



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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Σχεδιασμός και υλοποίηση βάσης δεδομένων
οδηγών επαγγελματιών / ειδικοτήτων ΤΕΙ**

Της φοιτήτριας

Τσαντοπούλου Βαΐα- Αικατερίνη

Αρ. Μητρώου: 04/2557

Επιβλέπων καθηγητής

Βασίλης Κώστογλου

Θεσσαλονίκη 2011

ΠΡΟΛΟΓΟΣ

Η εφαρμογή αυτή δημιουργήθηκε από την Βαΐα-Αικατερίνη Τσαντοπούλου, φοιτήτρια του τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, υπό την εποπτεία του Αναπληρωτή καθηγητή κ. Βασίλη Κώστογλου, στα πλαίσια εκπόνησης πτυχιακής εργασίας με τίτλο: Σχεδιασμός και υλοποίηση βάσης δεδομένων οδηγών επαγγελματιών / ειδικοτήτων ΤΕΙ.

Οι οδηγοί επαγγελματιών / ειδικοτήτων είναι αποτέλεσμα της μελέτης που εκπονήθηκε στα πλαίσια της υλοποίησης του έργου της οριζόντιας δράσης των Γραφείων Διασύνδεσης με τίτλο «Οριζόντια δικτύωση των Γραφείων Διασύνδεσης των Ελληνικών ΤΕΙ». Η «Οριζόντια δικτύωση των Γραφείων Διασύνδεσης των Ελληνικών ΤΕΙ» αποτελεί έργο του ΕΠΕΑΕΚ ΙΙ (Επιχειρησιακό Πρόγραμμα Εκπαίδευσης και Αρχικής Επαγγελματικής Κατάρτισης), το οποίο υλοποιήθηκε στα πλαίσια του Γ΄ Κοινοτικού Πλαισίου Στήριξης και συγχρηματοδοτήθηκε από το Ευρωπαϊκό Κοινωνικό Ταμείο (75%) και Εθνικούς Πόρους (25%).

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της πτυχιακής αυτής, σχεδιάζεται και αναπτύσσεται ένας “ψηφιακός χάρτης” ειδικοτήτων της ανώτατης τεχνολογικής εκπαίδευσης με χρήση των τεχνολογιών HTML, CSS, PHP, MySQL και JavaScript.

Οι πληροφορίες που παρέχονται – και ειδικότερα οι περιληπτικές παρουσιάσεις της μεγάλης πλειοψηφίας των επαγγελματών ΤΕΙ – είναι χρήσιμες τόσο για τους αντίστοιχους αποφοίτους, δηλαδή τα σημερινά και προσεχή στελέχη της αγοράς εργασίας, όσο και για όσους (τελειόφοιτους λυκείων) ενδιαφέρονται να επιλέξουν ένα επάγγελμα με ευοίωνες προοπτικές. Πέρα από τους παραπάνω οι πληροφορίες της εφαρμογής είναι χρήσιμες για τους φορείς που σχετίζονται με την ανώτατη τεχνολογική εκπαίδευση είτε σε επίπεδο διοίκησης (ΥΠΕΠΘ, Διοικήσεις ΤΕΙ), είτε λειτουργίας (σχολές και τμήματα).

Έτσι χρήστες της εφαρμογής είναι οι υποψήφιοι της Ανωτάτης Εκπαίδευσης αλλά και κάθε ενδιαφερόμενος που θέλει να εστιάσει στο αντικείμενο κάθε Τμήματος αλλά και να γνωρίσει τις δυνατότητες επαγγελματικής αποκατάστασης και Μεταπτυχιακών Σπουδών.

