράδειγμα της JavaScript, το απλό κείμενο μπορεί να έχει διαμορφωθεί ως JSON από τον web server και να αξιολογηθεί εντός της JavaScript για να δημιουργήσει ένα αντικείμενο δεδομένων που θα χρησιμοποιηθεί στο τρέχων DOM έγγραφο.

Το αντικείμενο XMLHttpRequest έχει ένα αρκετά σημαντικό ρόλο στην τεχνική AJAX. Χρησιμοποιείται σήμερα από πολλές ιστοσελίδες για να εφαρμόζουν και να ανταποκρίνονται σε δυναμικές web εφαρμογές. Παραδείγματα αυτών των εφαρμογών web περιλαμβάνουν το Gmail, το Google Maps, οι Bing Χάρτες, η δυναμική διεπαφή χάρτη MapQuest, το Facebook, και άλλα.

1.8.3.1 Η αίτηση HTTP

Παρακάτω θα δούμε πώς γίνεται μια αίτηση χρησιμοποιώντας τις λειτουργίες του αντικείμενου XMLHttpRequest σε ένα πράκτορα χρήστη με βάση το σχέδιο εργασίας του W3C. Δεδομένου ότι το πρότυπο του W3C για το αντικείμενο XMLHttpRequest εξακολουθεί να είναι ένα σχέδιο, οι πράκτορες του χρήστη ίσως να μην μπορούν να τηρήσουν όλες τις λειτουργίες του ορισμού του W3C και οποιοδήποτε από τα παρακάτω υπόκεινται σε αλλαγές. Εξαιρετική προσοχή θα πρέπει να λαμβάνεται υπόψη κατά το scripting με το αντικείμενο XMLHttpRequest σε πολλούς πράκτορες χρήστη. Παρακάτω απαριθμούνται οι ασυμβατότητες μεταξύ των κυριότερων πρακτόρων χρήστη.

Η μέθοδος open

Τα HTTP και HTTPS αιτήματα του αντικειμένου XMLHttpRequest πρέπει να αρχικοποιηθούν μέσω της μεθόδου open. Η μέθοδος αυτή πρέπει να κληθεί πριν από την πραγματική αποστολή της αίτησης για να επικυρωθεί και να επιλυθεί η μέθοδος αίτησης, ενώ το URL και URI πληροφορίες του χρήστη θα χρησιμοποιηθούν για την αίτηση. Η μέθοδος αυτή δεν διαβεβαιώνει ότι η διεύθυνση URL υπάρχει ή ότι τα στοιχεία του χρήστη είναι σωστά. Αυτή η μέθοδος μπορεί να δεχτεί μέχρι και πέντε παραμέτρους, ωστόσο δύο μόνο, είναι αρκετοί για να αρχικοποιηθεί μια αίτηση.

Η πρώτη παράμετρος της μεθόδου είναι μια συμβολοσειρά κειμένου που αναφέρει τη μέθοδο- αίτησης HTTP που θα χρησιμοποιηθεί. Οι μέθοδοι αίτησης που πρέπει να υποστηρίζονται από έναν πράκτορα χρήστη, και έχουν οριστεί από το σχέδιο του W3C για το αντικείμενο XMLHttpRequest, περιλαμβάνονται προς το παρόν τα εξής:

- GET (Υποστηρίζεται από IE7 +, Mozilla 1 +)
- POST (Υποστηρίζεται από IE7 +, Mozilla 1 +)
- HEAD (Υποστηριζόμενο από IE7 +)
- PUT
- DELETE
- OPTIONS (Υποστηριζόμενο από IE7 +)

Ωστόσο, οι μέθοδοι αίτησης δεν περιορίζονται μόνο σε εκείνες που αναγράφονται παραπάνω. Το σχέδιο του W3C δηλώνει ότι ένα πρόγραμμα περιήγησης μπορεί να υποστηρίζει επιπρόσθετες μεθόδους αίτησης κατά την κρίση τους.

Η δεύτερη παράμετρος της μεθόδου είναι άλλη μια συμβολοσειρά κειμένου, που αναφέρει το URL της αίτησης HTTP. Το W3C συνιστά ότι οι περιηγητές θα πρέπει να αναφέρουν ένα λάθος και να μην επιτρέπουν την αίτηση του URL σε μια διαφορετική θύρα ή με ένα διαφορετικό ihost URI από ότι το τρέχον έγγραφο.

Στη τρίτη παράμετρο, υπάρχει μια boolean τιμή οι οποίοι συμβολίζει εάν η αίτηση θα είναι ή δεν θα είναι ασύγχρονη, βέβαια αυτή δεν είναι μια απαραίτητη παράμετρος που αναφέρεται στο σχέδιο του W3C. Η προεπιλεγμένη τιμή αυτής της παραμέτρου θα πρέπει να θεωρείται ότι είναι αληθής σύμφωνα με το W3C από κάθε πράκτορα χρήστη σε περίπτωση που αυτή δεν παρέχεται. Μια ασύγχρονη αίτηση ("true") δεν θα περιμένει για μια απάντηση από τον server πριν συνεχίσει κανονικά με την εκτέλεση του τρέχοντος script. Θα επικαλεστεί, αντ' αυτού το onreadystatechange event listener του αντικειμένου XMLHttpRequest κατά τα διάφορα στάδια της αίτησης. Αντίθετα μια σύγχρονη αίτηση ("false"), θα παγώσει την εκτέλεση του τρέχοντος script έως ότου η αίτηση ολοκληρωθεί, χωρίς να επικαλεστεί το onreadystatechange event listener.

Η τέταρτη και πέμπτη παράμετροι είναι τα URI, όνομα χρήστη και κωδικό πρόσβασης, αντίστοιχα. Αυτές οι παράμετροι, ή απλά το όνομα χρήστη, παρέχονται για τον έλεγχο της ταυτότητας του χρήστη, εφόσον αυτή απαιτείται από το διακομιστή για το συγκεκριμένο αίτημα.

Η μέθοδος setRequestHeader

Με την επιτυχή αρχικοποίηση μιας αίτησης, η μέθοδος setRequestHeader του αντικειμένου XMLHttpRequest μπορεί να χρησιμοποιηθεί για να αποστείλει κεφαλίδες HTTP μαζί με την αίτηση. Η πρώτη παράμετρος αυτής της μεθόδου είναι η συμβολοσειρά του ονόματος του κειμένου της κεφαλίδας και η δεύτερη παράμετρος είναι η συμβολοσειρά της τιμής του κειμένου. Στη μέθοδο αυτή πρέπει να γίνει επίκληση για κάθε κεφαλίδα που πρέπει να αποσταλεί με την αίτηση αλλιώς κάθε επιπλέον κεφαλίδα που είναι συνημμένη θα αφαιρείται την επόμενη φορά που η μέθοδος open θα επικαλεστεί σε ένα πράκτορα χρήστη.

Η μέθοδος αποστολής (send)

Για να σταλεί μια αίτηση HTTP, πρέπει να επικαλεστεί η μέθοδος αποστολής του XMLHttpRequest. Αυτή η μέθοδος δέχεται μία μόνο παράμετρο, που περιέχει το περιεχόμενο που θα αποσταλεί με την αίτηση, η οποία παράμετρος μπορεί να παραληφθεί εάν δεν υπάρχει κάποιο περιεχόμενο για να αποσταλεί. Το σχέδιο του W3C αναφέρει ότι η παράμετρος αυτή μπορεί να είναι οποιουδήποτε είδους που είναι στη διάθεση της scripting language, εφόσον αυτό μπορεί να μετατραπεί σε μια συμβολοσειρά κειμένου, με εξαίρεση το αντικείμενο εγγράφου DOM. Εάν ένας πράκτορας χρήστη δεν μπορεί να κάνει συμβολοσειρά την παράμετρο, τότε η παράμετρος αυτή θα αγνοείται.

Εάν η παράμετρος αποτελεί ένα αντικείμενο εγγράφου DOM, τότε ο πράκτορας του χρήστη θα πρέπει να διαβεβαιώσει ότι το έγγραφο έχει μετατραπεί σε καλά μορφοποιημένο XML το οποίο χρησιμοποιεί την κωδικοποίηση που υποδεικνύεται από το inputEncoding του αντικειμένου του εγγράφου. Εάν η κεφαλίδα αίτησης Content-Type δεν έχει προστεθεί μέσω του setRequestHeader, θα πρέπει αυτόματα να προστίθεται από τον πράκτορα του χρήστη ως

"application/xml;charset=charset" όπου το charset είναι η κωδικοποίηση που χρησιμοποιείται για την κωδικοποίηση του εγγράφου.

O onreadystatechange event listener

Εάν γίνει επίκληση της μεθόδου open του αντικειμένου XMLHttpRequest, με την τρίτη παράμετρο να ισχύει για ασύγχρονο αίτημα, θα γίνει αυτόματα επίκληση του onreadystatechange event listener για κάθε μία από τις ακόλουθες ενέργειες που αλλάζουν την ιδιότητα readyState του αντικειμένου XMLHttpRequest.

- Αφού έχει επικληθεί με επιτυχία η μέθοδος open, στην ιδιότητα του readyState του αντικειμένου XMLHttpRequest θα πρέπει να έχει αποδοθεί η τιμή 1.
- Αφού έχει επικληθεί με επιτυχία η μέθοδος send και η απάντηση των κεφαλίδων HTTP έχει ληφθεί, στην ιδιότητα readyState του αντικειμένου XMLHttpRequest θα πρέπει να έχει αποδοθεί η τιμή 2.
- Μόλις το περιεχόμενο της απάντησης HTTP αρχίζει να φορτώνετε, στην ιδιότητα readyState του αντικειμένου XMLHttpRequest θα πρέπει να έχει αποδοθεί η τιμή 3.
- Μόλις το φόρτωμα του περιεχομένου της απάντησης HTTP έχει ολοκληρωθεί, στην ιδιότητα readyState του αντικειμένου XMLHttpRequest θα πρέπει να έχει αποδοθεί η τιμή 4.

Η απάντηση HTTP

Μετά από μια επιτυχημένη και ολοκληρωμένη κλήση της μεθόδου send του XMLHttpRequest αντικειμένου, εάν η απάντηση από τον server ήταν ένα έγκυρο XML και η κεφαλίδα του content-type που αποστάληκε από το διακομιστή είναι κατανοητή από τον πράκτορα του χρήστη ως Internet media type για XML, η ιδιότητα responseXML του αντικειμένου XMLHttpRequest θα περιέχει ένα αντικείμενο εγγράφου DOM. Το responseText θα περιέχει την απάντηση του server σε μορφή απλού κειμένου από ένα σύμφωνο πράκτορα χρήστη, ανεξάρτητα από το αν ήταν ή δεν ήταν κατανοητό ως XML.

1.9 Java Servlets

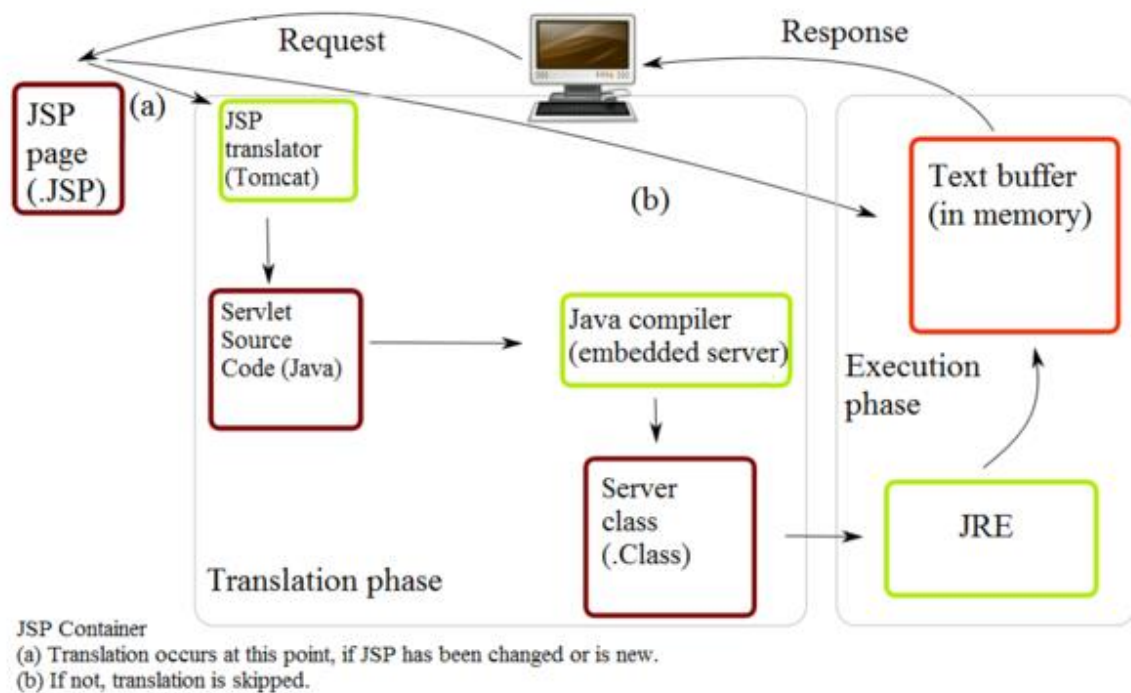
Τα Servlets είναι αντικείμενα της γλώσσας προγραμματισμού Java που επεξεργάζονται δυναμικά την αίτηση και την κατασκευή απάντησης. Η Java Servlet API επιτρέπει την ανάπτυξη λογισμικού ώστε να είναι δυνατόν να προσθεθεί περιεχόμενο σε ένα διακομιστή Web που χρησιμοποιεί την πλατφόρμα Java, δυναμικά. Το περιεχόμενο που παράγεται συνήθως είναι HTML, αλλά μπορεί να είναι άλλων μορφών, όπως XML. Τα Servlets είναι τα υποκατάστατα για τις μη δυναμικές τεχνολογίες περιεχομένου Java στον Παγκόσμιο Ιστό, όπως η CGI και ASP.NET. Τα Servlets μπορούν να διατηρήσουν την κατάσταση τους σε πολλές συναλλαγές διακομιστή χρησιμοποιώντας τα HTTP cookies, τις μεταβλητές περιόδου ή την επανεγγραφή του URL.

Το Servlet API, το οποίο περιέχεται στο πακέτο javax.servlet της Java, καθορίζει όλες τις αναμενόμενες αλληλεπιδράσεις του web container και του servlet. Ένας web container είναι ουσιαστικά το συστατικό του διακομιστή Web που αλληλεπιδρά με τα servlets και είναι υπεύθυνος για τη διαχείριση του κύκλου ζωής των servlets, τη χαρτογράφηση μιας διεύθυνσης URL σε ένα συγκεκριμένο servlet και την διασφάλιση ότι ο αιτών του URL έχει τα σωστά δικαιώματα πρόσβασης.

Ένα Servlet είναι ένα αντικείμενο που δέχεται μια αίτηση και δημιουργεί μια απάντηση που βασίζεται στην εν λόγω αίτηση. Το βασικό πακέτο του servlet ορίζει τα αντικείμενα της Java τα οποία εκπροσωπούν τις αιτήσεις των servlet αλλά και τις απαντήσεις, καθώς και αντικείμενα τα οποία αντανακλούν τις παραμέτρους διαμόρφωσης του servlet και το περιβάλλον εκτέλεσης τους. Το πακέτο javax.servlet.http ορίζει τις ειδικές HTTP υποκατηγορίες των γενικών στοιχείων των servlet, συμπεριλαμβανομένων και εκείνων των αντικειμένων διαχείρισης σύνδεσης που παρακολουθούν τις πολλαπλές αιτήσεις και απαντήσεις μεταξύ του Web server και ενός πελάτη. Τα Servlets μπορούν να συμπεριλαμβάνονται σε ένα αρχείο WAR ως μια εφαρμογή web.

Τα Servlets μπορούν να παράγονται αυτόματα από τα JavaServer Pages (JSP) compiler, ή εναλλακτικά από πρότυπους μηχανισμούς όπως WebMacro ή το apache velocity ώστε να παράγουν HTML. Συχνά τα Servlets χρησιμοποιούνται σε συνδυασμό με τα JSPs σε ένα μοτίβο που ονομάζεται "Model 2", το οποίο είναι ένα υποσύνολο του μοτίβου model-view-controller.

1.9.1 Κύκλος ζωής ενός Java Servlet



Εικόνα 1 «Κύκλος ζωής ενός Java Servlet»

Η διάρκεια του κύκλου ζωής ενός servlet αποτελείται από τα ακόλουθα βήματα:

1. Η κλάση servlet φορτώνεται από τον container κατά τη διάρκεια της εκκίνησης.
2. Ο container καλεί τη μέθοδο `init()` η οποία αρχικοποιεί το servlet και πρέπει να καλείται προτού το servlet μπορεί να είναι σε θέση να εξυπηρετήσει τις όποιες αιτήσεις, και καλείται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής του.
3. Μετά την αρχικοποίηση, το servlet μπορεί να εξυπηρετήσει τις αιτήσεις των πελατών. Κάθε αίτηση εξυπηρετείται με το δικό της ξεχωριστό αίτημα. Ο container καλεί την μέθοδο `service()` του servlet για κάθε αίτηση η οποία με την σειρά της προσδιορίζει το είδος του αιτήματος και το προωθεί στην κατάλληλη μέθοδο ώστε να το χειριστεί. Ο προγραμματιστής του servlet πρέπει να παρέχει την εφαρμογή αυτών των μεθόδων. Εάν πραγματοποιηθεί αίτημα για μια μέθοδο που δεν έχει εφαρμοστεί από το servlet, η μέθοδος της μητρικής κλάσης καλείται, οδηγώντας συνήθως σε σφάλμα που επιστρέφεται στον αιτούντα.
4. Τέλος, ο container καλεί τη μέθοδο `destroy()` που θέτει το servlet εκτός υπηρεσίας. Η μέθοδος `destroy()`, όπως και η μέθοδος `init()`, καλείται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής του servlet.

Παρακάτω παραθέτετε ένα απλό servlet που παράγει μόνο HTML. Το HttpServlet είναι μια υποκατηγορία των GenericServlet. Η μέθοδος service() αποστέλλει αιτήσεις στις μεθόδους doGet(), doPost(), doPut(), doDelete(), κ.λπ., ανάλογα με το αίτημα HTTP.

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \"
+\"Transitional//EN\">\n" +
"<html>\n" +
"<head><title>Hello WWW</title></head>\n" +
"<body>\n" +
"<h1>Hello WWW</h1>\n" +
"</body></html>");
    }
}
```

ServletConfig and ServletContext

Σε κάθε εφαρμογή υπάρχει μόνο ένα ServletContext. Αυτό το αντικείμενο μπορεί να χρησιμοποιηθεί από όλα τα servlets για να πάρει πληροφορίες από το επίπεδο της εφαρμογής ή τις λεπτομέρειες του container. Κάθε servlet, από την άλλη πλευρά, παίρνει το δικό του αντικείμενο ServletConfig το οποίο αναλαμβάνει να αρχικοποιήσει τις διάφορες παραμέτρους για ένα servlet. Οι προγραμματιστές μπορούν να λάβουν αναφορές για το ServletContext χρησιμοποιώντας το αντικείμενο ServletConfig.

1.9.2 Servlet Containers

Ένας servlet container είναι ένας εξειδικευμένος web server που υποστηρίζει την εκτέλεση του servlet. Συνδυάζει τις βασικές λειτουργίες ενός web server με συγκεκριμένες Java / servlet βελτιστοποιήσεις και επεκτάσεις, όπως ένα ολοκληρωμένο περιβάλλον χρόνου εκτέλεσης Java, καθώς και την ικανότητα να μεταφράζει αυτόματα συγκεκριμένες διευθύνσεις URL σε αιτήματα servlet. Μεμονωμένα servlets καταχωρούνται με servlet containers, παρέχοντας στο container τις πληροφορίες για το τι λειτουργίες θα παρέχουν, και ποια διεύθυνση URL ή ποιο άλλο εντοπιστή πόρων θα χρησιμοποιήσουν για να αναγνωρίσουν τη

ταυτότητα τους. Επίτα ο container του servlet είναι σε θέση να αρχικοποιήσει το servlet ανάλογα με τις ανάγκες και να παραδώσει τις αιτήσεις στο servlet, καθώς αυτές καταφθάνουν. Πολλοί containers έχουν την δυνατότητα να προσθέτουν και να αφαιρούν servlets δυναμικά από το σύστημα, επιτρέποντας έτσι στα νέα servlets να εγκαθίστανται ή να αναιρούνται γρήγορα χωρίς να επηρεάζουν τα άλλα servlets που δουλεύουν από τον ίδιο container. Τα Servlet containers αναφέρονται επίσης ως web containers ή web engines.

Όπως και τα άλλα Java API, διαφορετικά vendors παρέχουν τις δικές τους υλοποιήσεις για το πρότυπο του servlet container. Μια λίστα με μερικά από τα ελεύθερα και web containers παρέχεται παρακάτω (Με τον όρο «ελεύθερα» σημαίνει ότι η μη εμπορική χρήση είναι δωρεάν, ενώ ορισμένα από τα εμπορικά containers, όπως το Resin και το Orion, είναι ελεύθερά στο να χρησιμοποιούνται σε ένα περιβάλλον server από μη κερδοσκοπικούς οργανισμούς):

- Apache Tomcat (πρώην Jakarta Tomcat) είναι ένας web container ανοικτής πηγής διαθέσιμος στα πλαίσια του Apache Software License.
- Apache Geronimo είναι μια πλήρης εφαρμογή Java EE από την Apache.
- GlassFish (ανοιχτής πηγής κώδικας), από την Sun Microsystems.
- Jetty
- Jaminid περιέχει μεγαλύτερη άντληση από servlets.
- Enhydra
- Winstone στηρίζει προδιαγραφή v2.4, επικεντρώνετε στην ελάχιστη διαμόρφωση και στην ικανότητα να απλοποιεί τον container και να τον αφήνει μόνο με όσα χρειάζεται.
- Tiny Java Web Server (TJWS).

1.10 JSON



Η γλώσσα JSON (JavaScript Object Notation) είναι μια σημασιολογική γλώσσα σε μορφή κειμένου, παρόμοια με την XML, η οποία χρησιμοποιείται ως μια εναλλακτική μορφή αναπαράστασης απλών δομών δεδομένων και συστοιχιών που ονομάζονται αντικείμενα.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ],
  "newSubscription": false,
  "companyName": null
}
```

Εικόνα 2 «Παράδειγμα κώδικα Json»

Η μορφή JSON χρησιμοποιείται συχνά για το serialization και τη μεταβίβαση διαρθρωμένων δεδομένων (structured data) μέσω μιας σύνδεσης στο δίκτυο. Η κύρια εφαρμογή της μορφής αυτής βρίσκεται στην τεχνολογία AJAX, την οποία θα αναλύσουμε στην συνέχεια.