Η εφαρμογή θα παρέχει σε κάθε χρήστη τη δυνατότητα αναζήτησης οδηγού επαγγέλματος / ειδικότητας βάση σύνθετων φίλτρων στην βάση δεδομένων. Τα φίλτρα αυτά εκτός από βασικά χαρακτηριστικά ενός οδηγού επαγγέλματος / ειδικότητας όπως τίτλο ή ΤΕΙ περιλαμβάνουν επιμέρους χαρακτηριστικά, έτσι ώστε το αποτέλεσμα της αναζήτησης να καλύπτει όσο το δυνατό περισσότερο τις απαιτήσεις και τις ανάγκες του χρηστή. Όταν εκτελεστεί μια αναζήτηση τα αποτελέσματα που θα επιστρέψει θα τοποθετούνται με μια συνοπτική περιγραφή στη σελίδα. Τέλος, η εφαρμογή θα αναρτηθεί σε μία πραγματική ιστοσελίδα για να είναι διαθέσιμη για όλους.

Λέξεις Κλειδιά: τεχνολογίες πληροφορικής, οδηγός επαγγέλματος, ανώτατη τεχνολογική εκπαίδευση

ABSTRACT

In this project, a “digital map” of specialties of higher technological education is being designed and developed, using HTML, PHP, CSS, MySQL and JavaScript.

The information - and especially the summary of a great majority of T.E.I. occupations- are useful for graduates, the current and coming labor market strains, as those (school graduates) who are being interested to choose a profession with good prospects.

Apart from that, application’s information is useful for associated entities with higher technological education, either administration’s level (Y.P.E.P.TH., administrations of T.E.I.), or operation’s level (faculties and departments).

So application’s users are the Higher Education’s candidates and every person who wants to focus on the subject of each section, and to discover the opportunities to learn vocational rehabilitation and Graduate Studies The application will provide to each user the ability to search profession’s guide/ specialty, based on composite filters in the database. These filters apart from the basic features of a professional’s guide/ specialty, such a title or college (T.E.I.), are including individual characteristics, so that the search result to cover as much as possible user’s demands and needs. When a search will be run, returned results will be placed with a short description on the page. Finally, the application will be published on a real website to be available for all.

Keywords: technologies, professional guide, higher technical education

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω όλους όσους συμμετείχαν και συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή μου, κ. Βασίλη Κώστογλου για την βοήθεια και την καθοδήγησή του. Τα μέλη της εξεταστικής επιτροπής που μου έκαναν την τιμή να αξιολογήσουν την προσπάθειά μου και την οικογένεια μου, που όλα αυτά τα χρόνια με στήριξε από όλες τις πλευρές.

Σας ευχαριστώ,

Βαία

ΕΥΡΕΤΗΡΙΟ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ	II
ΠΕΡΙΛΗΨΗ	III
ABSTRACT	IV
ΕΥΧΑΡΙΣΤΙΕΣ.....	V
ΕΥΡΕΤΗΡΙΟ ΠΕΡΙΕΧΟΜΕΝΩΝ	VI
ΕΙΣΑΓΩΓΗ	1
ΚΕΦΑΛΑΙΟ 1 - ΟΔΗΓΟΙ ΕΠΑΓΓΕΛΜΑΤΩΝ / ΕΙΔΙΚΟΤΗΤΩΝ	3
1.1. ΕΙΣΑΓΩΓΗ.....	3
1.2. ΠΕΔΙΑ ΟΔΗΓΩΝ ΕΠΑΓΓΕΛΜΑΤΩΝ / ΕΙΔΙΚΟΤΗΤΩΝ.....	6
1.3. ΕΠΙΛΟΓΟΣ.....	6
ΚΕΦΑΛΑΙΟ 2 - HTML	7
2.1. ΕΙΣΑΓΩΓΗ.....	7
2.2. WORLD WIDE WEB CONSORTIUM (W3C).	7
2.3. ΕΞΕΛΙΞΗ ΤΗΣ HTML ΜΕΧΡΙ ΣΗΜΕΡΑ	7
2.4. ΣΤΟΙΧΕΙΑ ΤΗΣ HTML.....	8
2.5. ΧΡΗΣΗ ΤΗΣ HTML ΣΤΗΝ ΕΦΑΡΜΟΓΗ	9
2.6. ΕΠΙΛΟΓΟΣ.....	17
ΚΕΦΑΛΑΙΟ 3 - CSS.....	18
3.1. ΕΙΣΑΓΩΓΗ.....	18
3.2. ΠΛΕΟΝΕΚΤΗΜΑΤΑ CSS	18
3.3. ΣΥΝΤΑΞΗ ΤΩΝ CSS	19
3.4. ΙΔΙΟΤΗΤΕΣ ΦΟΝΤΟΥ ΤΩΝ CSS	19
3.5. ΙΔΙΟΤΗΤΕΣ ΚΕΙΜΕΝΟΥ ΤΩΝ CSS.....	20
3.6. ΙΔΙΟΤΗΤΕΣ ΓΡΑΜΜΑΤΟΣΕΙΡΑΣ ΤΩΝ CSS.....	22
3.7. ΙΔΙΟΤΗΤΕΣ ΠΕΡΙΓΡΑΜΜΑΤΩΝ ΤΩΝ CSS.....	23
3.8. ΙΔΙΟΤΗΤΕΣ ΠΕΡΙΘΩΡΙΟΥ ΤΩΝ CSS	25
3.9. ΕΦΑΡΜΟΓΗ CSS.....	26
3.10. ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΤΥΛ.....	26
3.11. ΕΞΩΤΕΡΙΚΑ ΣΤΥΛ	27
3.12. ΈΝΘΕΤΑ ΣΤΥΛ.....	28
3.13. ΤΑ ΠΟΛΛΑΠΛΑ ΦΥΛΛΑ ΣΤΥΛ.....	29
3.14. ΥΠΟΣΤΗΡΙΞΗ ΤΩΝ CSS ΑΠΟ ΤΟΥΣ BROWSERS	29
3.15. ΕΠΙΛΟΓΟΣ	31
ΚΕΦΑΛΑΙΟ 4 - JAVASCRIPT.....	32
4.1. ΕΙΣΑΓΩΓΗ.....	32

4.2.	JAVA Vs JAVASCRIPT.....	32
4.3.	ΠΩΣ ΕΙΣΑΓΕΤΑΙ Η JAVASCRIPT	33
4.4.	ΠΟΥ ΕΙΣΑΓΕΤΑΙ Η JAVASCRIPT.....	34
4.5.	ΧΕΙΡΙΣΤΗΡΙΑ ΣΥΜΒΑΝΤΟΣ ΤΗΣ JAVASCRIPT.....	35
4.6.	ΣΥΝΑΡΤΗΣΕΙΣ ΤΗΣ JAVASCRIPT.....	37
4.7.	ΕΝΤΟΛΕΣ ΤΗΣ JAVASCRIPT	38
4.8.	DOCUMENT OBJECT MODEL (DOM).....	43
4.9	ΣΥΝΔΥΑΖΟΝΤΑΣ HTML, JAVASCRIPT ΚΑΙ CSS	48
4.10.	ΕΠΙΛΟΓΟΣ	58
ΚΕΦΑΛΑΙΟ 5 - APACHE, MYSQL & PHP		59
5.1.	ΕΙΣΑΓΩΓΗ.....	59
5.2.	APACHE HTTP WEB SERVER.....	59
5.3.	MYSQL	61
5.4.	PHP.....	65
5.5.	ΕΓΚΑΤΑΣΤΑΣΗ APACHE, MYSQL, PHP & PHPMYADMIN	68
5.6.	ΕΠΙΛΟΓΟΣ.....	73
ΚΕΦΑΛΑΙΟ 6 - ΥΛΟΠΟΙΗΣΗ		74
6.1.	ΕΙΣΑΓΩΓΗ.....	74
6.2.	ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ	74
6.3.	ΟΝΤΟΤΗΤΕΣ ΚΑΙ ΣΥΣΧΕΤΙΣΕΙΣ	76
6.5.	ΣΥΝΔΕΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	85
6.6.	ΕΠΙΛΟΓΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	85
6.7.	ΔΙΑΧΕΙΡΙΣΗ ΛΑΘΩΝ	86
6.8.	ΕΚΤΕΛΕΣΗ SQL ΕΡΩΤΗΜΑΤΩΝ.....	86
6.9.	ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ.....	87
ΚΕΦΑΛΑΙΟ 7 - ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ		91
7.1.	ΕΙΣΑΓΩΓΗ.....	91
7.2.	ΑΡΧΙΚΗ ΣΕΛΙΔΑ.....	91
7.2.	ΜΕΝΟΥ ΠΕΡΙΗΓΗΣΗΣ	91
7.3.	ΓΕΝΙΚΗ ΑΝΑΖΗΤΗΣΗ	92
7.4.	ΕΙΔΙΚΗ ΑΝΑΖΗΤΗΣΗ	94
7.5.	ΣΤΑΤΙΣΤΙΚΑ	94
7.6.	ΔΙΑΧΕΙΡΙΣΗ	96
7.7.	ΒΟΗΘΕΙΑ	96
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		97
ΠΑΡΑΡΤΗΜΑ Α.....		98