Παρά το γεγονός ότι η JSON προορίζεται ως μια μορφή serialization δεδομένων, ο σχεδιασμός της όμως ως υποσύνολο της γλώσσας JavaScript, δημιουργεί πολλές ανησυχίες για την ασφάλεια. Οι εν λόγω ανησυχίες συσσωρεύονται στην χρήση του διερμηνέα της JavaScript για την εκτέλεση δυναμικού κώδικα JSON όπως και JavaScript, εκθέτοντας έτσι ένα πρόγραμμα σε κακόβουλα scripts που μπορεί να περιέχονται στον κώδικα JSON – συχνά κύριο μέλημα όταν ασχολείται κανείς με την ανάκτηση δεδομένων από το διαδίκτυο. Αν και δεν είναι ο μοναδικός τρόπος για την επεξεργασία JSON, είναι όμως μια εύκολη, απλή και αρκετά δημοφιλής τεχνική που απορρέει από τον σχεδιασμό της μορφής JSON με κύριο σκοπό την πλήρη συμβατότητα της με την συνάρτηση eval() της JavaScript.

Μολονότι η μορφή JSON βασίστηκε σε ένα υποσύνολο της γλώσσας JavaScript (Standard ECMA-262 3rd Edition – December 1999) και χρησιμοποιείται συνήθως με αυτήν, θεωρείται ότι είναι μια μορφή δεδομένων ανεξάρτητη γλώσσας (language-independent). Ο κώδικας για την μεταγλώττιση και την παραγωγή δεδομένων σε μορφή JSON είναι άμεσα διαθέσιμος στο διαδίκτυο σε μια μεγάλη ποικιλία γλωσσών προγραμματισμού. [9]

1.11 Ajax

Η τεχνολογία Ajax, που μερικές φορές γράφεται ως AJAX (συντομογραφία για το Asynchronous JavaScript and XML), είναι μια ομάδα αλληλένδετων τεχνικών ανάπτυξης ιστοσελίδων, που χρησιμοποιούνται από τον υπολογιστή-πελάτη(client-side), για τη δημιουργία διαδραστικών διαδικτυακών εφαρμογών ή εφαρμογές διαδικτύου. Με τη τεχνική Ajax, οι διαδικτυακές εφαρμογές μπορούν να ανακτούν δεδομένα από το διακομιστή ασύγχρονα στο παρασκήνιο χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά των υφιστάμενων σελίδων. Η χρήση της τεχνικής Ajax έχει οδηγήσει σε αύξηση των διαλογικών ή διαδραστικών διεπαφών στις δυναμικές ιστοσελίδες καθώς και στη βελτίωση της ποιότητας των υπηρεσιών του διαδικτύου, λόγω της ασύγχρονης λειτουργίας του. Τα δεδομένα συνήθως ανακτώνται χρησιμοποιώντας το αντικείμενο XMLHttpRequest. Παρά την ονομασία, η χρήση του JavaScript και XML δεν απαιτείται στην πραγματικότητα, ούτε τα αιτήματα που κάνουμε πρέπει να είναι ασύγχρονα.

Η τεχνική Ajax δεν είναι μια τεχνολογία από μόνη της, αλλά μια ομάδα τεχνολογιών. Χρησιμοποιεί ένα συνδυασμό:

- HTML και CSS για τη σήμανση και για την εμφάνιση.
- Πρόσβαση στην DOM με την JavaScript, για να εμφανίζονται και να αλληλεπιδρούν δυναμικά με τις πληροφορίες που παρουσιάζονται.
- Μια μέθοδο για την ασύγχρονη ανταλλαγή δεδομένων μεταξύ browser και server, αποφεύγοντας έτσι τις επαναφορτώσεις των σελίδων. Το αντικείμενο αυτό είναι το XMLHttpRequest (XHR) και χρησιμοποιείται τις

περισσότερες φορές, αλλά μερικές φορές ένα αντικείμενο `iframe` ή μια δυναμικά προστιθέμενη `<script>` ετικέτα αντί αυτού.

- Μια μορφή για τα δεδομένα τα οποία αποστέλλονται στον browser. Οι πιο συνηθισμένες μορφές είναι είτε XML, είτε προ-μορφοποιημένη HTML, είτε απλό κείμενο, ή JavaScript Object Notation (JSON) αλλά θα μπορούσαν να δημιουργούνται δυναμικά από κάποιον υπολογιστή-εξυπηρετητή (server-side scripting).

Οι τεχνικές για την ασύγχρονη φόρτωση του περιεχομένου χρονολογούνται από τα μέσα της δεκαετίας του 1990. Τα Java applets εισήχθησαν στην πρώτη έκδοση της γλώσσας Java, το 1995 και επέτρεπαν σε client-side κώδικα να φορτώνει τα δεδομένα ασύγχρονα από τον web server αφού αυτή η ιστοσελίδα φορτωθεί. Το 1996, ο Internet Explorer παρουσίασε το στοιχείο `IFrame` στην HTML, το οποίο επέτρεπε, επίσης, να φορτώνει ασύγχρονα τα δεδομένα. Το 1999, η Microsoft δημιούργησε το XMLHTTP ActiveX control στον Internet Explorer 5, το οποίο υποστηρίζεται τώρα από τον Mozilla, το Safari και άλλα προγράμματα περιήγησης, τα οποία υποστηρίζουν και το αντικείμενο `XMLHttpRequest`. Ωστόσο, η τεχνική AJAX έγινε ευρύτερα γνωστή, αφού χρησιμοποιήθηκε από το Gmail (2004) και το Google Maps (2005).

Ο όρος "Ajax" επινοήθηκε το 2005 από τον Jesse James Garrett, ο οποίος σκέφτηκε τον όρο "Ajax", ενώ ήταν στο ντους, όταν συνειδητοποίησε την ανάγκη για ένα συνοπτικό όνομα που να εκπροσωπεί την ακολουθία των τεχνολογιών που είχε να προτείνει σε έναν πελάτη.

Στις 5 Απριλίου του 2006 το World Wide Web Consortium (W3C) κυκλοφόρησε το πρώτο σχέδιο προδιαγραφής για το αντικείμενο, σε μια προσπάθεια να δημιουργηθεί ένα πρότυπο επίσημης ιστοσελίδας.

1.12 Αρχείο `.htaccess`

Τι είναι το `.htaccess` αρχείο;

Το `.htaccess` είναι ένα αρχείο ρυθμίσεων του Apache web server, μέσω του οποίου μπορείτε να πραγματοποιήσετε διάφορες ρυθμίσεις που αφορούν το πακέτο hosting. [9]

Πως δημιουργώ ένα αρχείο `.htaccess`;

Η διαδικασία είναι πολύ εύκολη και μπορείτε να δημιουργήσετε το αρχείο χρησιμοποιώντας έναν απλό text editor όπως το notepad ή textpad. Αρχικά βεβαιωθείτε ότι δεν υπάρχει ήδη αρχείο `.htaccess` στο hosting σας. Αν υπάρχει

"κατεβάστε" το στον υπολογιστή σας και κάντε τις ρυθμίσεις που επιθυμείτε σε αυτό, για να μην επηρεάσει τυχόν υπάρχουσες ρυθμίσεις του site σας.

Αν δεν υπάρχει, ανοίξτε ένα κενό αρχείο με το notepad και αποθηκεύστε το ως .htaccess αρχείο. Μερικοί editors (πχ. notepad) δεν υποστηρίζουν αυτή τη μορφή, οπότε μπορείτε να το αποθηκεύσετε ως htaccess.txt και να το μετονομάσετε στην πορεία, σε .htaccess μέσω FTP client. Θα πρέπει να "ανεβάσετε" το αρχείο .htaccess στο κεντρικό φάκελο (httpdocs) του site σας, ή σε κάποιον υποφάκελο του httpdocs, με FTP.

Τι μπορώ να κάνω με το .htaccess αρχείο.

1. Δημιουργία δικών σας error pages

Μπορείτε να καθορίσετε το περιεχόμενο των error pages, ώστε να βελτιώσετε την εικόνα αυτών των σελίδων στην περίπτωση που εμφανιστεί κάποιο error στους επισκέπτες του site σας. Μερικές από τις δημοφιλέστερες error σελίδες είναι:

- **401 Authorization required.** Η σελίδα αυτή εμφανίζεται στις περιπτώσεις που οι χρήστες προσπαθούν να εισέλθουν σε προτατευόμενες με κωδικό σελίδες και έχουν εισάγει λάθος στοιχεία σύνδεσης.
- **403 Forbidden.** Εμφανίζεται όταν οι χρήστες προσπαθούν να αποκτήσουν πρόσβαση σε ένα αρχείο (κώδικα, web αρχείο, γραφικό κτλ.), του οποίου τα δικαιώματα δεν επιτρέπουν τέτοιες ενέργειες.
- **404 Not found.** Εμφανίζεται όταν οι χρήστες προσπαθούν να επισκεφτούν μια σελίδα, η οποία δεν υπάρχει στο site σας.
- **500 Internal Server Error.** Εμφανίζεται όταν υπάρχει πρόβλημα με το Hosting, που μπορεί να οφείλεται σε διάφορες αιτίες.

Αυτό που μπορείτε να κάνετε είναι να κρατήσετε το βασικό μήνυμα της σελίδας ώστε να είναι ξεκάθαρο το λάθος και να επεξεργαστείτε εικαστικά τις σελίδες. Στη συνέχεια αποθηκεύστε τις σε οποιαδήποτε σημείο στο server θέλετε και καθορίστε να εμφανίζονται ανάλογα. Εδώ θα χρησιμοποιήσετε και το .htaccess αρχείο.

Αν για παράδειγμα έχετε δημιουργήσει τις σελίδες 401.htm και 404.html, τότε θα πρέπει να εισάγετε μέσα στο αρχείο .htaccess τις παρακάτω γραμμές:

```
ErrorDocument 401 /401.html
```

```
ErrorDocument 404 /404.html
```

Αν για παράδειγμα, ονομάσατε τις σελίδες σας "notfound.html" και "notauthorized.html" και τις αποθηκεύσατε σε έναν υποφάκελο με όνομα "errorpages" τότε οι γραμμές που θα εισάγετε μέσα στο αρχείο .htaccess είναι:

```
ErrorDocument 401 /errorpages/notauthorized.html
```

```
ErrorDocument 404 /errorpages/notfound.html
```

Η συγκεκριμένη ενέργεια μπορεί να γίνει πολύ εύκολα και μέσα από το Plesk control panel.

2. Δυνατότητα απόκρυψης φακέλων ενός directory

Συμβαίνει αρκετά συχνά να μην έχει εισαχθεί από χρήστες ένα αρχείο index στον κατάλογο φακέλων τους, με αποτέλεσμα αν κάποιος πληκτρολογήσει το όνομα του directory σε έναν browser να εμφανιστούν όλα τα αρχεία μέσα σε αυτό. Αυτό είναι ρίσκο για την ασφάλεια του site σας. Το μόνο που χρειάζεται να κάνετε είναι να εισάγετε την παρακάτω εντολή στο αρχείο .htaccess για να αποκρύψετε τον κατάλογο των αρχείων σας, από τυχόν κακόβουλους χρήστες ή hackers:

```
Options -Indexes
```

3. Δυνατότητα αποκλεισμού συγκεκριμένων IP διευθύνσεων

Στην περίπτωση που θέλετε να επιτρέψετε την πρόσβαση στο site σας σε συγκεκριμένες IPs ή να μπλοκάρετε εντελώς IPs από το να βλέπουν το περιεχόμενο της σελίδας σας, μπορείτε να εισάγετε τις παρακάτω εντολές στο αρχείο .htaccess. Προϋποθέτει ότι γνωρίζετε την IP των χρηστών που θέλετε να περιορίσετε. (όπου IP εισάγετε την IP που επιθυμείτε)

```
order allow,deny
```

```
deny from IP  
allow from all
```

Για περισσότερες από μία IPs εισάγετε την μία κάτω από την άλλη, όπως εμφανίζεται στις παρακάτω εντολές. Επιπλέον μπορείτε να αποκλείσετε μαζικά IPs εισάγοντας τον αριθμό της IP έως το προτελευταίο ψηφίο:

```
order allow,deny  
deny from 125.30.5.1  
deny from 125.30.5.  
allow from all
```

4. Δυνατότητα ανακατεύθυνσης

Άλλη μία σημαντική ενέργεια που μπορείτε να πραγματοποιήσετε μέσω του .htaccess αρχείου είναι η ανακατεύθυνση αιτημάτων προς διαφορετικά αρχεία στον ίδιο server ή προς ένα εντελώς διαφορετικό website. Για παράδειγμα, εάν έχετε το αρχείο με όνομα "oldpage.htm" και θέλετε να κάνετε ανακατεύθυνση σε ένα άλλο, εντελώς διαφορετικό αρχείο, εισάγετε την εντολή:

```
Redirect /oldpage.htm http://www.diforetikosite.com/new/file/location.html
```

Αριστερά μόνο το path για το αρχείο και χωρισμένο με κενό, ολόκληρο το url στο οποίο θέλετε να κάνετε redirect στα δεξιά. Αν τώρα το παλιό αρχείο βρίσκεται σε υποφάκελο, πχ. old/oldpage.htm, τότε η εντολή θα είναι:

```
Redirect /old/oldpage.htm http://www.diforetikosite.com/new/file/location.html
```

Μπορείτε επίσης να ανακατευθύνετε ολόκληρους καταλόγους (directories) του site σας. Για παράδειγμα, έστω ότι έχετε ένα directory με υποφακέλους και αρχεία στο παλιό σας site και έχετε περάσει τα ίδια αρχεία σε ένα νέο site. Μπορείτε να κάνετε

ανακατεύθυνση όλων των αρχείων μαζί σε αυτό το directory, χωρίς να χρειαστεί να αναφέρετε σε εντολές το καθένα ξεχωριστά. Η εντολή θα είναι:

```
Redirect /olddirectory http://www.neosite.gr/newdirectory
```

Με αυτόν τον τρόπο κάθε αναζήτηση στο παλιό σας site που σχετίζεται με τα αρχεία του olddirectory θα ανακατευθύνεται στο νέο σας site, ενώ στο URL θα εμφανίζεται και η αλλαγή. Δηλαδή όταν κάποιος θέλει να μεταβεί πχ. στο URL <http://www.paliosite.gr/olddirectory/oldfiles/images/arxeio.jpg> θα γίνει αυτόματα redirect προς <http://www.beosite.gr/newdirectory/oldfiles/images/arxeio.jpg>

Σημείωση: Μια λίγο πιο advanced, δημοφιλής ρύθμιση με rewrite rule είναι το redirection από ένα domain χωρίς "www" προς το ίδιο domain με "www". Πχ. Όταν ο χρήστης ανοίγει το site "example.com" να τον κατευθύνει αυτόματα στο "www.example.com":

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^example.com [NC]
RewriteRule ^(.*)$ http://www.example.com/$1 [L,R=301]
```

ΕΠΙΛΟΓΟΣ

«Ο επίλογος θα ανακεφαλαιώνει όσα έχουν ειπωθεί στο κεφάλαιο αυτό και συγχρόνως θα υπάρξει σύνδεση με το επόμενο κεφάλαιο.»

Οι παραπάνω τεχνολογίες και οι γλωσσές που αναλυθηκαν στο κεφαλαιο αυτο θα χρησιμοποιηθουν στην συνεχεια για την ιλοποιηση της ιστοσελιδας με την βοηθεια καποιον προγραμματων των οποιον η χρηση και η εγκατασταση θα αναλυθει στο επομενο κεφαλαιο.

2. Εργαλεία ανάπτυξης δυναμικών εφαρμογών

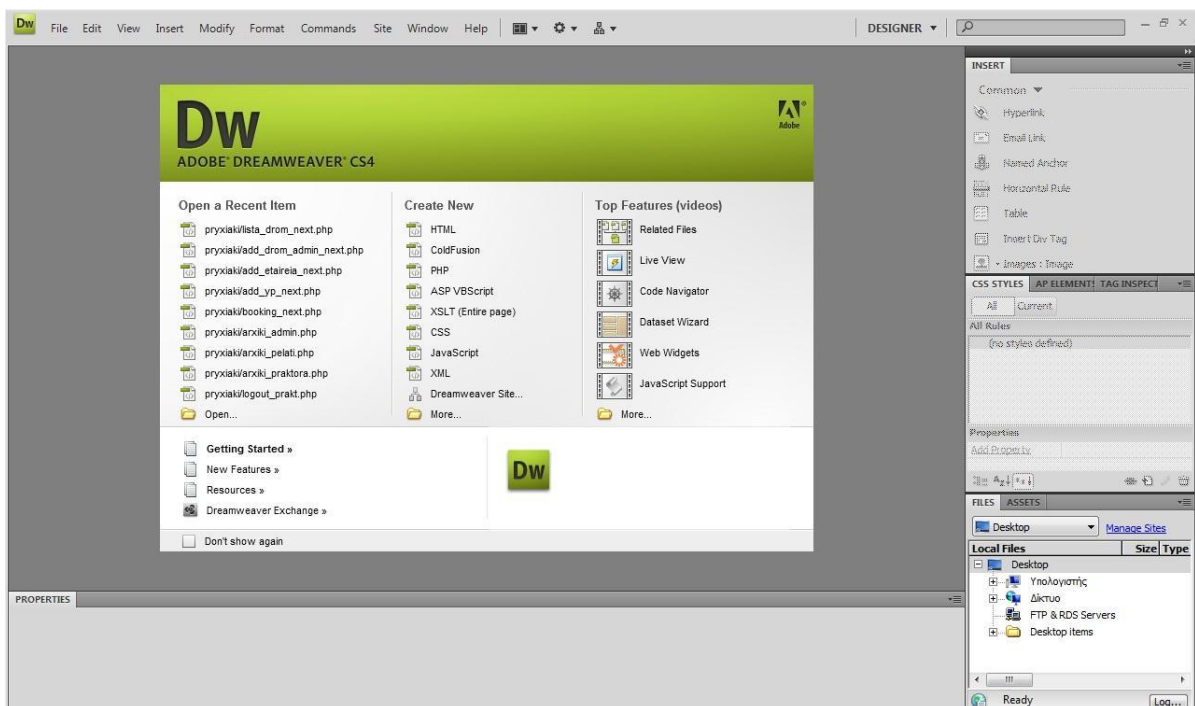
ΕΙΣΑΓΩΓΗ

Στο παρόν κεφάλαιο γίνεται αναφορά στις εφαρμογές και τεχνολογίες που χρησιμοποιήθηκαν για την διεκπεραίωση της πτυχιακής εργασίας.

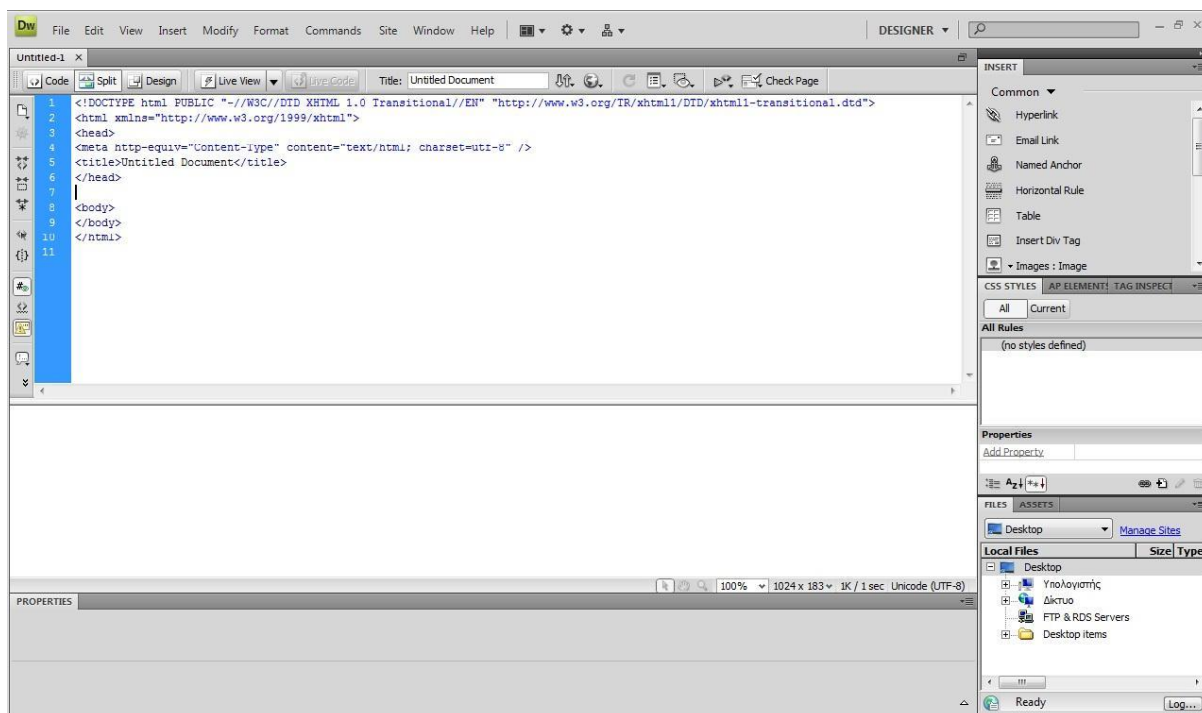
2.1 Adobe Dreamweaver

2.1.1 Τι είναι το Dreamweaver

Το Adobe Dreamweaver είναι μια εφαρμογή ανάπτυξης ιστοσελίδων η οποία αρχικά δημιουργήθηκε από τη Macromedia και πλέον ανήκει στην Adobe Systems η οποία απέκτησε την Macromedia το 2005. Το Dreamweaver είναι διαθέσιμο και για χρήστες που χρησιμοποιούν Mac αλλά και Windows λειτουργικό σύστημα. Οι πιο πρόσφατες εκδόσεις έχουν ενσωματώσει υποστήριξη για τεχνολογίες web όπως CSS, JavaScript και διάφορες server-side γλώσσες scripting όπως επίσης frameworks συμπεριλαμβανομένου ASP.NET, ColdFusion, JavaServer Pages και PHP. [13]



Εικόνα 3 «Αρχική οθόνη Dreamweaver»



Εικόνα 4 «Περιβάλλον διαχείρισης Dreamweaver»

2.1.2 Χαρακτηριστικά του Dreamweaver

Βασισμένο σε WYSIWYG Presto, το Dreamweaver μπορεί να κρύψει τις λεπτομέρειες του κώδικα HTML των σελίδων από το χρήστη, κάτι που καθιστά δυνατό για τους μη-επαγγελματίες να δημιουργήσουν ιστοσελίδες και ιστοχώρους ολόκληρους. Μια επαγγελματική κριτική αυτής της προσέγγισης είναι ότι παράγει HTML σελίδες των οποίων το μέγεθος αρχείων και η ποσότητα κώδικα HTML είναι πολύ μεγαλύτερα από αυτά που πρέπει να είναι, το οποίο μπορεί να προκαλέσει κακή απόδοση στις μηχανές αναζήτησης. Αυτό ισχύει ιδιαίτερα διότι η εφαρμογή το καθιστά πολύ εύκολο να δημιουργήσει σχεδιαγράμματα βασισμένα σε πίνακες (table-based layouts).

Επιπλέον, μερικοί υπεύθυνοι για την ανάπτυξη ιστοχώρων έχουν επικρίνει το Dreamweaver στο παρελθόν για την παραγωγή του κώδικα που συχνά δεν συμμορφώνεται με W3C πρότυπα, αν και οι πρόσφατες εκδόσεις είναι πιο υποχωρητικές. Το Dreamweaver 8.0 είχε κακή απόδοση στο test Acid2, το οποίο είναι αναπτυγμένο από Web Standards Project. Εντούτοις, η Macromedia έχει αυξήσει την υποστήριξη για CSS και άλλους τρόπους να σχεδιαστεί μια σελίδα χωρίς πίνακες στις πιο πρόσφατες εκδόσεις της εφαρμογής, με τη δυνατότητα να μετατραπούν οι πίνακες σε στρώματα και αντίστροφα. [13]

Το Dreamweaver επιτρέπει στους χρήστες των ιστοχώρων να μπορούν να προβάλουν τις ιστοσελίδες τους σε πολλούς browsers, υπό τον όρο ότι είναι

εγκατεστημένα στον υπολογιστή τους. Έχει επίσης μερικά εργαλεία διαχείρισης ιστοσελίδων, όπως η δυνατότητα να βρεθούν και να αντικατασταθούν οι γραμμές κειμένου ή κώδικα με οποιοδήποτε παράμετρο σε ολόκληρο τον ιστοχώρο, και ένα χαρακτηριστικό γνώρισμα templatization για τη δημιουργία των πολλαπλάσιων σελίδων με τις παρόμοιες δομές. Το behaviors panel επιτρέπει επίσης τη χρήση βασικού JavaScript χωρίς οποιαδήποτε γνώση κωδικοποίησης.

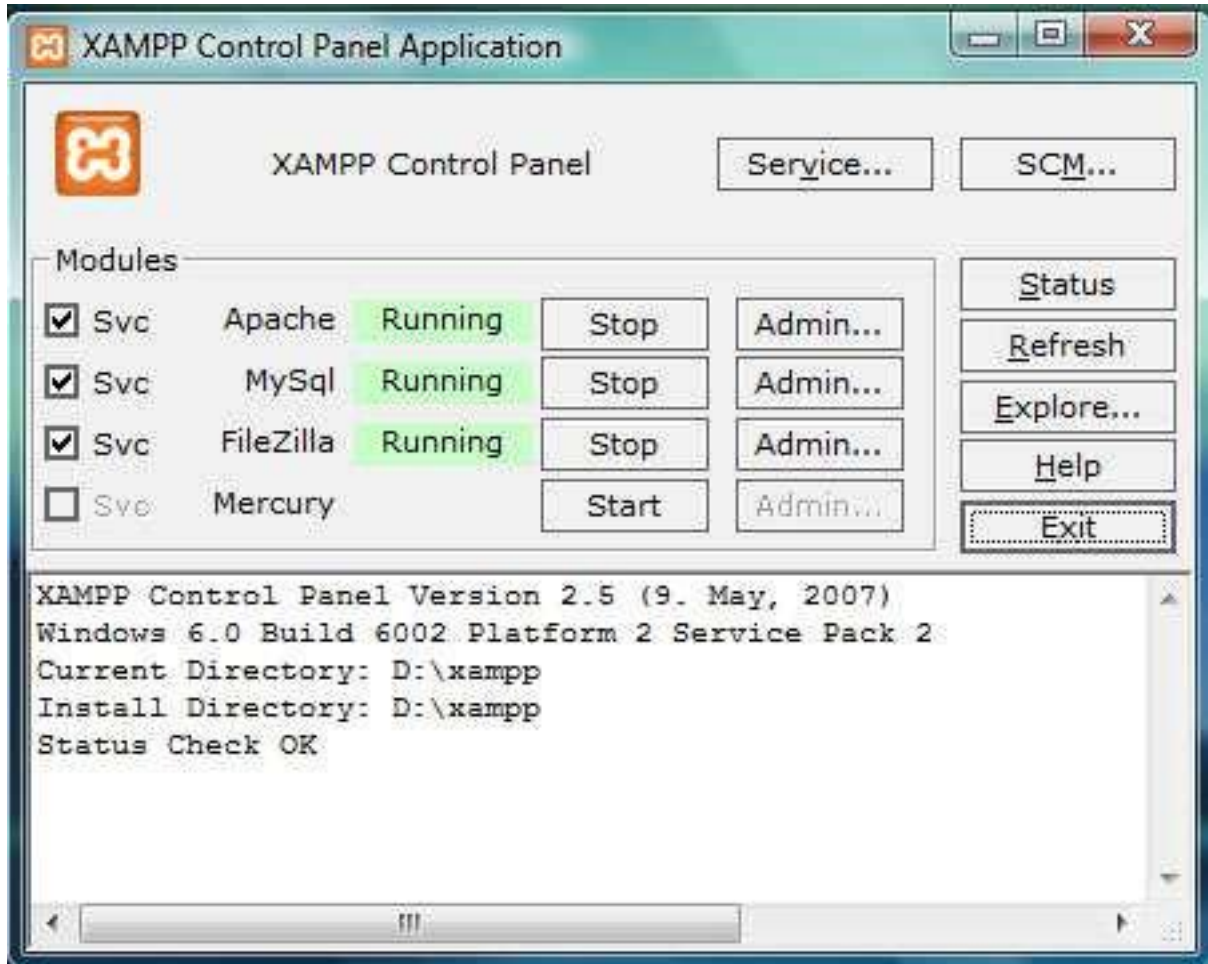
Το Dreamweaver μπορεί να χρησιμοποιήσει Extensions - μικρά προγράμματα, τα οποία οποιοσδήποτε υπεύθυνος για την ανάπτυξη Ιστού μπορεί να γράψει (συνήθως μέσω HTML και JavaScript). Τα Extensions παρέχουν επιπλέον λειτουργίες στο λογισμικό για οποιονδήποτε επιθυμεί να τις μεταφορτώσει και να τις εγκαταστήσει. Το Dreamweaver υποστηρίζεται από ένα μεγάλο αριθμό δημιουργών extensions που καθιστά τα extensions διαθέσιμα (και εμπορικά και ελεύθερα) για τους περισσότερους στόχους ανάπτυξης ιστοσελίδων από τα πιο απλά αποτελέσματα στα πιο πλήρη. Όπως άλλοι editors HTML, έτσι και το Dreamweaver επεξεργάζεται τα αρχεία τοπικά, κατόπιν φορτώνει όλα τα επεξεργασμένα αρχεία στο remote web server με χρησιμοποίηση δικτύου FTP, SFTP, ή WebDAV. [13]

2.2 XAMPP



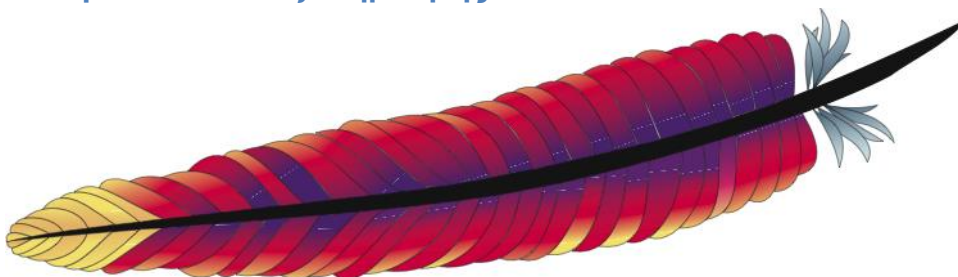
Το XAMPP είναι ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server. Αυτό είναι απαραίτητο πριν εγκαταστήσουμε το Joomla αφού η δημιουργία και η ανάπτυξη του ιστοτόπου μας θα γίνει τοπικά, οπότε χρειαζόμαστε να χρησιμοποιηθεί ένας τοπικός server, πάνω στον οποίο θα στηθεί η ιστοσελίδα. Το XAMPP είναι ένα ελεύθερο και ανοικτό cross-platform web server package, που αποτελείται κυρίως από τον Apache HTTP Server, τη MySQL βάση δεδομένων και των διεργασιών για scripts γραμμένα σε γλώσσες

προγραμματισμού PHP και Perl. Η εγκατάστασή του είναι πολύ απλή. Το μόνο που έχουμε να κάνουμε είναι να τρέξουμε το αρχείο που μπορούμε να βρούμε δωρεάν στο διαδίκτυο μιας και το XAMPP όπως και το Joomla είναι ανοικτού κώδικα (OpenSource) προγράμματα.



Εικόνα 5 «To Control Panel του XAMPP»

2.3 Apache HTTP εξυπηρετητής



Η δημιουργία ενός δυναμικού site προαπαιτεί κάποιες άλλες λειτουργίες. Αρχικά η εγκατάσταση του site θα γίνει τοπικά οπότε πρέπει να χρησιμοποιηθεί ένας τοπικός server, πάνω στον οποίο θα στηθεί το site. Σε αυτή την εργασία θα

χρησιμοποιηθεί ο Apache HTTP. Ο Apache HTTP γνωστός και απλά ως Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όταν επισκεπτόμαστε έναν ιστότοπο ο πλοηγός μας επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP (Hypertext Transfer Protocol), ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης.

Ο Apache περιέχει πολλές διαφορετικές λειτουργικές μονάδες ελέγχου ταυτότητας που μπορούν να χρησιμοποιηθούν για τον έλεγχο της εγκυρότητας των δεδομένων, που εισάγονται από τον χρήστη. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων πχ. Oracle, MySQL. [2]

2.4 MySQL



Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα μας διασφαλίζοντας ότι πολλοί χρήστες θα μπορούν να δουλεύουν ταυτόχρονα, παρέχει γρήγορη πρόσβαση και διασφαλίζει ότι μόνο οι πιστοποιημένοι χρήστες θα μπορούν να έχουν πρόσβαση. Συνεπώς, η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, σήμερα θυγατρική της Sun Microsystems. [19]

2.4.1 Πλεονεκτήματα της MySQL

Υψηλή απόδοση. Η MySQL είναι αρκετά πιο γρήγορη από τις ανταγωνίστριές της.

Χαμηλό κόστος. Η MySQL είναι διαθέσιμη δωρεάν, με άδεια ανοικτού κώδικα (Open Source), ή με χαμηλό κόστος, αν πάρετε εμπορική άδεια.

Ευκολία χρήσης. Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL.

Μεταφερσιμότητα. Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα UNIX, όπως επίσης και στα Windows.

Κώδικας προέλευσης. Όπως και με την PHP, μπορούμε να πάρουμε και να τροποποιήσουμε τον κώδικα προέλευσης της MySQL.

Διαθεσιμότητα υποστήριξης. Από την MySQLAB (www.mysql.com) μπορούμε να έχουμε υποστήριξη, εκπαίδευση, συμβουλές και πιστοποιητικά. [2]

2.5 Λογισμικό PhpMyAdmin



2.5.1 Τι είναι το PhpMyAdmin

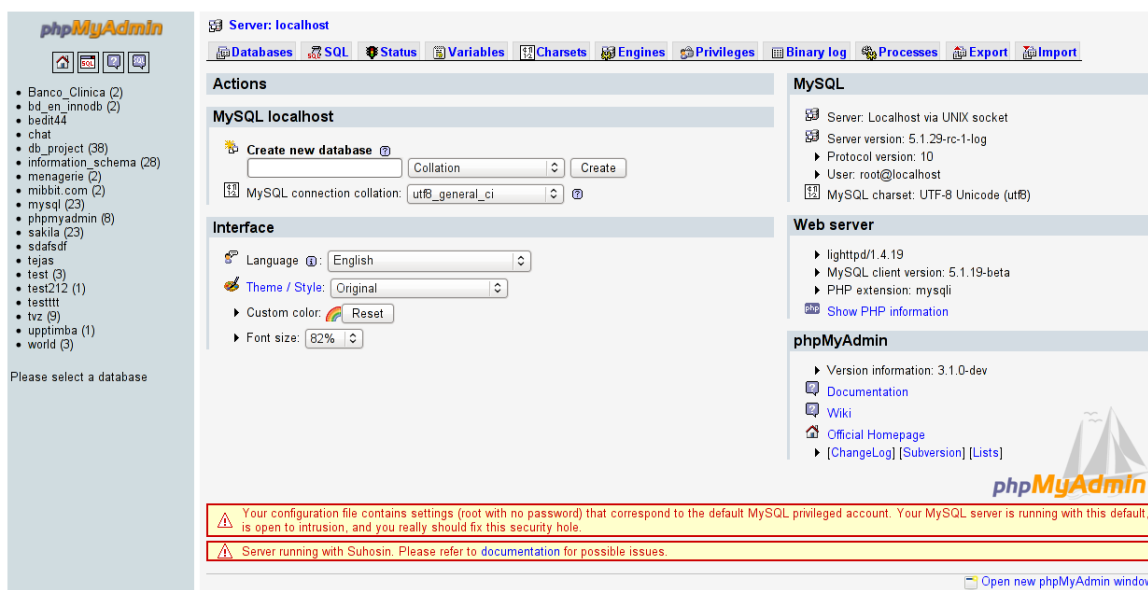
Το phpMyAdmin είναι ένα δωρεάν λογισμικό, ένα εργαλείο γραμμένο σε PHP που προορίζεται για το χειρισμό της MySQL μέσω του World Wide Web. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα ενεργειών με MySQL. Οι πιο συχνά χρησιμοποιούμενες λειτουργίες υποστηρίζονται από το περιβάλλον εργασίας του χρήστη είναι (διαχείριση βάσεων δεδομένων, πίνακες, πεδία, σχέσεις, ευρετήρια, οι χρήστες, δικαιώματα χρηστών κλπ), ενώ εξακολουθούμε να έχουμε τη δυνατότητα να εκτελέσουμε άμεσα οποιαδήποτε πρόταση SQL. Το phpMyAdmin έρχεται με ένα ευρύ φάσμα εγγράφων το οποίο βοηθάει το χρήστη στην εκτέλεση των εργασιών που υποστηρίζονται από το phpMyAdmin.

Για το phpMyAdmin επίσης μπορούμε να βρούμε βοήθεια και σε ένα βιβλίο που γράφτηκε από έναν από τους προγραμματιστές, το "Mastering phpMyAdmin for Effective MySQL Management". Για να είναι πιο εύκολη η χρήση του σε ένα ευρύ φάσμα ανθρώπων, το phpMyAdmin έχει μεταφραστεί σε 62 γλώσσες και υποστηρίζει τόσο από LTR όσο και από RTL γλώσσες. Από την έκδοση 3.0.0, στο phpMyAdmin προσχώρησαν στην GoPHP5 πρωτοβουλία και σβήστηκε ο κώδικας συμβατότητας για τις παλαιότερες εκδόσεις PHP και MySQL. Η έκδοση 3 και

αργότερα απαιτεί τουλάχιστον PHP 5.2 και MySQL 5. Το phpMyAdmin έχει κερδίσει πολλά βραβεία . Μεταξύ άλλων, επιλέχθηκε ως η καλύτερη PHP εφαρμογή σε διάφορα βραβεία και έχει κερδίσει κάθε χρόνο τα βραβεία SourceForge.net Community Choice Awards σαν "Best Tool or Utility for SysAdmins". Τέλος το phpMyAdmin είναι ένα δεκατριών ετών έργο με μια σταθερή και ευέλικτη βάση κώδικα. [12]

2.5.2 Χαρακτηριστικά του PhpMyAdmin

- Διορατικό Web interface
- Υποστήριξη για τα περισσότερα χαρακτηριστικά MySQL:
 - περιήγηση και κατάργηση βάσεων δεδομένων, πινάκων, πεδίων και ευρετηρίων
 - δημιουργία, αντιγραφή, κατάργηση, μετονομασία και αλλαγές στις βάσεις δεδομένων, πίνακες, πεδία και ευρετήρια
 - συντήρηση διακομιστή, βάσεις δεδομένων και πινάκων, με προτάσεις για τη διαμόρφωση του διακομιστή
 - εκτέλεση, επεξεργασία και επιλογή σελιδοδείκτη για οποιαδήποτε SQL-δήλωση.
 - διαχείριση MySQL χρηστών και τα προνόμια τους
 - διαχείριση αποθηκευμένων διαδικασιών και εναύσματα
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV, SQL, XML, PDF, το πρότυπο
- ISO / IEC 26300 - OpenDocument κειμένου και λογιστικών φύλλων,
- Word.
- Η διαχείριση πολλών διακομιστών
- Δημιουργία PDF γραφικών της διάταξης βάσης δεδομένων
- Δημιουργία σύνθετων εκφράσεων χρησιμοποιώντας Query-by-example (QBE)
- Αναζήτηση σε καθολικό επίπεδο σε μια βάση δεδομένων ή ένα υποσύνολο της
- Μετατροπή αποθηκευμένων δεδομένων σε οποιαδήποτε μορφή χρησιμοποιώντας ένα σύνολο από προκαθορισμένες λειτουργίες, όπως η εμφάνιση BLOB-δεδομένα ως εικόνα ή download-Link.



Εικόνα 6 «Σελίδα διαχείρισης της εφαρμογής phpMyAdmin»

2.6 Adobe Photoshop



Το Photoshop της εταιρείας Adobe αποτελεί ένα ιδιαίτερα εκτεταμένο λογισμικό με τμήματα του να είναι αρκετά πολύπλοκα. Όλοι όσοι ασχολούνται με υπολογιστές και ιδιαίτερα με επεξεργασία εικόνων και πολυμέσα έχουν τουλάχιστον ακούσει γι' αυτό σε κάποια φάση της δουλειάς τους και συνεπώς γνωρίζουν περί τίνος πρόκειται.

Ελάχιστοι, όμως, είναι αυτοί που εκμεταλλεύονται το μεγαλύτερο μέρος των δυνατοτήτων του και ακόμη λιγότεροι εκείνοι που έστω γνωρίζουν μέχρι ποιο σημείο μπορούν να φτάσουν με αυτό.

Το Photoshop είναι ένα λογισμικό με το οποίο μπορεί κανείς να επεξεργαστεί ψηφιακές εικόνες αλλά και να σχεδιάσει από την αρχή κάτι εντελώς δικό του. Η κυριότερη βέβαια εφαρμογή του είναι στην επεξεργασία εικόνας. Είναι σχεδιασμένο να ικανοποιεί τόσο τον εξειδικευμένο χρήστη με γνώσεις επεξεργασίας εικόνας όσο και τον καλλιτέχνη/φωτογράφο/σχεδιαστή. Με τα χρόνια και με τις συνεχείς βελτιώσεις του δίνει έμφαση στην αύξηση της παραγωγικότητας της επεξεργασίας εικόνας και ενσωματώνει ένα πλήθος από υλοποιημένους αλγόριθμους που συνεχώς προκύπτουν από το χώρο έρευνας της ψηφιακής επεξεργασίας εικόνας. Έτσι, έχει αποκτήσει το χαρακτηριστικό να μπορεί να πραγματοποιεί πλήθος από εργασίες εύκολα και γρήγορα και μάλιστα με πολλαπλούς τρόπους. Είναι δηλαδή δυνατό να εκτελεστεί μια συγκεκριμένη ενέργεια είτε με τη χρήση του μενού, είτε με τη χρήση συνδυασμού πλήκτρων από το πληκτρολόγιο είτε με τη χρήση εικονιδίων από τις διάφορες γραμμές εργαλείων που προσφέρει, έτσι ώστε τελικά να μπορεί να ανταποκρίνεται σε κάθε στυλ εργασίας που μπορεί ο χρήστης του να έχει ή να προτιμά. [9]

ΕΠΙΛΟΓΟΣ

Οι παραπάνω εφαρμογές χρησιμοποιήθηκαν για την διαδικασία ανάπτυξης της ιστοσελίδας της πτυχιακής εργασίας. Στην συνέχεια θα αναλυθεί το μέρος με την υλοποίηση της βάσης δεδομένων και την διαχείριση της.

ΚΕΦΑΛΑΙΟ 3

Σχεδιασμός Βάσεων Δεδομένων