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Εικόνα 2-1 : Τίτλος σελίδας	10
Εικόνα 2-2 : Φόρμα υποβολής στοιχείων	15
Εικόνα 2-3 : Ένα drop-down list box	16
Εικόνα 4-1 : Εμφάνιση μηνύματος με JavaScript	38
Εικόνα 4-2 : Τα αντικείμενα του DOM	43
Εικόνα 4-3 : Η αρχική σελίδα	54
Εικόνα 4-4 : Κάνοντας κλικ στην εικόνα	55
Εικόνα 4-5 : Αλλαγή φόντου	55
Εικόνα 4-6 : Αλλαγή φόντου2.....	56
Εικόνα 4-7 : Αλλαγή φόντου3.....	56
Εικόνα 4-8 : Μετακίνηση κειμένου.....	57
Εικόνα 4-9 : Drop down list	57
Εικόνα 4-10 : Εμφάνιση μηνύματος.....	58
Εικόνα 5-1 : Η λειτουργία ενός web server	60
Εικόνα 5-2 : Το μοντέλο Πελάτη/Εξυπηρετητή	62
Εικόνα 5-3 : Αποθήκευση αρχείου	68
Εικόνα 5-4 : Λήψη αρχείου.....	69
Εικόνα 5-5 : Εικονίδιο εγκατάστασης.....	69
Εικόνα 5-6 : Εγκατάσταση	69
Εικόνα 5-7 : Εγκατάσταση 2.....	70
Εικόνα 5-8 : Εγκατάσταση 3.....	70
Εικόνα 5-9 : Εγκατάσταση 4.....	71
Εικόνα 5-10 : Εγκατάσταση 5.....	71
Εικόνα 5-11 : Εγκατάσταση 6.....	72
Εικόνα 5-12 : Τέλος εγκατάστασης.....	72
Εικόνα 5-13 : Λειτουργίες WampServer	73
Εικόνα 6-2 : Οι πίνακες της εφαρμογής.....	78
Εικόνα 6-3 : Ψηφιακός χάρτης	88
Εικόνα 6-4 : Μενού πλοήγησης της εφαρμογής.....	89
Εικόνα 7-1 : Η πρώτη σελίδα της εφαρμογής	91
Εικόνα 7-2 : Μενού της εφαρμογής.....	92
Εικόνα 7-3 : Φόρμα Γενικής αναζήτησης.....	93
Εικόνα 7-4 : Αποτελέσματα της αναζήτησης	93
Εικόνα 7-5 : Φόρμα Ειδικής Αναζήτησης.....	94
Εικόνα 7-6 : Στατιστικά στοιχεία τμημάτων.....	95
Εικόνα 7-7 : Στατιστικά στοιχεία τμημάτων 2.....	95
Εικόνα 7-8 : Είσοδος διαχειριστή	96
Εικόνα 7-9 : Σελίδα Βοήθειας.....	96