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται η μελέτη των βάσεων δεδομένων, αναλύονται οι μορφές, οι αρχιτεκτονικές και διάφορα στοιχεία τους. Αναφέρονται τα συνηθισμένα προβλήματα ή δυσκολίες που συναντώνται καθώς και ο τρόπος σύνδεσης της βάσης δεδομένων με PHP.

3.1 Βασικές Έννοιες



3.1.1 Τι είναι οι βάσεις δεδομένων

Ως βάση δεδομένων ορίζουμε μια συλλογή από συστηματικά οργανωμένα (formatted) και σχετιζόμενα μεταξύ τους δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων - τμημάτων πληροφορίας – ηλεκτρονικά αποθηκευμένων. [1]

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

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

3.1.3 Τι είναι το Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή Database System(DBS);

ΣΒΔ καλείται ο συνδυασμός και η σύνδεση της βάσης με το λογισμικό.

3.1.4 Τι είναι ο DataBase Administrator (DBA) ή διαχειριστής μιας Βάσης δεδομένων (ΔΒΑ) και ποιος ο ρόλος του;

Ο διαχειριστής μιας Βάσης δεδομένων (DBA, DataBase Administrator) είναι αυτός που έχει την ευθύνη για τον σωστό, αποδοτικό και αξιόπιστο τρόπο δημιουργίας και λειτουργίας μια βάσης δεδομένων. Οι αρμοδιότητές του είναι οι εξής :

- Η απόφαση για το είδος των πληροφοριών που πρέπει να αποθηκευτούν.
- Η απόφαση για τον τρόπο αποθήκευσης και πρόσβασης στις πληροφορίες αυτές.
- Η συνεργασία με τους τελικούς χρήστες.
- Η απόφαση για τον τρόπο εξασφάλισης των πληροφοριών.
- Η απόφαση για το κάθε πότε θα γίνονται αντίγραφα ασφαλείας (backup) των αρχείων.
- Η παρακολούθηση της σωστής λειτουργίας της βάσης δεδομένων και η απαιτούμενη προσαρμογή της.

3.2 Αρχιτεκτονική τριών επιπέδων

Ένα σύστημα βάσεων δεδομένων παρέχει στους χρήστες μια αφαιρετική άποψη. Το σύστημα κρύβει (από τους χρήστες) ορισμένες λεπτομέρειες για το πώς τα δεδομένα αποθηκεύονται και διατηρούνται. Γι' αυτό υπάρχουν τρία επίπεδα αφαίρεσης:

- 1. Φυσικό επίπεδο:

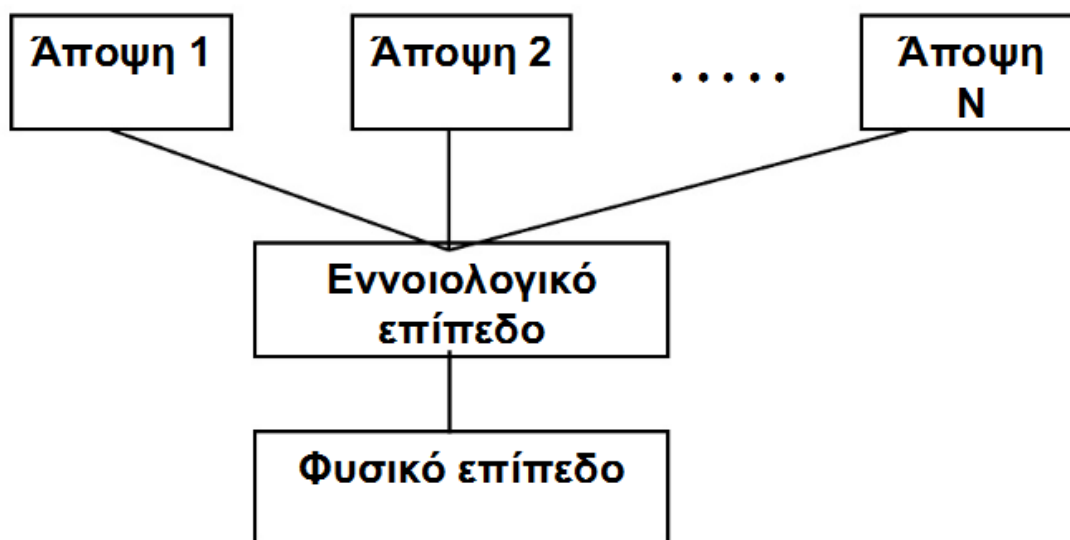
Είναι το χαμηλότερο επίπεδο αφαίρεσης και περιγράφει πώς αποθηκεύονται τα δεδομένα.

- 2. Εννοιολογικό επίπεδο:

Το αμέσως υψηλότερο επίπεδο αφαίρεσης. Περιγράφει τι είδους δεδομένα αποθηκεύονται και τις συσχετίσεις μεταξύ αυτών των δεδομένων.

- 3. Επίπεδο άποψης:

Το πιο υψηλό επίπεδο και περιγράφει τμήματα της βάσης δεδομένων για κάθε συγκεκριμένη ομάδα χρηστών, μπορεί να υπάρχουν πολλές διαφορετικές απόψεις, π.χ. Οι ταμίες σε μία τράπεζα παίρνουν μία άποψη των απολογισμών πελατών, αλλά όχι των δεδομένων μισθοδοτικών καταστάσεων.



Εικόνα 7 «Επίπεδο άποψης»

3.3 Μορφές Βάσεων Δεδομένων

- **Οι Αντικειμενοστραφείς Βάσεις Δεδομένων**

Μια βάση δεδομένων αντικειμένου (Object database) (επίσης αναφέρεται και ως

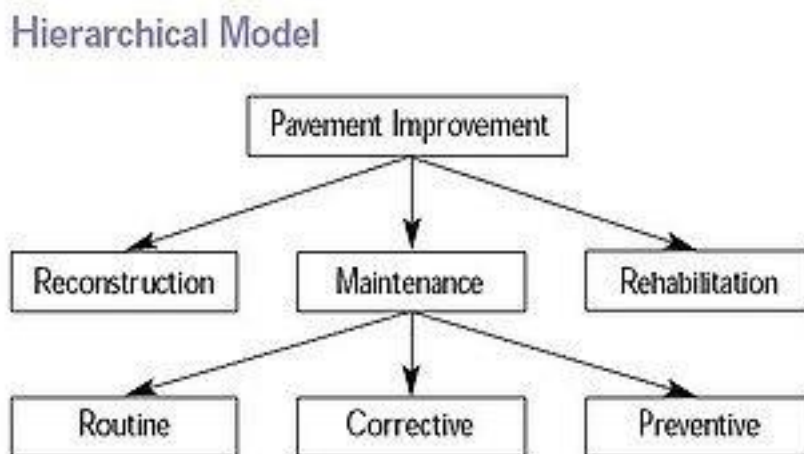
αντικειμενοστραφές σύστημα διαχείρισης βάσεων δεδομένων) είναι ένα σύστημα διαχείρισης βάσεων δεδομένων στο οποίο οι πληροφορίες αναπαρίστανται με τη μορφή αντικειμένων όπως χρησιμοποιούνται στον αντικειμενοστραφή προγραμματισμό. Οι βάσεις δεδομένων αντικειμένου είναι ένας τομέας μέσα στην ευρύτερη αγορά συστημάτων διαχείρισης βάσεων δεδομένων (DBMS) όπου κυριαρχούν τα μοντέλα σχεσιακής διαχείρισης βάσης δεδομένων.

- **Οι Ημιδομημένες Βάσεις Δεδομένων**

Στο ημιδομημένο μοντέλο (semi-structured) δεν υπάρχει κανένας διαχωρισμός μεταξύ των δεδομένων και του σχήματος, και το ποσό δομής χρησιμοποιούμενο αλλά εξαρτάται από το εκάστοτε σκοπό.

- **Οι Ιεραρχικές Βάσεις Δεδομένων**

Στις Ιεραρχικές (Hierarchical) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links). Η κάθε εγγραφή μπορεί να συνδέεται προς τα πάνω μόνο με μία άλλη εγγραφή (γονέας), ενώ μπορεί να έχει έως δύο εγγραφές που να εξαρτώνται απ' αυτήν (παιδιά). Υπάρχει μία μόνο εγγραφή ρίζα (root), απ' την οποία εξαρτώνται όλες οι άλλες εγγραφές της βάσης δεδομένων. Έχουν το μειονέκτημα ότι είναι πολύπλοκες στην επεξεργασία των εγγραφών τους (προσθήκη, διαγραφή, τροποποίηση).



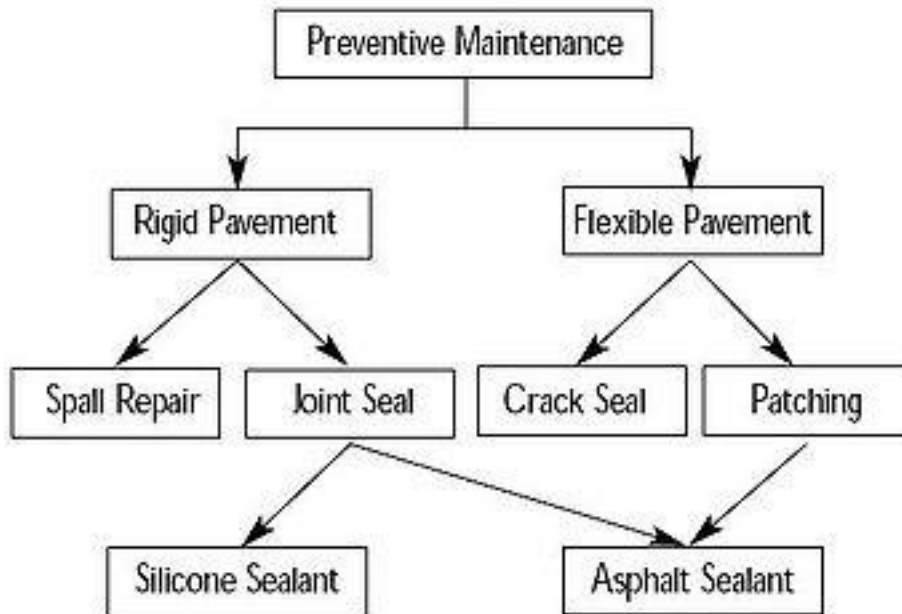
Εικόνα 8 «Ιεραρχικό Μοντέλο»

- **Οι Δικτυωτές Βάσεις Δεδομένων**

Στις Δικτυωτές (Network) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links), όπως ακριβώς και στις ιεραρχικές βάσεις δεδομένων, με την διαφορά ότι μια εγγραφή μπορεί να συνδέεται προς τα πάνω με περισσότερες από μία

πατρικές εγγραφές (parent records). Είναι λογικά πιο δύσχρηστες αλλά και πιο γρήγορες από τις ιεραρχικές βάσεις δεδομένων.

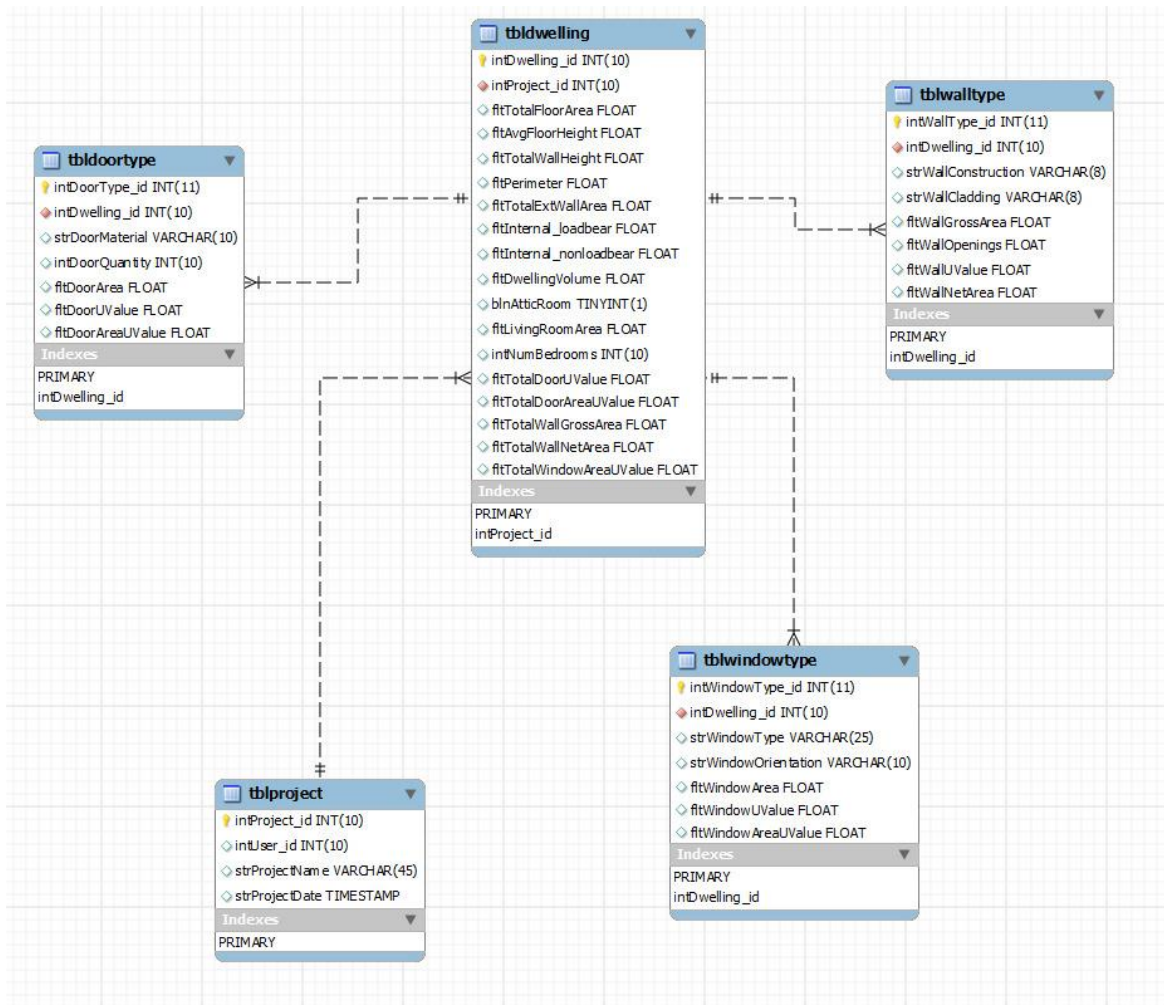
Network Model



Εικόνα 9 «Δικτυωτές Βάσεις Δεδομένων»

- **Οι Σχεσιακές Βάσεις Δεδομένων**

Στις Σχεσιακές (Relational) βάσεις δεδομένων, τα δεδομένα συνδέονται μεταξύ τους με σχέσεις (relations), οι οποίες προκύπτουν από τα κοινά πεδία που υπάρχουν σε διαφορετικά αρχεία. Τα αρχεία αποκαλούνται πίνακες (tables), οι εγγραφές γραμμές (rows) και τα πεδία στήλες (columns). Η ύπαρξη μιας κοινής τιμής στα πεδία δύο αρχείων καθορίζει και μια σχέση μεταξύ των γραμμών διαφορετικών πινάκων. Οι σχεσιακές βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι λογικά κατανοητές και πολύ ευέλικτες και δεκτικές σε αλλαγές. [4]



Εικόνα 10 «Σχισιακές Βάσεις Δεδομένων»

3.4 Στοιχεία Βάσεων Δεδομένων

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται (και οργανώνονται) στις εξής στοιχειώδεις μορφές:

- **Πεδίο (Field)**, είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.
- **Εγγραφή (Record)**, είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.
- **Αρχείο (File)**, είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.
- **Πρωτεύον Κλειδί (Primary Key)**, είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή. Αρχικό κλειδί ενός σχεσιακού πίνακα προσδιορίζει μεμονωμένα κάθε εγγραφή του πίνακα. Μπορεί είτε να είναι μια κανονική ιδιότητα που είναι εγγυημένη για να είναι μοναδική (όπως

ο αριθμός κοινωνικής ασφάλισης σε έναν πίνακα χωρίς περισσότερα από ένα αρχεία ανά άτομο) ή μπορεί να παραχθεί από το DBMS (σύστημα διαχείρισης βάσεων δεδομένων). Τα αρχικά κλειδιά μπορούν να αποτελούν τόσο απλές ιδιότητες όσο και πολλαπλές σε συνδυασμό.

- **Ξένο Κλειδί (Foreign Key)**, είναι ένα πεδίο που έχει το το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου αρχείου. Αναλυτικότερα ξένο κλειδί είναι μια ιδιότητα σε έναν σχεσιακό πίνακα που ταιριάζει με ένα υποψήφιο κλειδί ενός άλλου πίνακα. Το ξένο κλειδί προσδιορίζει μια στήλη ή ένα σύνολο στηλών σε έναν πίνακα που αναφέρεται σε μια στήλη ή ένα σύνολο στηλών σε έναν άλλο πίνακα.

3.5 Συνηθισμένα προβλήματα ή δυσκολίες που συναντώνται στις ΒΔ

1. Ασυνέπειας και πλεονασμού δεδομένων

Ο πλεονασμός στοιχείων εμφανίζεται στα συστήματα βάσεων δεδομένων που έχουν έναν πεδίο που επαναλαμβάνεται σε δύο ή περισσότερους πίνακες. Για παράδειγμα, η περίπτωση που πελάτης έχει καταχωρηθεί δύο φορές και συνδέεται με κάθε προϊόν που αγοράζεται επιφέρει πλεονασμός στα δεδομένων, γνωστό ως πηγή ασυνέπειας, δεδομένου ότι ο πελάτης να εμφανιστεί με τις διαφορετικές τιμές για τις δεδομένες ιδιότητες.

2. Ακεραιότητα δεδομένων

Οι περιορισμοί ακεραιότητας χρησιμοποιούνται για να εξασφαλίσουν την ακρίβεια και τη συνέπεια των στοιχείων σε μια σχεσιακή βάση δεδομένων. Η ακεραιότητα στοιχείων αντιμετωπίζεται σε μια σχεσιακή βάση δεδομένων μέσω της έννοιας της αναφερόμενης ακεραιότητας. Υπάρχουν πολλοί τύποι περιορισμών ακεραιότητας που διαδραματίζουν έναν ρόλο στην αναφερόμενη ακεραιότητα. Χωρίζονται στις εξής κατηγορίες σύμφωνα με τον Codd.

2.1. Ακεραιότητα οντοτήτων (Entity integrity)

Ο περιορισμός ακεραιότητας οντοτήτων δηλώνει ότι καμία τιμή βασικού κλειδιού δεν μπορεί να είναι μηδενική. Αυτό είναι επειδή η τιμή βασικού κλειδιού χρησιμοποιείται για να προσδιορίσει μεμονωμένα πλειάδες σε μια σχέση. Η κατοχή της μηδενικής αξίας για το αρχικό κλειδί υπονοεί ότι δεν μπορούμε να προσδιορίσουμε σίγουρα κάποιες πλειάδες. Αυτό επίσης εξασφαλίζει ότι δεν μπορούν να υπάρξουν οποιεσδήποτε διπλές καταχωρήσεις στην αρχική βασική στήλη (column).

2.2. Αναφερόμενη ακεραιότητα (Referential Integrity)

Ο αναφερόμενος περιορισμός ακεραιότητας αφορά δύο συσχετίσεις και χρησιμοποιείται για να διατηρηθεί η συνέπεια μεταξύ πλειάδων στις δύο σχέσεις. Πιο απλά, ο αναφερόμενος περιορισμός ακεραιότητας δηλώνει ότι πλειάδα σε μια

συσχέτιση που αναφέρεται σε μια άλλη συσχέτιση πρέπει να αναφερθεί σε (reference to) πλειάδα σε αυτή τη άλλη συγκεκριμένη συσχέτιση.

2.3. Ακεραιότητα περιοχών (Domain Integrity)

Η ακεραιότητα περιοχών δηλώνει ότι κάθε στοιχείο από μια συσχέτιση πρέπει να ανταποκρίνεται στον τύπο και τους περιορισμούς των αντίστοιχων ιδιοτήτων του. Οι περιορισμοί θα μπορούσαν να είναι η πεδίο τιμών που το στοιχείο μπορεί να έχει, η προκαθορισμένη τιμή εάν καμία δεν παρέχεται, τότε η τιμή του στοιχείο μπορεί να είναι μηδενική(NULL).

2.4. Καθορισμένη από το χρήστη ακεραιότητα (User Defined Integrity)

π.χ.: Ηλικία \geq 18 && Ηλικία \leq 60

3. Ασφαλείας

- Οι διαρροές στα συστήματα βάσεων δεδομένων περιλαμβάνουν, παραδείγματος χάριν:
- Αναρμόδια ή απρομελέτητη δραστηριότητα ή κακή χρήση από τους εξουσιοδοτημένους χρήστες βάσεων δεδομένων ή και hacking.
- Μολύνσεις Malware που προκαλούν τα γεγονότα όπως η αναρμόδια πρόσβαση και διαρροή.
- Υπερφορτώσεις, περιορισμοί απόδοσης με συνέπεια την ανικανότητα των εξουσιοδοτημένων χρηστών να χρησιμοποιήσουν τις βάσεις δεδομένων όπως θα έπρεπε .
- Φυσική ζημία στους κεντρικούς υπολογιστές βάσεων δεδομένων. Σχεδιαστικά σφάλματα και bugs κατά των προγραμματισμό των ΒΔ
- Απώλεια δεδομένων που προκαλούνται από την είσοδο των άκυρων καταχωρήσεων ή λανθασμένων εντολών, π.χ. τα λάθη στις διαδικασίες διαχείρισης βάσεων δεδομένων κ.λπ.

4. Ανεξαρτησίας δεδομένων (δηλαδή πολλαπλά αρχεία και formats)

Η ανεξαρτησία στοιχείων είναι μια μορφή διαχείρισης βάσεων δεδομένων που κρατά τα στοιχεία χωρισμένα από όλα τα προγράμματα που χρησιμοποιούν τα στοιχεία. Σαν ακρογωνιαίο λίθο για την ιδέα ενός ΣΔΒΔ (Σύστημα Διαχείρισης Βάσεων Δεδομένων) η ανεξαρτησία δεδομένων εξασφαλίζει ότι τα στοιχεία δεν μπορούν να επαναπροσδιοριστούν ή να αναδιοργανωθούν από οποιαδήποτε από τα προγράμματα τα οποία τα χρησιμοποιούν. Κατά αυτόν τον τρόπο, το στοιχείο παραμένει προσπελάσιμο, αλλά είναι επίσης σταθερό και δεν μπορεί να αλλοιωθεί από τις εφαρμογές χρησιμοποιώντας το.

5. Προσπέλασης δεδομένων

6. Ενημέρωση πολλών αρχείων

Συνεπώς, στόχοι ώστε να επιτύχουμε μια ορθα σχεδιασμένη βάση δεδομένων πρέπει να είναι οι εξής:

1. Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
2. Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
3. Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.
4. Η επιβολή κανόνων ασφαλείας (security).
5. Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
6. Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

Και τα κατά σειρά **βήματα σχεδιασμού της**:

Ο σχεδιασμός για τη δημιουργία μίας βάσης δεδομένων περνάει από τρία βασικά στάδια , Αρχικά :

1. Ανάλυση των Απαιτήσεων:

Δηλαδή τι τύπου δεδομένα θέλουμε να αποθηκευτούν, ποιες διεργασίες θα πρέπει να γίνονται πάνω σε αυτά ή συχνότητα λειτουργιών που χρησιμοποιούνται. Κατόπιν,

2. Εννοιολογικός σχεδιασμός:

Ακριβής και υψηλού επιπέδου περιγραφή των δεδομένων που πρόκειται να αποθηκευτούν στη ΒΔ, ταυτόχρονη εμφάνιση των περιορισμών και σχέσεων μεταξύ δεδομένων (χρήση e-r μοντέλου). Τελικά,

3. Λογικός Σχεδιασμός:

Επιλογή ενός εργαλείου σχεδίασης βάσεων δεδομένων για την υλοποίηση του σχεδιασμού,

μετατροπή του εννοιολογικού σχεδιασμού σε ένα σχήμα στο μοντέλο δεδομένων του επιλεγμένου ΣΔΒΔ (σ.σ. ΣΔΒΔ : Σύστημα Διαχείρισης Βάσης Δεδομένων - Database Management System - είναι ένα πακέτο λογισμικού για τη συστηματική αποθήκευση -storage-, αναζήτηση -retrieval- και συντήρηση -maintenance- δεδομένων). [4] [1]

3.6 Σύνδεση στην Βάση Δεδομένων με την PHP

Πριν να έχουμε πρόσβαση στα δεδομένα σε μια βάση δεδομένων θα πρέπει να δημιουργήσουμε μια σύνδεση σε αυτήν [5]. Στην PHP, η εντολή είναι η συνάρτηση `mysql_connect()` και συντάσσεται ως εξής:

Σύνταξη

```
mysql_connect ( servername, username, password);
```

servername: Προαιρετικό. Καθορίζει το διακομιστή για να συνδεθούμε στην βάση. Η προ- επιλεγμένη τιμή είναι "localhost: 3306"

username: Προαιρετικό. Καθορίζει το όνομα χρήστη για να συνδεθούμε με την βάση. Προεπιλεγμένη τιμή είναι το όνομα του χρήστη στον οποίο ανήκει ο διακομιστής. **password :** Προαιρετικό. Καθορίζει τον κωδικό πρόσβασης για να συνδεθείτε με. Η προεπιλο- γή είναι ""

Στο παρακάτω παράδειγμα, αποθηκεύουμε τη σύνδεση σε μια μεταβλητή (\$ con). Το τμήμα "die" θα εκτελεστεί αν η σύνδεση αποτύχει:

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
?>
```

Το κλείσιμο μιας σύνδεσης

Η σύνδεση θα κλείσει αυτόματα όταν το script τελειώσει. Για να κλείσουμε τη σύνδεση, χρησιμοποιούμε την `mysql_close ()` λειτουργία:

```
<? php
$con = mysql_connect ("localhost", "root", "");
if (! $ con){
    die ("Δεν ήταν δυνατή η σύνδεση:". mysql_error ());
}
mysql_close ($ con);
?>
```

3.6.1 Εκτέλεση SQL Ερωτημάτων (Queries) με την PHP

Είδαμε πώς μπορούμε να συνδεθούμε σ' έναν database server της MySQL με το πρό- γραμμα `mysql` για να μπορέσουμε να γράψουμε ερωτήματα (queries) ή εντολές (commands) της SQL και να δούμε αμέσως τα αποτελέσματα των

ερωτημάτων. Στην PHP υπάρχει ένας παρόμοιος μηχανισμός, η συνάρτηση `mysql_query()`. [2]

```
mysql_query(<query>, <connection id>);
```

Όπου το `<query>` είναι ένα string που περιέχει τις εντολές της SQL που θα εκτελεσθούν. Όπως και με τη συνάρτηση `mysql_select_db()`, η παράμετρος `connection identifier` (α- ναγνωριστικό σύνδεσης) είναι προαιρετική. Η τιμή επιστροφής αυτής της συνάρτησης εξαρτάται από το είδος του ερωτήματος που στέλνεται. Για τις περισσότερες εντολές της SQL, η συνάρτηση `mysql_query()` επιστρέφει `true` ή `false` για να δείξει την επιτυχία ή την αποτυχία της αντίστοιχα.

Η συνάρτηση `mysql_error()` επιστρέφει ένα string κειμένου που περιγράφει το τελευταίο μήνυμα λάθους (error message) που στάλθηκε από τον MySQL server.

Για τα ερωτήματα DELETE, INSERT και UPDATE, με τα οποία μπορούμε να τροποποιήσουμε τα αποθηκευμένα δεδομένα, η MySQL κατά-γράφει επίσης τον αριθμό των γραμμών του πίνακα (καταχωρήσεων) που επηρεάστηκαν από το ερώτημα (query).

3.6.1.1 Εισαγωγή (insert) δεδομένων στην Βάση Δεδομένων με την PHP

Η εντολή για να εισάγουμε δεδομένα σε μια βάση δεδομένων αποκαλείται INSERT και υπάρχουν οι εξής δύο βασικές μορφές αυτής της εντολής [3] :

Σύνταξη

```
INSERT INTO table_name  
VALUES (value1, value2, value3,...)  
INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)
```

Το ακόλουθο παράδειγμα προσθέτει δύο νέες εγγραφές στον πίνακα "Πελάτες":

Παράδειγμα

```
<? php  
$ con = mysql_connect ("localhost", "root", "");  
if (! $ con){  
    die ("Δεν ήταν δυνατή η σύνδεση:". mysql_error ());  
}  
mysql_select_db ("my_db", $ con);  
  
mysql_query ("INSERT INTO πελάτες (Όνομα, Επώνυμο, Ηλικία) VALUES («Μαρία»,  
«Παπαδοπούλου», '35')");  
  
mysql_query ("INSERT INTO πελάτες (Όνομα, Επώνυμο, Ηλικία) VALUES («Γιώργος», «Μήτρου»,  
'33')");  
mysql_close ($con);  
?>
```

3.6.1.2 Επιλογή (select) δεδομένων στην Βάση Δεδομένων με την PHP

Η πρόταση SELECT χρησιμοποιείται για την επιλογή δεδομένων από μια βάση δεδομένων.

Σύνταξη

```
SELECT column_name (s) FROM table_name
```

Για να εκτελέσει η PHP την παραπάνω δήλωση πρέπει να χρησιμοποιήσουμε την `mysql_query ()` συνάρτηση. Αυτή η λειτουργία χρησιμοποιείται για να στείλει ένα ερώτημα ή την εντολή σε μια σύνδεση MySQL.

Το ακόλουθο παράδειγμα επιλέγει όλα τα δεδομένα που είναι αποθηκευμένα στον πίνακα "Πρόσωπα" (Ο χαρακτήρας * επιλέγει όλα τα στοιχεία στον πίνακα):

Παράδειγμα 1

```
<? Php
$ Con = mysql_connect ("localhost", "root", "");
if (! $ con){
    die ("Δεν ήταν δυνατή η σύνδεση:". mysql_error ());
}
mysql_select_db ("my_db", $ con);

$ result = mysql_query ("SELECT * FROM Πελάτες");
while ($row = mysql_fetch_array ($ result)) {
echo $row ["Όνομα"]. " ". $ row ['Επώνυμο']; echo "<br />";
}
mysql_close ($ con);
?>
```

Το παραπάνω παράδειγμα αποθηκεύει τα δεδομένα που επιστρέφονται από την `mysql_query()` συνάρτηση στην μεταβλητή `$ result`. Η συνάρτηση `mysql_fetch_array()` δέχεται ένα σύνολο αποτελεσμάτων (`result set`) σαν παράμετρο, που είναι αποθηκευμένο στη μεταβλητή `$result` στη συγκεκριμένη περίπτωση, και επιστρέφει την επόμενη γραμμή του `result set` σαν έναν πίνακα (`array`). Όταν δεν υπάρχουν άλλες γραμμές στο `result set`, η συνάρτηση `mysql_fetch_array()` επιστρέφει την τιμή `false`.

Η παραπάνω εντολή εκχωρεί μια τιμή στη μεταβλητή `$row`, αλλά ταυτόχρονα η ίδια η εντολή αποκτά την ίδια τιμή. Αυτός είναι ο λόγος που μπορούμε να χρησιμοποιήσουμε την εντολή για τη συνθήκη του βρόχου `while`.

Εφόσον οι βρόχοι `while` συνεχίζουν να εκτελούνται μέχρις ότου η συνθήκη τους γίνει ίση με `false`, ο βρόχος θα επαναληφθεί για όσο υπάρχουν γραμμές στο `result set`, με τη μεταβλητή `$row` να αποκτά κάθε φορά την τιμή της επόμενης γραμμής. Αυτό που έμεινε είναι να δούμε πώς μπορούμε κάθε φορά να παίρνουμε τις τιμές

της μεταβλητής \$row. Για να εκτυπώσουμε την τιμή της κάθε γραμμής, χρησιμοποιούμε την μεταβλητή \$row (\$row ["Όνομα"] και \$row ["Επώνυμο"]).

Η έξοδος του παραπάνω κώδικα θα είναι:

<i>Μαρία</i>	<i>Παπαδοπούλου</i>
<i>Γιώργος</i>	<i>Μήτρου</i>

Εμφάνιση του αποτελέσματος σε έναν πίνακα HTML

Το ακόλουθο παράδειγμα επιλέγει τα ίδια στοιχεία με το παραπάνω παράδειγμα, αλλά θα εμφανίζει τα δεδομένα σε πίνακα HTML:

```
Παράδειγμα 2
<? Php
$ Con = mysql_connect ("localhost", "root", "");
if (! $ con){
    die ("Δεν ήταν δυνατή η σύνδεση:". mysql_error ());
}
mysql_select_db ("my_db", $ con)?
$ result = mysql_query ("SELECT * FROM Πελάτες");
echo "<table border='1'> <tr>
<th>Όνομα </ th>
<th> Επώνυμο </ th>
</ Tr> ";
while ($ row = mysql_fetch_array ($ result)) {
echo "<tr>";
echo "<td>". $ row [«Όνομα»]. "</ Td>"? echo "<td>". $ row ["Επώνυμο"]. "</ Td>"? echo "</
tr>";
}
echo "</ table>";
mysql_close ($ con); ?>
```

Η έξοδος του παραπάνω κώδικα θα είναι:

Όνομα	Επώνυμο
<i>Μαρία</i>	<i>Παπαδοπούλου</i>
<i>Γιώργος</i>	<i>Μήτρου</i>

3.6.1.3 Ο όρος WHERE στην Βάση Δεδομένων με την PHP

Ο όρος WHERE χρησιμοποιείται για την εξαγωγή μόνο εκείνων των εγγραφών που πληρούν ένα συγκεκριμένο κριτήριο.

Σύνταξη

```
SELECT column_name (s) FROM table_name
```

Για να εκτελέσει η PHP την παραπάνω δήλωση πρέπει να χρησιμοποιήσουμε την mysql_query () συνάρτηση. Αυτή η λειτουργία χρησιμοποιείται για να στείλει ένα ερώτημα ή την εντολή σε μια σύνδεση MySQL.

Το ακόλουθο παράδειγμα επιλέγει όλες τις γραμμές από τον πίνακα "Πελάτες", όπου "Όνομα = 'Γιώργος':

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Πελάτες WHERE Όνομα = 'Γιώργος'");
while($row = mysql_fetch_array($result)) {
    echo $row["Όνομα"] . " " . $row["Επώνυμο"]; echo "<br />";
}
?>
```

Η έξοδος του παραπάνω κώδικα θα είναι:

```
Γιώργος    Μήτρου
```

3.6.1.4 Ο όρος ORDER BY στην Βάση Δεδομένων με την PHP

Αν θέλουμε να ταξινομήσουμε τις εγγραφές σε φθίνουσα ή αύξουσα σειρά, μπορούμε να χρησιμοποιήσουμε τη λέξη-κλειδί DESC ή ASC αντίστοιχα .

Σύνταξη

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s) ASC|DESC
```

Το ακόλουθο παράδειγμα επιλέγει όλα τα δεδομένα που είναι αποθηκευμένα στον πίνακα "Πελάτες", και ταξινομεί το αποτέλεσμα με βάση την στήλη "ηλικία":

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
```

```
if (!$con){
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Πελάτες ORDER BY Ηλικία");
while($row = mysql_fetch_array($result)) {
    echo $row['Όνομα'];
    echo " " . $row['Επώνυμο']; echo " " . $row['Ηλικία']; echo "<br />";
}
mysql_close($con); ?>
```

Η έξοδος του παραπάνω κώδικα θα είναι:

Γιώργος	Μήτρου	33
Μαρία	Παπαδοπούλου	35

Ταξινόμηση ανά δύο στήλες

Σύνταξη

```
SELECT column_name(s) FROM table_name
ORDER BY column1, column2
```

3.6.1.5 Ενημέρωση (update) δεδομένων στην Βάση Δεδομένων με την PHP

Η εντολή UPDATE χρησιμοποιείται για να ενημερώσουμε υπάρχουσες εγγραφές σε έναν πίνακα.

Σύνταξη

```
UPDATE table_name
SET column1=value, column2=value2 WHERE some_column=some_value
```

Το ακόλουθο παράδειγμα ενημερώνει ορισμένα στοιχεία στον πίνακα "Πελάτες":

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
mysql_query("UPDATE Πελάτες SET Age = '36' WHERE Όνομα = 'Γιώργος' AND Επώνυμο = 'Μήτρου'");
mysql_close($con);
?>
```

Η έξοδος του παραπάνω κώδικα θα είναι:

Όνομα	Επώνυμο	Ηλικία
Γιώργος	Μήτρου	36
Μαρία	Παπαδοπούλου	35

3.6.1.6 Διαγραφή (delete) δεδομένων από την Βάση Δεδομένων με την PHP

Η εντολή DELETE FROM χρησιμοποιείται για να διαγράψουμε εγγραφές από έναν πίνακα της βάσης δεδομένων. [3]

Σύνταξη

```
DELETE FROM table_name WHERE some_column = some_value
```

Το ακόλουθο παράδειγμα διαγράφει όλες τις εγγραφές του πίνακα "Πελάτες", όπου Επώνυμο = "Μήτρου":

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
mysql_query("DELETE FROM Πελάτες WHERE Επώνυμο = 'Μήτρου'");
mysql_close($con); ?>
```

Μετά τη διαγραφή, ο πίνακας θα μοιάζει κάπως έτσι:

Όνομα	Επώνυμο	Ηλικία
Μαρία	Παπαδοπούλου	35

3.6.2 Εκτέλεση SQL Ερωτημάτων (Queries) με συναρτήσεις συγκεντρωτικών αποτελεσμάτων (Aggregate Functions) με την PHP

3.6.2.1 MySQL GROUP BY

Ο όρος GROUP BY (ομαδοποίηση κατά δήλωση) χρησιμοποιείται σε συνδυασμό με τις συναρτήσεις συγκεντρωτικών αποτελεσμάτων (Aggregate Functions) σε μια ομάδα αποτελεσμάτων από μία ή περισσότερες στήλες. [3]

Σύνταξη

```
SELECT column_name, aggregate_function (column_name) FROM table_name
```

WHERE column_name operator value
GROUP BY column_name

3.6.2.2 MySQL Συνάρτηση - COUNT()

Η συνάρτηση count () μετράει τα στοιχεία ενός πίνακα, ή τις ιδιότητες ενός αντικειμένου.

Σύνταξη

SELECT COUNT(column_name) FROM table_name

Το ακόλουθο παράδειγμα μετράει πόσες εγγραφές υπάρχουν στον πίνακα "Πελάτες", ανά το ίδιο είδος εισιτηρίου :

Παράδειγμα

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$query = "SELECT type, COUNT(όνομα) FROM Πελάτες GROUP BY είδος εισιτηρίου";
while($row = mysql_fetch_array($result)){
    echo "Υπάρχουν " . $row['COUNT(όνομα)'] . " " . $row[' είδος εισιτηρίου ']. "
    εισιτήρια.";
    echo "<br />";
}
mysql_close($con);
?>
```

Ο πίνακας στην βάση είναι έτσι:

ΑΔΤ	όνομα	Είδος εισιτηρίου
1234	Νίκος	ΟΙΝΟΝΟΜΙΚΟ
8745	Μαρία	ΟΙΝΟΝΟΜΙΚΟ
4356	Γιώργος	ΦΟΙΤΗΤΙΚΟ

Το αποτέλεσμα θα είναι:

Υπάρχουν 2 ΟΙΝΟΝΟΜΙΚΟ εισιτήρια.

Υπάρχουν 1 ΦΟΙΤΗΤΙΚΟ εισιτήρια.

3.6.2.3 MySQL Συνάρτηση - SUM()

Η συνάρτηση SUM () επιστρέφει το συνολικό άθροισμα των τιμών μιας αριθμητικής στήλης.

Σύνταξη

```
SELECT SUM(column_name) FROM table_name
```

Το ακόλουθο παράδειγμα υπολογίζει το άθροισμα της στήλης Τιμή στον πίνακα "Πελάτες" με βάση το είδος του εισιτηρίου:

Παράδειγμα

```
<?php
$query = "SELECT type, SUM(τιμή) FROM products GROUP BY είδος εισιτηρίου";
$result = mysql_query($query) or die(mysql_error());
// Εκτύπωση αποτελέσματος
while($row = mysql_fetch_array($result)){
echo "Συνολικό κόστος ". $row['είδος εισιτηρίου']. " = ". $row['SUM(τιμή)'] ." ευρώ.";
echo "<br />";
}
?>
```

Ο πίνακας στην βάση είναι έτσι:

ΑΔΤ	όνομα	Είδος εισιτηρίου	Τιμή
1234	Νίκος	ΟΙΝΟΝΟΜΙΚΟ	34 ευρώ
8745	Μαρία	ΟΙΝΟΝΟΜΙΚΟ	26 ευρώ
4356	Γιώργος	ΦΟΙΤΗΤΙΚΟ	17 ευρώ

Το αποτέλεσμα θα είναι:

Συνολικό κόστος ΟΙΝΟΝΟΜΙΚΟ = 60 ευρώ.

Συνολικό κόστος ΦΟΙΤΗΤΙΚΟ = 17 ευρώ.

3.6.2.4 Άλλες MySQL συναρτήσεις - AVG(), MIN(), MAX()

- AVG (): Η συνάρτηση αυτή επιστρέφει τη μέση τιμή της μια αριθμητικής στήλης.

Σύνταξη

```
SELECT AVG (column_name) FROM table_name
```

- MIN (): Η συνάρτηση αυτή επιστρέφει την μικρότερη τιμή της επιλεγμένης στήλης.

Σύνταξη

```
SELECT MIN (column_name) FROM table_name
```

- MAX (): Η συνάρτηση αυτή επιστρέφει μικρότερη τιμή της επιλεγμένης στήλης.

Σύνταξη

```
SELECT MAX (column_name) FROM table_name
```

3.6.3 Εκτέλεση Προχωρημένων SQL Ερωτημάτων (Queries) με την PHP

3.6.3.1 MySQL DATE Συναρτήσεις

Ο παρακάτω πίνακας περιλαμβάνει τις σημαντικότερες συναρτήσεις για την ημερομηνία:

- NOW () : Επιστρέφει την τρέχουσα ημερομηνία και ώρα
- CURDATE () : Επιστρέφει την τρέχουσα ημερομηνία
- CURTIME () : Επιστρέφει την τρέχουσα ώρα
- DATE () : Αποσπάσματα από την πλευρά ημερομηνία μιας ημερομηνίας ή την ημερομηνία / ώρα
- EXTRACT () : Επιστρέφει ένα μόνο μέρος από μια ημερομηνία / ώρα
- DATE_ADD () : Προσθέτει ένα καθορισμένο χρονικό διάστημα σε μια ημερομηνία
- DATE_SUB () : Αφαιρεί ένα καθορισμένο χρονικό διάστημα από την ημερομηνία
- DATEDIFF () : Επιστρέφει τον αριθμό των ημερών μεταξύ δύο ημερομηνιών
- DATE_FORMAT () : Εμφανίζει την ημερομηνία / ώρα σε διαφορετικές μορφές

Το ακόλουθο παράδειγμα εισάγει στον πίνακα 'Κράτηση' την τρέχουσα ώρα στην στήλη dp_time και την τρέχουσα ημερομηνία στην στήλη dp_date:

Παράδειγμα 1

```
<?
$query_auto = "INSERT INTO Κράτηση (dp_time, dp_date)
VALUE (CURTIME (), CURDATE ())";
mysql_query ($query_manual) or die(mysql_error());
?>
```

Το ακόλουθο παράδειγμα επιλέγει τη στήλη Ημερομηνία στον πίνακα 'Δρομολόγιο' και αφαιρεί 1 ΜΕΡΑ. Άρα αν η ημερομηνία είναι 12-12-2011 θα την κάνει 11-12-2011. Αυτό γίνεται με τον παρακάτω κώδικα:

Παράδειγμα 2

```
<?  
// ΜΙΑ ΜΕΡΑ ΠΡΙΝ ΤΟ ΔΡΟΜΟΛΟΓΙΟ  
$sqlquery="SELECT DATE_SUB (ημερομηνία, INTERVAL 1 DAY) AS date FROM Δρομολόγιο";  
$result=mysql_query ($sqlquery, $conn) or die(mysql_error());  
>
```

ΕΠΙΛΟΓΟΣ

Στο επόμενο κεφάλαιο θα αναλυθεί η ιστοσελίδα με τις σελίδες τις καθώς και η χρήση της. Επιπρόσθετα θα αναλύεται ο τρόπος με τον οποίον γίνεται υλοποίηση της ιστοσελίδας.

ΚΕΦΑΛΑΙΟ 4

Υλοποίηση Ιστοσελίδας

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο να κάνουμε μια σύντομη περίληψη του έργου που φέρει το όνομα seekme.gr, και θα αναλύσουμε την υλοποίηση της πτυχιακής καθώς και τα σενάρια χρήσης της.

4.1 Ανάλυση προβλήματος

Για τις ανάγκες επίλυσης του προβλήματος μας χρειάζεται να μάθουμε κάποιες βασικές αρχές κάποιων βασικών γλωσσών προγραμματισμού, μελέτη για το περιβάλλον και δομή των τύπων για το πώς δομείται ένα προφίλ και ένα βιογραφικό.

Η βασική μας θεωρία για την υλοποίηση αυτής της ιστοσελίδας είναι το όσο το να καταφέρουμε να επιλύσουμε το πρόβλημα μας με απλό και κατανοητό τρόπο για όλους τους πιθανούς χρήστες.

Η τελική απαίτηση από την δημιουργία της ιστοσελίδας είναι να είναι φιλική προς τους χρήστες και να μπορεί να εξυπηρετήσει τις ανάγκες καταχώρησης ή αναζήτησης βιογραφικού.

4.1.1 Πλατφόρμα υλοποίησης

Η εφαρμογή κατά βάση είναι υλοποιημένη με τεχνολογίες Java και AJAX, κάνοντας χρήση XHTML, CSS καθώς και άλλων λοιπών τεχνολογιών αιχμής. PHP για την δημιουργία του δυναμικού περιεχομένου. MySQL ως συστημα διαχείρισης βάσεως δεδομένων και τη χρήση του Apache Server για το στησιμο της ιστοσελίδας και την εκτέλεση των αρχείων της PHP.

4.1.2 Θέματα ασφάλειας του χρήστη

Η ασφάλεια στο internet είναι ένα από τα μεγαλύτερα προβλήματα για τους χρήστες αλλά και τους κατασκευαστές ιστοσελίδων. Ακόμα και μεγάλα site όπου

έχουν μεγάλα budget και αριθμό προσωπικού έχουν πέσει θύματα επιθέσεων από hackers. Ένα λάθος που κάνουν αρκετοί developers, είναι ο σχεδιασμός μίας βάσης δεδομένων και ο τρόπος που αποθηκεύονται τα password των χρηστών.

Ας υποθέσουμε ότι κάνουμε εγγραφή σε ένα site και βάζω για username: johndoe και password asdf1234. Στην βάση δεδομένων αποθηκεύεται το username άλλα και το password σαν απλό κείμενο. Αυτό σημαίνει ότι εάν γίνει παραβίαση της βάσης δεδομένων κάποιος θα μπορέσει να δει όλους τους συνδυασμούς username/password και να παραβιάσει μετά όλα τα accounts του site. Επίσης αρκετοί χρήστες χρησιμοποιούν το ίδιο password σε όλα τα sites που μπαίνουν οπότε ο hacker μπορεί μετά να έχει πρόσβαση και στο email τους.

Για να αποφύγουμε αυτό το scenario θα πρέπει τουλάχιστον να χρησιμοποιήσουμε έναν αλγόριθμο encryption όπως το MD5 η το SHA1. Με encryption τα password αποθηκεύονται σε μορφή χαρακτήρων/αριθμών μέσω ενός μαθηματικού αλγόριθμου. πχ:

Το password asdf1234 με encryption θα αποθηκευτεί στην βάση δεδομένων ως εξής:

MD5: 1adbb3178591fd5bb0c248518f39bf6d

SHA1: f58cf5e7e10f195e21b553096d092c763ed18b0e
--

Με αυτόν τον τρόπο ασφαλίζουμε περισσότερο την βάση δεδομένων ενώ σε περίπτωση που παραβιαστεί η βάση θα είναι πολύ δύσκολο για κάποιον να παραβιάσει τα accounts χρηστών.

4.1.3 Προσωπική σελίδα του χρήστη

Με την εγγραφή του χρήστη καταχωρείται το μοναδικό username του στη βάση δεδομένων. Το username αυτό θα χρησιμοποιηθεί για την πρόσβαση στην σελίδα-βιογραφικό του κάθε χρήστη με την εξής χρήση:

Στο URL της ιστοσελίδας, για παράδειγμα στην *seekme.gr* προσθέτουμε ένα slash ("/") και στην συνέχεια το username του χρήστη. Τότε το URL γράφεται *seekme.gr/username*

Στη πραγματικότητα το URL γίνεται *seekme.gr/index.php?user=username* όπου περνάμε σαν παράμετρο το user στην σελίδα *index.php* και μας παρουσιάζει τις

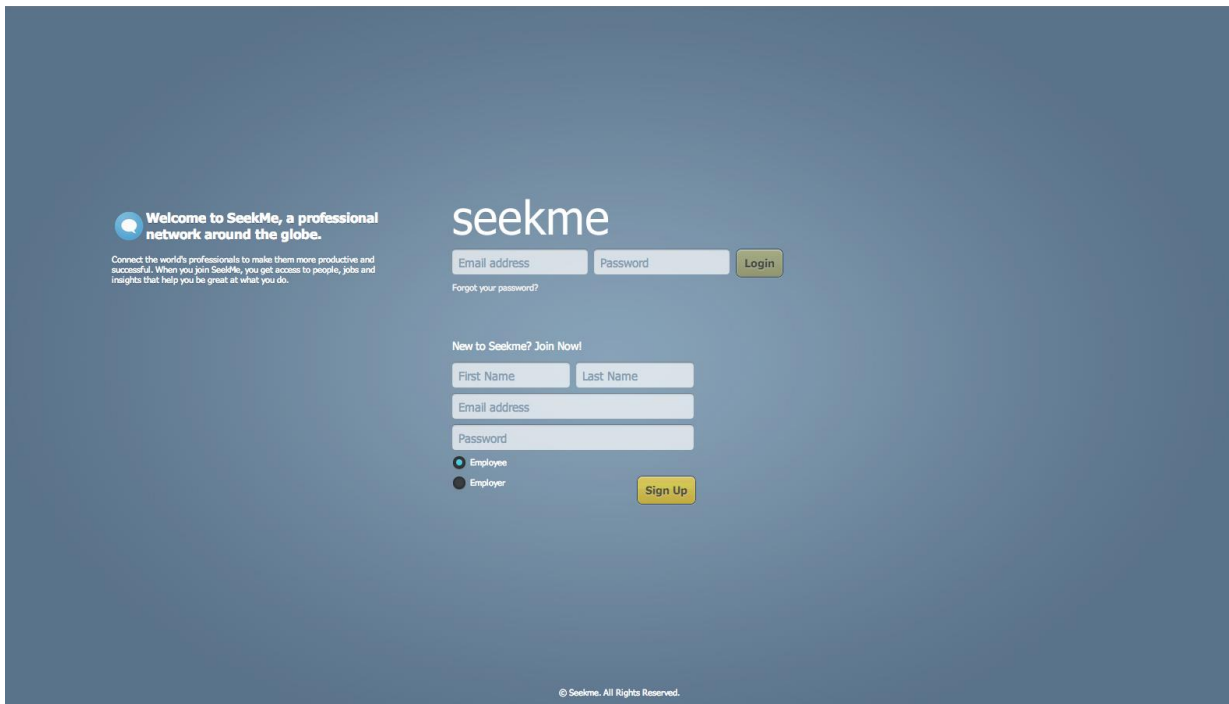
πληροφορίες του χρήστη. Αυτό επιτυγχάνεται με το αρχείο .htaccess και συγκεκριμένα με το RewriteEngine με τον παρακάτω κώδικα στο αρχείο:

```
RewriteEngine On # Ενεργοποίηση του mod_rewrite
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?user=$1 [QSA]
```

4.2 Γενική διευθέτηση ιστοτόπου

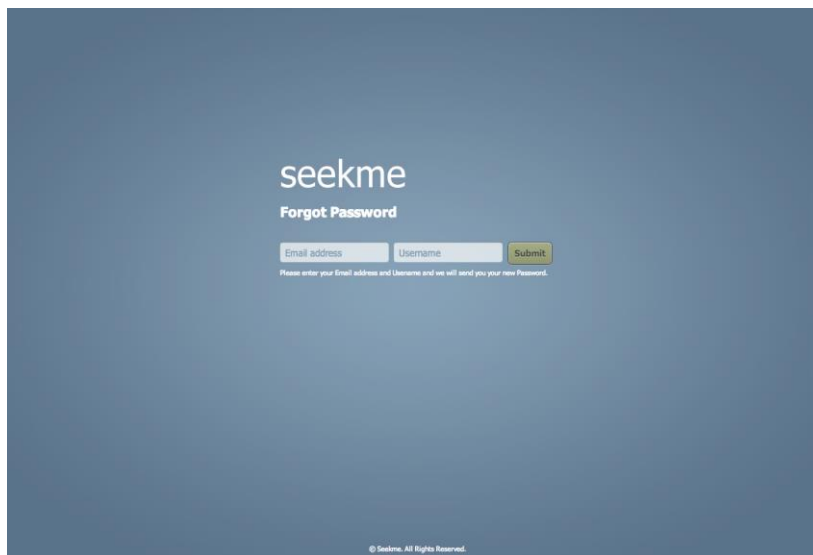
Έδω θα δούμε κάποιες από τις βασικές σελίδες του site μας όπως:

Κεντρική Σελίδα: Εδώ ο νέος χρήστης μπορεί να κάνει εγγραφή στην ιστοσελίδα μας για να μπορεί να συντάξει το βιογραφικό του. Επίσης ο ήδη εγγεγραμμένος χρήστης μπορεί να λάβει email για υπενθύμιση του κωδικού πρόσβασης του ή να συνδεθεί με την ιστοσελίδα.



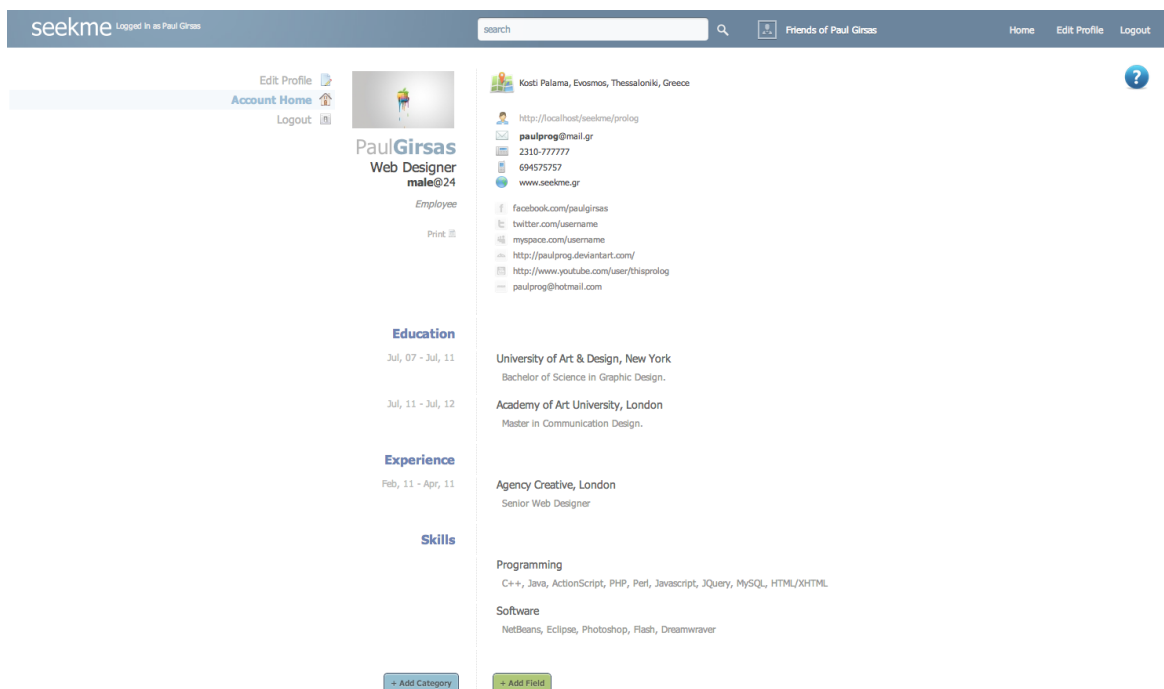
Εικόνα 11 «Κεντρική Σελίδα»

Σελίδα ανάκτησης κωδικού πρόσβασης: από την σελίδα αυτή ο χρήστης σε περίπτωση που έχει ξεχάσει τον κωδικό του συμπληρώνει το email και το username. Έτσι του αποστέλλεται ένας προσωρινός κωδικός με το οποίο μπορεί να εισέλθει στο σύστημα.



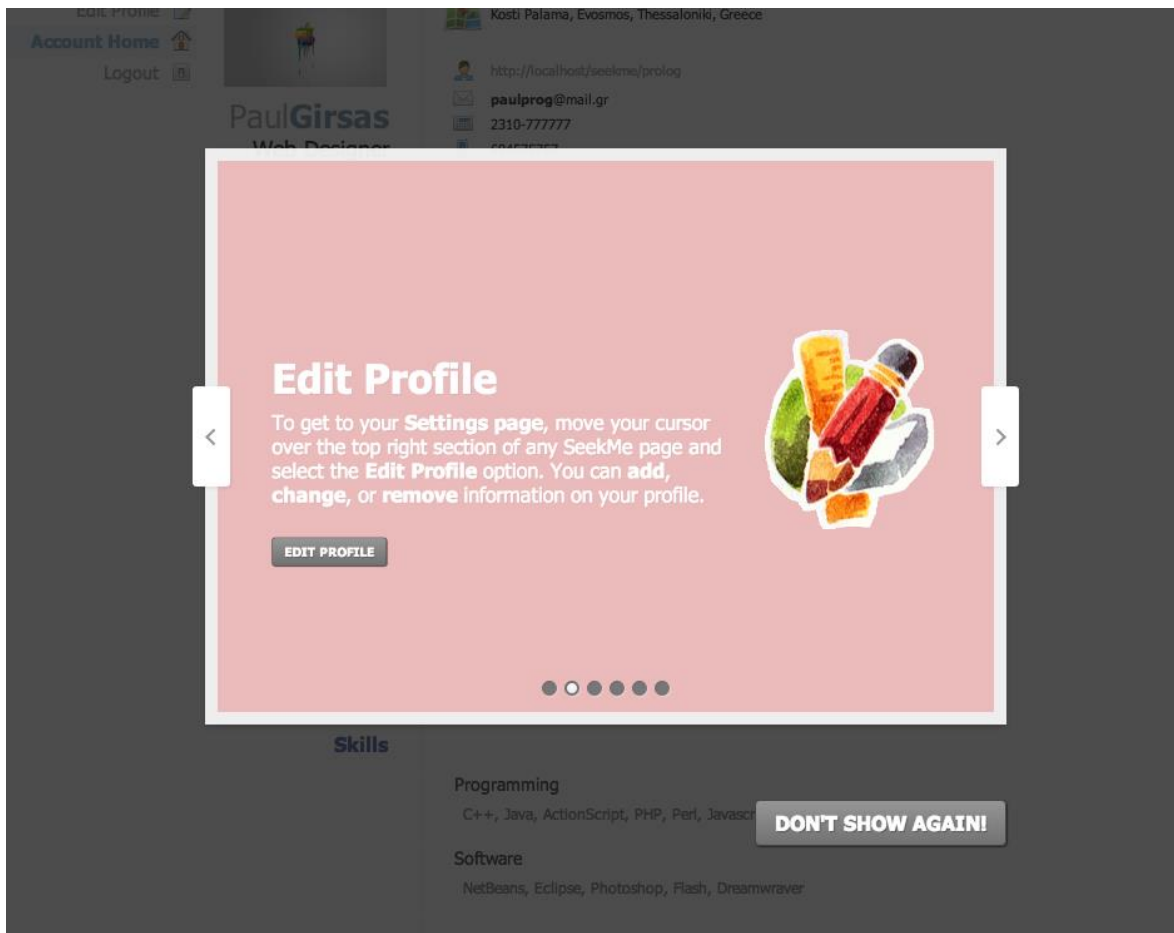
Εικόνα 12 «Σελίδα ανάκτησης κωδικού πρόσβασης»

Κεντρική Σελίδα του χρήστη: από αυτή την σελίδα ο κάθε ένας χρήστης στα αριστερά έχει το μενού που του δίνει τις επιλογές της διαχείρισης των προσωπικών του πληροφοριών καθώς και την επιλογή να εξέλθει από το σύστημα.



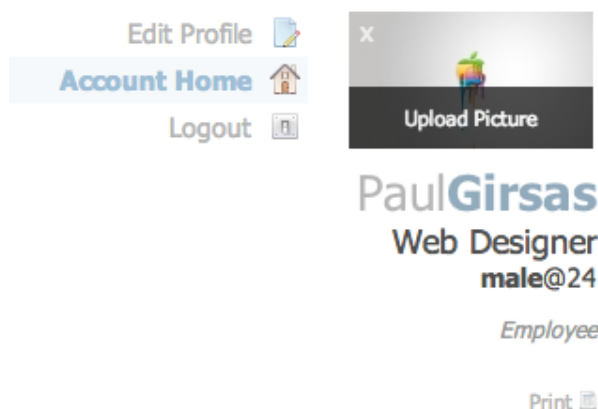
Εικόνα 13 «Κεντρική Σελίδα του χρήστη»

Με την εγγραφή του χρήστη και κατά τις πρώτες συνδέσεις στην σελίδα εμφανίζεται ένα παράθυρο εκμάθησης της διαδικασίας επεξεργασίας του προφίλ και διαχείρισης όλης της σελίδας γενικότερα. Ο χρήστης έχει την δυνατότητα να κλείσει το παράθυρο για πάντα ή να το κλείσει μόνο μια φορά και στην επόμενη είσοδο του να εμφανιστεί ξανά.



Εικόνα 14 «Tutorial»

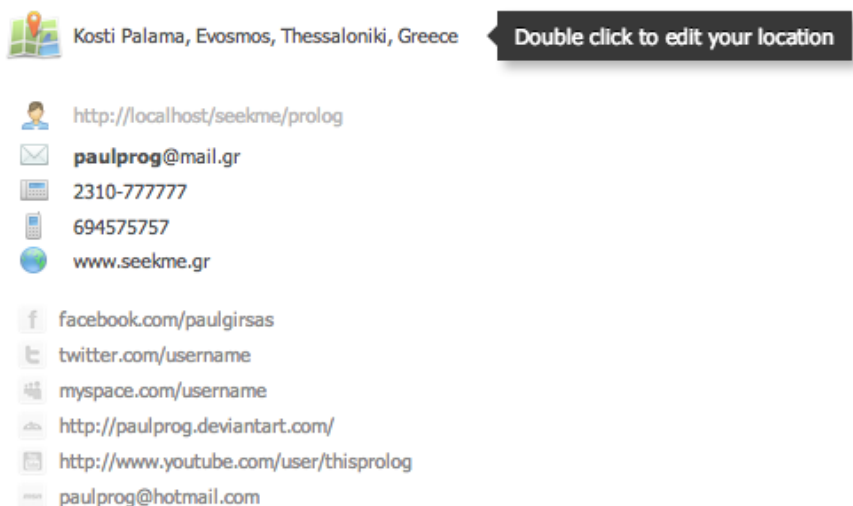
Δεξιά από το μενού ο χρήστης έχει την δυνατότητα να κάνει upload κάποια φωτογραφία αν το επιθυμεί ή να την διαγράψει. Επίσης στο κάτω μέρος βλέπουμε την επιλογή για εκτύπωση του προφίλ ή την αποθήκευση σε pdf μορφή.



Εικόνα 15 «Upload»

Στο δεξί μέρος της οθόνης παρουσιάζονται κάποιες βασικές πληροφορίες για τον χρήστη καθώς και στοιχεία επικοινωνίας μαζί με τα social network accounts του.

Ο χρήστης μπορεί επίσης να ανανεώσει την τοποθεσία του.



Εικόνα 16 «Βασικές πληροφορίες χρήστη»

Κάτω από της πληροφορίες του χρήστη αρχίζει το κύριο μέρος του βιογραφικού σημειώματος. Στο αριστερό μέρος της οθόνης βλέπουμε τις κατηγορίες των πεδίων που θα δούμε στα δεξιά. Από δεξιά βλέπουμε τις επικεφαλίδες του περιεχομένου του πεδίου καθώς και την ημερομηνίες σε περίπτωση που το επιθυμεί ο χρήστης και ακριβώς από κάτω το περιεχόμενο του πεδίου.

The screenshot shows a resume form with three main sections: Education, Experience, and Skills. The Education section lists two entries: 'University of Art & Design, New York' (Bachelor of Science in Graphic Design, Jul 07 - Jul 11) and 'Academy of Art University, London' (Master in Communication Design, Jul 11 - Jul 12). The Experience section lists 'Agency Creative, London' (Senior Web Designer, Feb 11 - Apr 11). The Skills section is divided into 'Programming' (C++, Java, ActionScript, PHP, Perl, Javascript, JQuery, MySQL, HTML/XHTML) and 'Software' (NetBeans, Eclipse, Photoshop, Flash, Dreamwaver). At the bottom, there are two buttons: '+ Add Category' and '+ Add Field'.

Εικόνα 17 «Μέρος βιογραφικού»

This screenshot shows a close-up of the Skills section. The 'Programming' section is visible with its list of languages. A modal dialog box is open for editing the 'Software' field. The dialog has a title bar 'Software', a text input field containing 'NetBeans, Eclipse, Photoshop, Flash, Dreamwaver', and 'Save' and 'Cancel' buttons at the bottom.

Εικόνα 18 «Επεξεργασία πεδίου»

Αξίζει να σημειωθεί πως ο χρήστης δεν είναι υποχρεωμένος να διαλέξει κάποιο έτοιμο τίτλο πεδίου, επικεφαλίδα ή περιεχόμενο ή ακόμα και να εισάγει πληροφορίες σε έτοιμα πεδία. Έχει την ελευθερία να εισάγει ότι πεδία ή επικεφαλίδες επιθυμεί χωρίς κάποιο περιορισμό. Απλά πατώντας το κουμπί Add Category για προσθήκη κατηγορίας και Add Field για προσθήκη πεδίου.

This screenshot shows the bottom part of the Skills section. The text 'NetBeans, Eclipse, Photoshop, Flash, Dreamwaver' is visible above two buttons: '+ Add Category' (blue) and '+ Add Field' (green).

Εικόνα 19 «Κουμπιά προσθήκης κατηγορίας και πεδίου»

Ο χρήστης μπορεί να σύρει όλη την σειρά και να βάλει μια προτεραιότητα στα πεδία του και στις κατηγορίες. Επίσης μπορεί να το διαγράψει αν πατήσει το εικονίδιο διαγραφή πάνω δεξιά ή να το επεξεργαστεί πατώντας το εικονίδιο επεξεργασία ή κάνοντας διπλό κλικ πάνω σε κάποιο πεδίο.

Στο κάτω μέρος της σελίδας βλέπουμε και το προσωπικό QR Code του χρήστη το οποίο μπορεί κανείς να σαρώσει με ένα QR Code Scanner και θα τον καθοδηγήσει στην σελίδα του χρήστη. Επίσης βλέπουμε και τα εικονίδια από τις social network σελίδες του χρήστη που είδαμε και πιο πάνω. Το κλικ κάνει μεταφορά στην αντίστοιχη σελίδα.



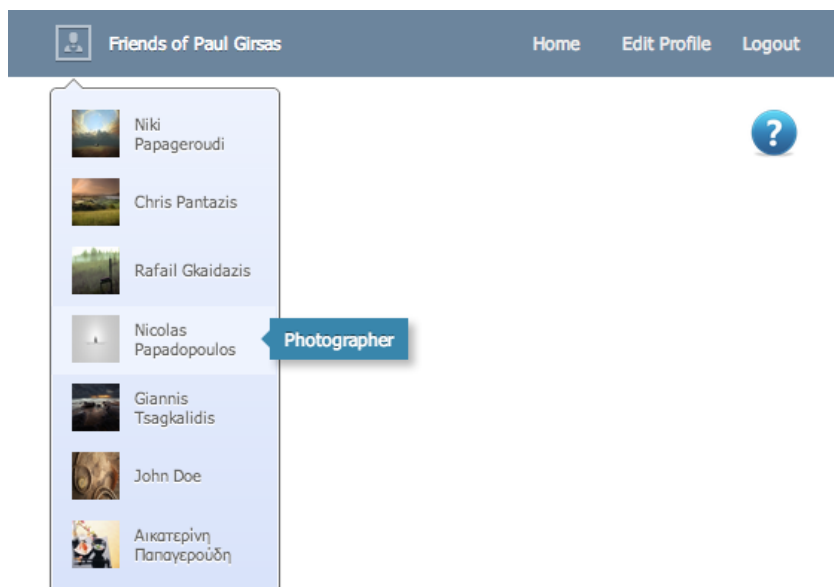
Copyright 2013 © Seekme. All rights reserved.
Use of this website signifies your agreement to the Terms of Use and Online Privacy Policy.

Εικόνα 20 «QR Code και Social Network Icons»

Για την παρακολούθηση άλλων χρηστών υπάρχει το κουμπί Follow στις σελίδες τους. Με το πάτημα ο χρήστης προστίθεται στην λίστα φίλων στο κεντρικό προφίλ του χρήστη.

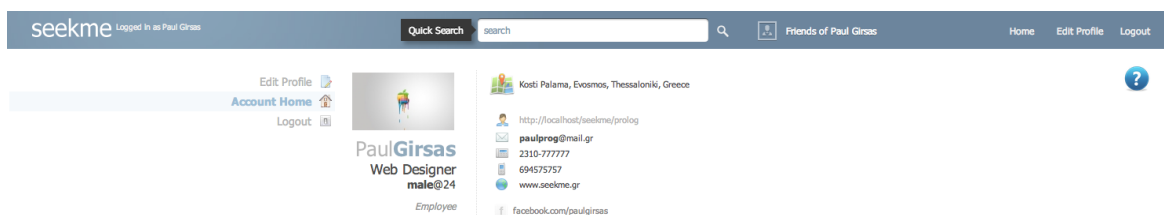


Εικόνα 21 «Follow»



Εικόνα 22 «Λίστα φίλων χρήστη»

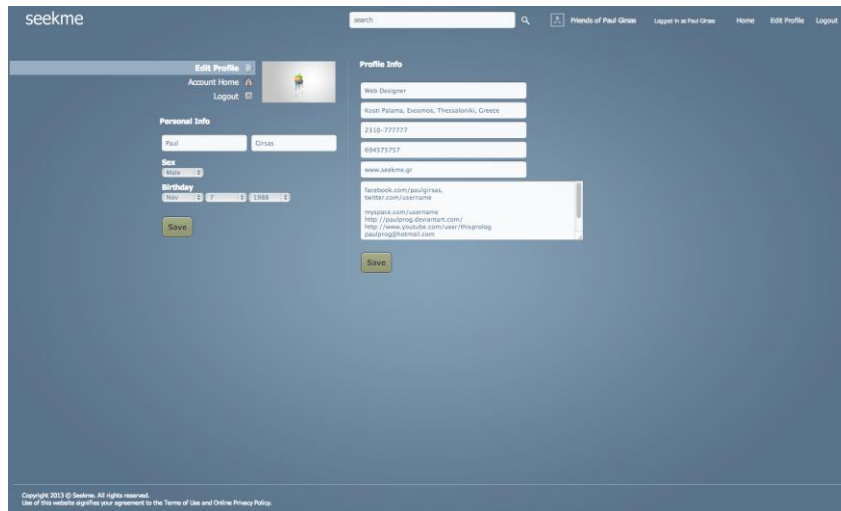
Στο πάνω μέρος της σελίδας βλέπουμε την κεντρική μπάρα από την οποία μπορούμε να κάνουμε αναζήτηση εύρεσης βιογραφικού ή εργοδότη.



Εικόνα 23 «Κεντρική μπάρα και αναζήτηση»

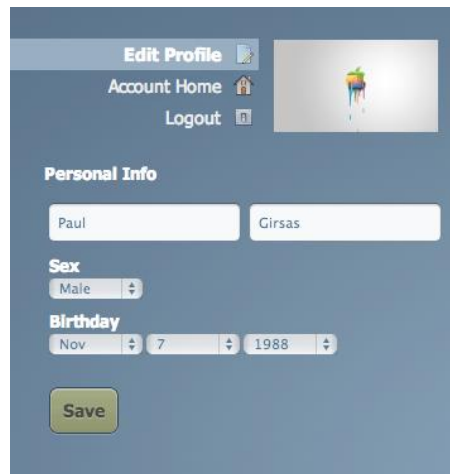
Σελίδα επεξεργασίας προσωπικών στοιχείων χρήστη: Εδώ ο χρήστης μπορεί να εισάγει βασικές πληροφορίες, να τις επεξεργαστεί ή να τις διαγράψει.

Πτυχιακή εργασία του φοιτητή Γκίρσα Παύλου



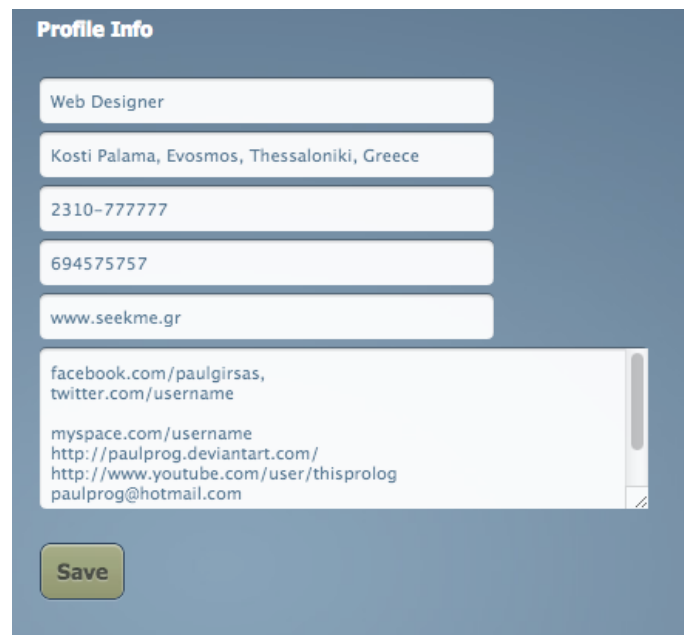
The screenshot shows the 'seekme' website's profile editing interface. At the top, there is a search bar and navigation links for 'Friends of Paul Girsas', 'Logout in as Paul Girsas', 'Home', 'Edit Profile', and 'Logout'. The main content area is divided into two sections: 'Personal Info' and 'Profile Info'. The 'Personal Info' section contains input fields for 'First Name' (Paul) and 'Last Name' (Girsas), a 'Sex' dropdown menu (set to Male), and a 'Birthday' section with dropdowns for month (Nov), day (7), and year (1988). A 'Save' button is located below these fields. The 'Profile Info' section contains input fields for 'Web Designer', 'Location' (Kosti Palama, Evosmos, Thessaloniki, Greece), 'Phone' (2310-777777), 'Mobile' (694575757), and 'Website' (www.seekme.gr). Below these are social media links for Facebook, Twitter, MySpace, DeviantArt, YouTube, and Email, with a 'Save' button at the bottom.

Εικόνα 24 «Σελίδα επεξεργασίας στοιχείων χρήστη»



This close-up screenshot focuses on the 'Personal Info' section of the profile editing page. It shows the 'Edit Profile' button, 'Account Home' link, and 'Logout' button. The 'Personal Info' section includes input fields for 'First Name' (Paul) and 'Last Name' (Girsas), a 'Sex' dropdown menu (set to Male), and a 'Birthday' section with dropdowns for month (Nov), day (7), and year (1988). A 'Save' button is positioned below the birthday fields.

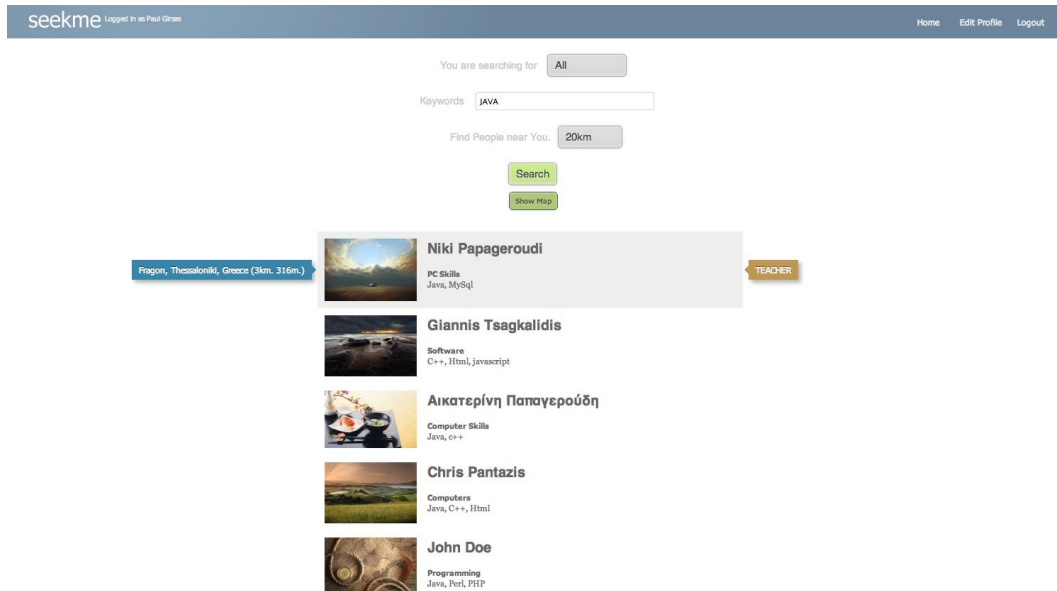
Εικόνα 25 «Επεξεργασίας προσωπικών στοιχείων χρήστη»



This close-up screenshot focuses on the 'Profile Info' section of the profile editing page. It shows the 'Profile Info' title and several input fields containing the user's details: 'Web Designer', 'Kosti Palama, Evosmos, Thessaloniki, Greece', '2310-777777', '694575757', and 'www.seekme.gr'. Below these are social media links for Facebook, Twitter, MySpace, DeviantArt, YouTube, and Email, with a 'Save' button at the bottom.

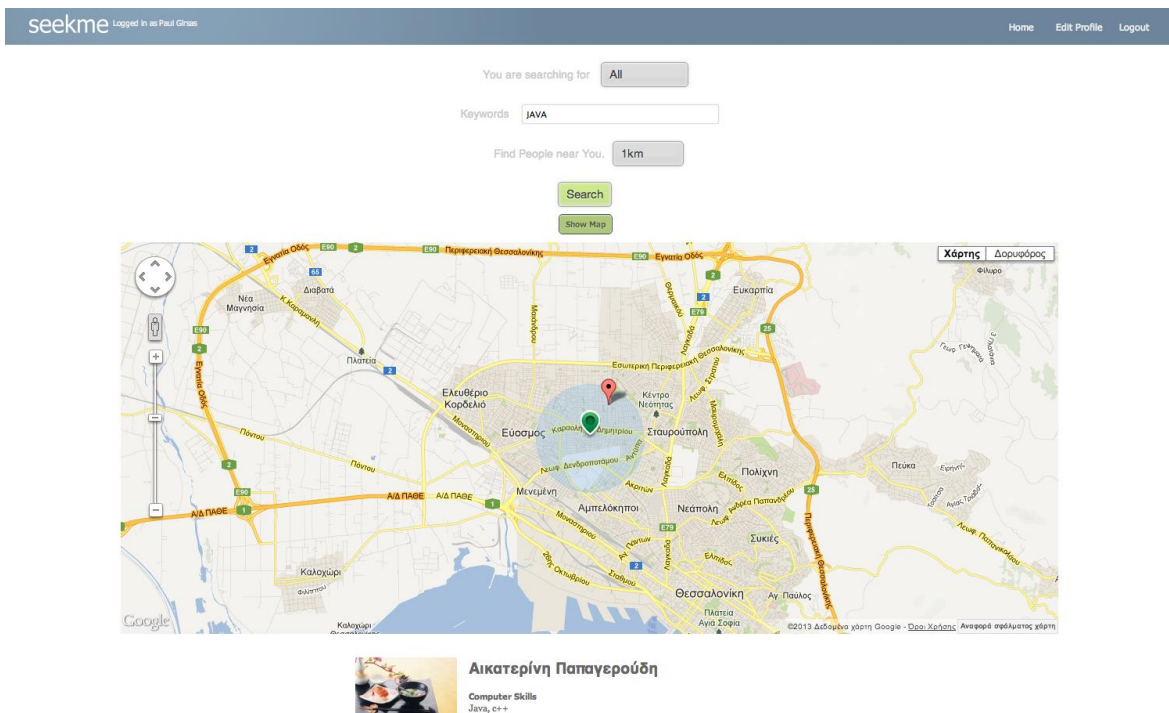
Εικόνα 26 «Στοιχεία επικοινωνίας χρήστη»

Σελίδα αναζήτησης: Εδώ ο χρήστης μπορεί να ψάξει στη βάση δεδομένων για προφίλ. Υπάρχει φίλτρο επιλογής για αναζήτηση εργαζομένων ή εργοδότη. Το κλικ πάνω σε ένα αποτέλεσμα ανοίγει το προφίλ του αντίστοιχου χρήστη.



Εικόνα 27 «Αναζήτηση»

Επίσης ο χρήστης έχει την δυνατότητα να δει τα αποτελέσματα της αναζήτησης ως σημεία στον χάρτη με το πάτημα του κουμπιού **Show Map**



Εικόνα 28 «Αναζήτηση με χάρτη»

4.3 Βάση Δεδομένων Εφαρμογής

4.3.1 Πίνακας Users

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 User_ID	int(11)			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2 Username	varchar(150)	utf8_bin		No	None		
<input type="checkbox"/>	3 Name	varchar(32)	utf8_bin		No	None		
<input type="checkbox"/>	4 Lastname	varchar(50)	utf8_bin		No	None		
<input type="checkbox"/>	5 Birthday	varchar(12)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	6 Sex	int(1)			No	None		
<input type="checkbox"/>	7 Profession	varchar(255)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	8 Location	varchar(255)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	9 Ion	float			No	None		
<input type="checkbox"/>	10 lat	float			No	None		
<input type="checkbox"/>	11 Phone	varchar(50)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	12 Mobile	varchar(50)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	13 Website	varchar(50)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/>	14 Contacts	varchar(255)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	15 Username_Clean	varchar(150)	utf8_bin		No	None		
<input type="checkbox"/>	16 Password	varchar(225)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	17 Email	varchar(150)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	18 Pic	tinyint(4)			Yes	NULL		
<input type="checkbox"/>	19 ActivationToken	varchar(225)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	20 LastActivationRequest	int(11)			No	None		
<input type="checkbox"/>	21 LostPasswordRequest	int(1)			No	0		
<input type="checkbox"/>	22 Active	int(1)			No	None		
<input type="checkbox"/>	23 Group_ID	int(11)			No	None		
<input type="checkbox"/>	24 SignUpDate	int(11)			No	None		
<input type="checkbox"/>	25 LastSignIn	int(11)			No	None		
<input type="checkbox"/>	26 type	int(1)			No	None		
<input type="checkbox"/>	27 help	tinyint(1)			No	1		

Εικόνα 29 «Πίνακας Users»

Ο πίνακας Users περιέχει όλους τους χρήστες με τις βασικές τους πληροφορίες όπως στοιχεία επικοινωνίας, τοποθεσία, ημερομηνία γέννησης κ.τ.λ. Περιέχει έναν μοναδικό αριθμό **User_ID** που αντιστοιχεί στον χρήστη αυτόν και μόνο. Επίσης περιέχει πεδίο που προσδιορίζει αν ο χρήστης είναι εργαζόμενος η εργοδότης.

4.3.2 Πίνακας Fields

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 ID	int(11)			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2 User_ID	int(11)			No	None		
<input type="checkbox"/>	3 Title	text	utf8_bin		No	None		
<input type="checkbox"/>	4 Content	text	utf8_bin		No	None		
<input type="checkbox"/>	5 Date	varchar(32)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	6 Dateto	varchar(32)	utf8_bin		Yes	NULL		
<input type="checkbox"/>	7 Pos	int(3)			Yes	NULL		

Εικόνα 30 «Πίνακας Fields»

Ο πίνακας Fields περιέχει τα πεδία των χρηστών και περιέχει και αυτός ένα μοναδικό κλειδί **ID**. Συνδέεται με τον χρήστη με το **User_ID** του και περιέχει τον τίτλο, το περιεχόμενο καθώς και τις 2 ημερομηνίες του πεδίου, δηλαδή τις «από» και «έως». Ένα ακόμα σημαντικό πεδίο στον πίνακα είναι το position, είναι η σειρά με την οποία εμφανίζονται τα πεδία κατά την εμφάνιση του προφίλ του χρήστη.

4.3.3 Πίνακας Fields

Browse Structure SQL Search Insert Export Import								
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT	Change Drop
<input type="checkbox"/>	2 userid	int(10)		UNSIGNED	No	None		Change Drop
<input type="checkbox"/>	3 fid	int(10)		UNSIGNED	No	None		Change Drop

Εικόνα 31 «Πίνακας Friends»

Ο πίνακας Friends περιέχει τους φίλους ενός χρήστη. Ο χρήστης καταγράφεται με το **userid** ενώ ο φίλος θα καταγράφεται με το **fid**.

4.4 Ανάλυση Κώδικα Εφαρμογής

Εδώ θα αναλύσουμε κάποια βασικά σημεία του κώδικα της ιστοσελίδας.

settings.php

```
//Database Information
$dbtype = "mysql";
$db_host = "localhost";
$db_user = "root";
$db_pass = "12345";
$db_name = "seekme";
$db_port = "";
$db_table_prefix = "";
$langauge = "en";
```

Στο αρχείο settings.php έχουμε τις βασικές πληροφορίες της σελίδας καθώς και στοιχεία για την σύνδεση με βάση δεδομένων.

class.user.php

Το class.user.php αποτελεί την βασική κλάση που αποτελεί τον εγγεγραμμένο χρήστη και διευκολύνει όλη την διαδικασία διαχείρισής του. Η κλάση περιέχει όλα τα πεδία όπως το username το id το ονοματεπώνυμο του χρήστη και λοιπά στοιχεία. Επίσης διαθέτει όλες τις μεθόδους που χρειάζονται.

```
//function to update personal info
public function updatePersonal($firstname="", $lastname="", $birthday="", $sex="") {
    global $db, $db_table_prefix;
    $this->firstname= $firstname;
    $this->lastname = $lastname;
    $this->fullname = $firstname." ".$lastname;
    $this->birthday = $birthday;
    $this->sex = $sex;
    $sql = "UPDATE ``.$db_table_prefix.`Users` SET
        Name=" . $db->sql_escape($this->firstname) . ",
        Lastname=" . $db->sql_escape($this->lastname) . ",
        Birthday=" . $db->sql_escape($this->birthday) . ",
        Sex=" . $db->sql_escape($this->sex) . "
        WHERE User_ID = " . $db->sql_escape($this->user_id) . """;
    return ($db->sql_query($sql));
}
```

για παράδειγμα η μέθοδος `updatePersonal` καλείται μέσα από την σελίδα επεξεργασίας προφίλ με την τεχνική Ajax και ενημερώνει τις προσωπικές πληροφορίες του χρήστη και στην συνέχεια τις αποθηκεύει στην βάση δεδομένων.

```
//Simple function to update the last sign in of a user
public function updateLastSignIn() {
global $db,$db_table_prefix;
$sql = "UPDATE ".$db_table_prefix."Users
      SET LastSignIn = '".time()."'
      WHERE
      User_ID = '".$db->sql_escape($this->user_id)."'";
return ($db->sql_query($sql));
}
```

Η παραπάνω μέθοδος για παράδειγμα ενημερώνει τον πίνακα `Users` και συγκεκριμένα ενημερώνει το πεδίο `LastSignIn` με την τρέχουσα ώρα (timestamp) κάθε φορά που ο χρήστης εισέρχεται στο σύστημα για να έχουμε την ώρα της τελευταίας εισόδου.

Υπάρχει ένας αριθμός μεθόδων για την ενημέρωση κάποιων πεδίων του πίνακα του χρήστη. Για παράδειγμα παρακάτω βλέπουμε μια μέθοδο ενημέρωσης και αλλαγής του email του χρήστη.

```
//Update a users email
public function updateEmail($email) {
global $db,$db_table_prefix;
$this->email = $email;
$sql = "UPDATE ".$db_table_prefix."Users
      SET Email = '".$email."'
      WHERE
      User_ID = '".$db->sql_escape($this->user_id)."'";
return ($db->sql_query($sql));
}
```

Παρακάτω βλέπουμε την μέθοδο με την οποία προστίθεται η αφαιρείται αντίστοιχα ένας χρήστης στον πίνακα φίλων πολύ απλά μόνο με δυο παραμέτρους, την `userid` και την `id φίλου (fid)`.

```
//Update a users fiends list
public function addFriend($userid, $fid) {
global $db,$db_table_prefix;
$db->sql_query($sql);
}
```

```
$sql = "INSERT INTO friends (userid, fid)
      VALUES (".$userid.", ".$fid.)";
return ($db->sql_query($sql));
}
//Remove friend
public function removeFriend($userid, $fid) {
global $db,$db_table_prefix;
$db->sql_query($sql);
$sql = "DELETE FROM friends WHERE userid = ".$userid." AND fid = ".$fid.";";
return ($db->sql_query($sql));
}
```

Επίσης μια σημαντική μέθοδος είναι η `getFriends` η οποία επιστρέφει τους φίλους του χρήστη.

```
public function getFriends() {
global $db,$db_table_prefix;
$sql = "SELECT
      fid
      FROM
      ".$db_table_prefix."friends
      WHERE userid = '". $db->sql_escape($this->user_id)."'";
$result = mysql_query($sql);
$row = mysql_fetch_array($result);
return($row);
}
```

func.user.php

Η `func.user.php` είναι ένα `php` αρχείο με διάφορες γενικές μεθόδους που χρειάζονται για την σωστή λειτουργία του χρήστη.

Ένα παράδειγμα τέτοιας μεθόδου είναι η `usernameExists` που πραγματοποιείται κατά την εγγραφή του χρήστη και ελέγχει αν υπάρχει ήδη εγγεγραμμένος χρήστης με το ίδιο `username`.

```
function usernameExists($username) {
global $db,$db_table_prefix;
$sql = "SELECT Active
      FROM ".$db_table_prefix."Users
      WHERE Username_Clean = '". $db->sql_escape(sanitize($username))."'
      LIMIT 1";
if(returns_result($sql) > 0)
return true;
else
return false;
}
```

Παρόμοια μέθοδος είναι και η `emailExists` η οποία ελέγχει αν υπάρχει το email αυτό στην βάση.

```
function emailExists($email) {
global $db,$db_table_prefix;
$sql = "SELECT Active FROM ".$db_table_prefix."Users
WHERE
Email = ".$db->sql_escape(sanitize($email))."
LIMIT 1";
if(returns_result($sql) > 0)
return true;
else
return false;
}
```

login.php

Εδώ θα αναλύσουμε μερικά βασικά μέρη του κώδικα που αξίζει να δώσουμε σημασία. Η σελίδα `login.php` περιέχει την φόρμα για είσοδο του χρήστη καθώς και για την εγγραφή του.

```
require_once("class.user.php");
require_once("class.mail.php");
require_once("funcs.user.php");
require_once("funcs.general.php");

session_start();

//Global User Object Var
//loggedInUser can be used globally
if(isset($_SESSION["userSeekmeUser"]) && is_object($_SESSION["userSeekmeUser"])) {
    $loggedInUser = $_SESSION["userCakeUser"];
}
```

Στο συγκεκριμένο κομμάτι κώδικα βλέπουμε την φόρτωση των απαραίτητων αρχείων και κλάσεων που αναλύσαμε παραπάνω για την σωστή λειτουργία της σελίδας. Στην συνέχεια με την εντολή `session_start()` αρχίζουμε ένα Session (την οποία θα καταστρέψουμε κατά το `logout`). Τέλος ελέγχουμε αν έχουμε ήδη θέσει στο Session μας τον χρήστη, αν όχι τον θέτουμε.

```
//Prevent the user visiting the logged in page if he/she is already logged in
if(isUserLoggedIn()) {
    header("Location: " . "http://" . $_SERVER['HTTP_HOST'] . $location); die();
}
```

Εδώ ελέγχουμε για το αν ο χρήστης έχει ήδη εισέλθει, και αν ναι, τότε μεταφέρεται στην κεντρική σελίδα.

Παράδειγμα κώδικα HTML της φόρμας για εγγραφή του χρήστη

```
<form id="regForm" action="submit.php" method="post">
<div id="join">New to Seekme? Join Now!</div></br>
<div class="holding" style="float:left">
    <label for="fname">First Name</label> <input name="fname" id="fname" type="text"
    style="width:137px;"/>
</div>
<div class="holding" style="float:left;">
    <label for="lname">Last Name</label>
    <input name="lname" id="lname" type="text" style="width:137px;"/>
</div>
<div class="holding" style="clear:both;">
    <label for="regemail">Email address</label>
    <input name="email" id="regemail" type="text" />
</div>
<div class="holding">
    <label for="pass">Password</label>
    <input name="pass" id="pass" type="password" />
</div>
<p> <input id="r1" type="radio" name="rb" checked="checked" value="0">
<label for="r1">Employee</label>
</p> </br> </br>
<p>
<input id="r2" type="radio" name="rb" value="1">
<label for="r2">Employer</label> </p>
<button type="submit" ><span>Sign Up</span></button>

</form>
```

Κατά το πάτημα του κουμπιού Login αντί να γίνει το submit τρέχουμε χωρίς να ξαναφορτώσουμε την σελίδα τον έλεγχο και όλες τις διαδικασίες που χειριζόμαστε. Αυτό το πετυχαίνουμε με την JQuery και την τεχνολογία Ajax με τον εξής τρόπο.

```
$(document).ready(function(){
    $('#regForm').submit(function(e) {
        register();
        e.preventDefault();
    });
});
function register() {
hideshow('loading',1);
error(0);
$.ajax({
    type: "POST",
    url: "regsubmit.php",
    data: $('#regForm').serialize(),
    dataType: "json",
    success: function(msg){
        if(parseInt(msg.status)==1) {
            //error(1,'Your registration was successful!);
            //window.location=msg.txt;
```

```
        window.location='login.php?success';
    } else if(parseInt(msg.status)==0) {
        error(1,msg.txt);
    }
    hideshow('loading',0);
}
});
}
```

Αυτό που κάνουμε ουσιαστικά είναι να αποτρέψουμε το κουμπί να κάνει το submit με την μέθοδο preventDefault και στην συνέχεια τρέχουμε την μέθοδο register. Η μέθοδος αυτή δείχνει την εικόνα της φόρτωσης στον χρήστη και αμέσως με την μέθοδο ajax στέλνει όλα τα δεδομένα της φόρμας στο αρχείο regssubmit.php το οποίο με πολύ απλές διαδικασίες κάνει την εγγραφή του χρήστη στην βάση.

Σε περίπτωση επιτυχίας έχουμε το αντίστοιχο μήνυμα και την εξαφάνιση της εικόνας φόρτωσης. Στην περίπτωση που αποτύχουμε μας παρουσιάζεται το μήνυμα με το πρόβλημα που αντιμετωπίστηκε.

index.php

Η σελίδα αυτή αντλεί της πληροφορίες από την βάση δεδομένων για τον χρήστη και της παρουσιάζει. Εδώ αξίζει να αναλυθεί ο μηχανισμός δημιουργίας των πεδίων της σελίδας προφίλ καθώς και η προβολή τους.

Το πεδίο με όλα τα στοιχεία του αποτελείται από την κλάση comment.php. Μέσα στην κλάση αυτήν υπάρχουν όλες οι μέθοδοι που χρειάζονται για την δημιουργία ενός πεδίου, την διαγραφή του, την τροποποίησή, την αλλαγή θέσης καθώς και την προβολή τους για κάποιον χρήστη.

Η παρακάτω μέθοδος παρουσιάζει όλα τα πεδία που έχουν δημιουργηθεί από τον χρήστη.

```
public static function find_fields_on($user_id=0) {
    global $db;
    $sql = "SELECT * FROM " . self::$table_name;
    $sql .= " WHERE User_ID=" . $db->sql_escape($user_id);
    $sql .= " ORDER BY Pos ASC";
    return self::find_by_sql($sql);
}
```

Και το παράδειγμα με την δημιουργία του πεδίου.

```
public function create() {
    global $db;
    $attributes = $this->sanitized_attributes();
    array_shift($attributes);
    $sql = "INSERT INTO " . self::$table_name . "(";
    $sql .= join(", ", array_keys($attributes));
```

```
$sql .= ") VALUES ('";  
$sql .= join("'", "", array_values($attributes));  
$sql .= "'");  
if($db->query($sql)) {  
    $this->ID = $db->insert_id();  
    return $this->ID;  
} else {  
    return false;  
}  
}
```

Αξίζει να σημειωθεί πως χρησιμοποιούμε το `sql_escape` κάθε φορά για να αποφύγουμε το πρόβλημα του `sql injection`.

search.php

Στην σελίδα `search.php` κάνουμε αναζήτηση με βάση κάποια λέξη κλειδί και γύρω από μια περιοχή εφόσον το επιθυμούμε.

Ένα σημαντικό στοιχείο της σελίδας αυτής είναι ο κώδικας SQL που χρησιμοποιούμε για να αντλήσουμε τις πληροφορίες που χρειαζόμαστε από την βάση μας.

Στο πάνω μέρος της σελίδας υπάρχει ένα στοιχείο `select` που λειτουργεί σαν φίλτρο και μας παρουσιάζει τους εργαζομένους, τους εργοδότες ή και τους 2 μαζί.

```
$("#filter").change(function () {  
    var str = "";  
    $("#filter option:selected").each(function () {  
        str = $(this).text();  
    });  
    if(str == 'Employee'){  
        $('li').filter('.Employer').slideUp(300);  
        $('li').filter('.Employee').slideDown(300);  
    } else if(str == 'Employer'){  
        $('li').filter('.Employee').slideUp(300);  
        $('li').filter('.Employer').slideDown(300);  
    } else {  
        $('li').slideDown(300);  
    }  
});
```

Παραπάνω βλέπουμε τον κώδικα JQuery για την περίπτωση που κάποιος επιλέξει κάτι από το φίλτρο. Για παράδειγμα αν επιλέξει κάποιος την επιλογή “Employee” τότε όλα τα πεδία (li) της λίστας που περιέχουν την κλάση Employer θα κάνουν `slideUp()` και θα εξαφανιστούν ενώ αυτά που περιέχουν την κλάση Employee την `slideDown()` και θα εμφανιστούν στην οθόνη.

Στην συνέχεια οι γίνεται η αναζήτηση με τον εξής τρόπο:

```
$query = "SELECT * FROM `lv_fields` WHERE";
$array = explode(" ", $search);
foreach ($array as $key => $keyword){
    $query .= " (upper(`Content`) LIKE '%$keyword%') OR (upper(`Title`) LIKE '%$keyword%')";
    if ($key != (sizeof($array) - 1)) //skip adding the last 'AND' to
        $query .= " AND ";           // prevent a bad query
}
}
```

Αυτό που κάνουμε είναι να χωρίσουμε της λέξεις κλειδιά με το κενό " " και τις τοποθετούμε σε ένα array. Στην συνέχεια για κάθε στοιχείο του πίνακα (λέξη κλειδί) δημιουργούμε ένα query string και το κολλάμε στο προηγούμενο με την AND. Στο τελευταίο στοιχείο παραβλέπουμε την πρόσθεση του AND για πιο σωστό query. Τέλος κάνουμε αναζήτηση στον πίνακα `lv_fields` με το query που δημιουργήσαμε προηγουμένως.

Εξίσου σημαντικό μέρος της σελίδας αυτής αποτελεί την αναζήτηση σύμφωνα με την ακτίνα γύρω από την τοποθεσία του χρήστη.

Για να βρούμε τους χρήστες βρίσκονται σε μια συγκεκριμένη ακτίνα απόστασης ενός συγκεκριμένου γεωγραφικού πλάτους / μήκους, χρησιμοποιούμε μια SELECT με βάση τον τύπο Haversine. Ο τύπος Haversine χρησιμοποιείται γενικά για τον υπολογισμό αποστάσεις ανάμεσα σε δύο ζεύγη συντεταγμένων σε μια σφαίρα.

Παρακάτω βλέπουμε τον κώδικα SQL που βρίσκει τις πλησιέστερες 20 τοποθεσίες που βρίσκονται σε ακτίνα 25 χιλιόμετρα για της συντεταγμένες 37, -122. Υπολογίζει την απόσταση με βάση το γεωγραφικό πλάτος / γεωγραφικό μήκος αυτής της γραμμής και του στόχου γεωγραφικού πλάτους / μήκους, και στη συνέχεια επιστέφει μόνο τις γραμμές όπου η τιμή «distance» είναι μικρότερη από 25, τέλος την περιορίζει σε 20 αποτελέσματα.

```
SELECT id, (6371* acos( cos( radians(37) ) * cos( radians( lat ) ) * cos( radians( lng ) - radians(-122) ) + sin( radians(37) ) * sin( radians( lat ) ) ) ) AS distance FROM markers HAVING distance < 25 ORDER BY distance LIMIT 0 , 20;
```


5. ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Προτάσεις διαχείριση της σελίδας

Ως χρήστης χρειάζεται να συντάξετε και να καταγράψετε επιμελώς το βιογραφικό σας - προτείνεται στην αγγλική για να είναι κατανοητό και σε χρήστες του εξωτερικού -, αναφέροντας την εκπαίδευση, προϋπηρεσία, πιστοποιήσεις ή ικανότητες που κατέχετε, γνώσεις ξένων γλωσσών, εθελοντικές ενέργειες στις οποίες έχετε λάβει μέρος, δημοσιεύσεις που τυχόν έχετε και αν το επιλέξετε προσωπικά σας στοιχεία επικοινωνίας. Επίσης, ενδείκνυται να αναρτήσετε μια φωτογραφία σας σε μέγεθος ταυτότητας, κατά προτίμηση επαγγελματικού ύφους. Τέλος, μπορείτε να συμπεριλάβετε δείγμα της δουλειάς σας.

Η συγκεκριμένη εφαρμογή στοχεύει στη διαφάνεια. Για προφανείς λόγους αξιοποίησης κάθε ευκαιρίας που ενδέχεται να προκύψει, είναι απαραίτητο να διατηρείτε το προφίλ σας διαρκώς ενημερωμένο, αναφέροντας επιγραμματικά ρόλους - επιτεύγματα της προϋπηρεσίας σας και λέξεις κλειδιά που αφορούν στις ικανότητες ή αρμοδιότητές σας. Έτσι, μελλοντικοί εργοδότες ή σύμβουλοι στελέχωσης θα έχουν άμεση και εύκολη πρόσβαση σε πρόσφατα στοιχεία σας, άσχετα αν βρίσκεστε σε αναζήτηση εργασίας ή όχι.

5.2 Συμπεράσματα τεχνολογιών

Το βασικότερο συμπέρασμα που μπορεί να εξαχθεί είναι πως η PHP, η MySQL και ο Apache Server συνεργάζονται άψογα μεταξύ τους. Μάλιστα στις μέρες μας είναι από τα πλέον χρησιμοποιούμενα εργαλεία για την ανάπτυξη διαδικτυακών εφαρμογών με υψηλά κριτήρια αξιοπιστίας και απόδοσης. Η ανάπτυξη μιας πτυχιακής βοηθά τον σπουδαστή να εμπεδώσει θεωρητικά και τεχνικά θέματα τα οποία διδάχθηκε κατά την διάρκεια των σπουδών του, δημιουργώντας τις προϋποθέσεις να τα χρησιμοποιήσει στην μετέπειτα σταδιοδρομία του.

5.3 Επεκτάσεις

Η συγκεκριμένη εφαρμογή έχει υλοποιηθεί με τέτοιο τρόπο, ώστε με μικρές τροποποιήσεις να μπορεί να μετατραπεί σε μηχανισμό προσωποποίησης οποιουδήποτε άλλου αντικειμένου. Δηλαδή πολύ εύκολα μπορεί να γίνει επέκταση και να χρησιμοποιηθεί σαν σελίδα εύρεσης κατοικίας, υπηρεσιών, συντρόφου και οτιδήποτε άλλο μπορεί να σκεφτεί κανείς.

5.4 Σύνοψη

Συνοψίζοντας μπορούμε με βεβαιότητα να πούμε ότι καλύφθηκε σε ένα κάλο επίπεδο την λειτουργία της ιστοσελίδας, την ευχρηστία της. Ακόμα ότι η βάση δεδομένων που κατασκευάστηκε, κανονικοποιήθηκε με τρόπο κατά τον οποίο ακολουθηθήκαν οι κοινοί, παγκόσμιοι κανόνες. Τέλος, μελετήθηκαν ως ένα βάθος οι προτάσεις κατασκευής του προφίλ και της σελίδας του χρήστη για καλύτερη λειτουργία της ιστοσελίδας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Ramakrishnan R. και Gehrke J. (μετάφραση Δ. Δέρβος, Α. Ευαγγελίδης). *Συστήματα Διαχείρισης Βάσεων Δεδομένων*. 2η Έκδοση. Τόμοι Α' και Β', Εκδ. Τζιόλας, 2002.
2. Gilmore W.J. *Beginning PHP and MySQL 5: From Novice to Professional*. 2nd Edition. Apress, 2006.
3. Welling L., Thomson L. *PHP and MySQL Web Development*. Sams, 2001. (Ανάπτυξη Web Εφαρμογών με PHP και MySQL).
4. Codd, E.F. *The Relational Model for Database Management (Version 2 ed.)*. Addison Wesley Publishing Company, 1990.
5. Converse T. Park J. *PHP Bible*, 2nd Edition, Wiley, 2002.

Πηγές

6. <http://www.w3schools.com/>
7. <http://www.neural.uom.gr/Documents/DataBases/chapter3.pdf>
8. http://www.netrino.gr/reloaded/blog-post.php?bp_id=798
9. <http://en.wikipedia.org>
10. <http://www.w3c.org>
11. <http://www.tizag.com/javascriptT/javascriptform.php>
12. <http://www.phpmyadmin.net/>
13. http://help.adobe.com/el_GR/Dreamweaver/10.0_Using/dreamweaver_cs4_help.pdf
14. <http://www.php.net/>
15. <http://myphp.gr/manual/el>
16. <http://www.techteam.gr/wiki/PHP>
17. <http://www.planetphp.net>
18. <http://www.phpfreaks.com>
19. <http://dev.mysql.com/doc/refman/5.1/en/history.html>
20. <http://www.clearlearning.com/v4/install/mysql.html>
21. <http://www.mysql.com/whymysql/>
22. <http://www.1keydata.com/sql/sql.html>
23. <http://www.sql.org>
24. www.wdf.gr/articles/html-jquery-css/jquery.html