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 1-1 : Οδηγοί επαγγελμάτων / ειδικοτήτων.....	4
Πίνακας 2-1 : Χρήση της σήμανσης <table>2.....	11
Πίνακας 3-1 : Ιδιότητες φόντου των CSS.....	20
Πίνακας 3-3 : Ιδιότητες κειμένου των CSS	21
Πίνακας 3-3 : Ιδιότητες γραμματοσειράς των CSS	22
Πίνακας 4-4 : Ιδιότητες περιγραμμάτων των CSS	23
Πίνακας 3-5 : Ιδιότητες περιθωρίου των CSS.....	25
Πίνακας 5-1 : Διαφορές Java - JavaScript.....	32
Πίνακας 6-1 : Μορφή πίνακα βάσης δεδομένων.....	63
Πίνακας 7-1 : Τα πεδία του πίνακα cities	78
Πίνακας 6-2 : Τα πεδία του πίνακα regions	79
Πίνακας 6-3 : Τα πεδία του πίνακα tei.....	79
Πίνακας 6-4 : Τα πεδία του πίνακα subjects	79
Πίνακας 6-5 : Τα πεδία του πίνακα tei_depts.....	81
Πίνακας 6-6 : Τα πεδία του πίνακα related_subjects_schools_depts.....	82

ΕΙΣΑΓΩΓΗ

Στόχος της εργασίας αυτής είναι η εξ αρχής κατασκευή ενός εύχρηστου, φιλικού προς το χρήστη “ψηφιακού χάρτη” ειδικοτήτων της ανώτατης τεχνολογικής εκπαίδευσης. Επιμέρους στόχος καθορίστηκε να είναι η εύκολη διαχείρισή του από την πλευρά του διαχειριστή (χωρίς να γνωρίζει προγραμματισμό ή άλλου είδους τεχνική λεπτομέρεια) καθώς και η εύκολη παραμετροποίηση του.

Μετά την θεωρητική μελέτη του ζητήματος, αποφασίστηκε ότι η εφαρμογή θα παρέχει στον χρήστη την δυνατότητα δύο ειδών αναζητήσεων: γενική και εξειδικευμένη. Επιπλέον, στην εφαρμογή παρουσιάζονται στατιστικά στοιχεία που αφορούν το ποσοστό των Ιδρυμάτων που πληρούν - ή όχι - το κάθε πεδίο που περιέχεται στον οδηγό επαγγέλματος / ειδικότητας.

Η εφαρμογή στηρίζεται στη δυναμική γλώσσα προγραμματισμού PHP. Απαραίτητη προϋπόθεση για την λειτουργία ενός τέτοιου web-site είναι η ύπαρξη βάσης δεδομένων (ΒΔ) για να γίνει δυνατή η αποθήκευση και η διαχείριση των δεδομένων και πολλών παραμέτρων καθώς και ένα σύστημα διαχείρισης βάσεων δεδομένων. Για την τελευταία ανάγκη επιλέχτηκε η MySQL. Και τα δύο πακέτα λογισμικού διατίθενται δωρεάν και αναλύονται παρακάτω.

Το παρόν κείμενο της εργασίας περιλαμβάνει 7 κεφάλαια τα οποία παρουσιάζονται συνοπτικά παρακάτω.

Το πρώτο κεφάλαιο είναι εισαγωγικό. Παρουσιάζεται τι είναι ένας οδηγός επαγγέλματος / ειδικότητας, τα πεδία των οδηγιών επαγγελμάτων / ειδικοτήτων που χρησιμοποιήθηκαν στην εφαρμογή καθώς και οι ειδικότητες ΤΕΙ που καλύπτει η εφαρμογή.

Στο δεύτερο κεφάλαιο παρουσιάζεται η γλώσσα προγραμματισμού HTML και δίνονται παραδείγματα χρήσης της στην εφαρμογή.

Στο τρίτο κεφάλαιο γίνεται αναφορά στα Διαδοχικά Φύλλα Στυλ τα οποία χρησιμοποιήθηκαν για τη διαμόρφωση της παρουσίασης.

Στο τέταρτο κεφάλαιο βλέπουμε ένα βασικό κομμάτι μιας σύγχρονης ιστοσελίδας που είναι η αλληλεπίδραση του χρήστη με την σελίδα.

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

Το πέμπτο κεφάλαιο αναφέρεται στον τρόπο με τον οποίο μία ιστοσελίδα μπορεί να γίνει database-driven Web site.

Στο έκτο κεφάλαιο γίνεται ανάλυση της εφαρμογής και παρουσιάζονται κάποια χαρακτηριστικά κομμάτια κώδικα.

Τέλος, στο έβδομο παρουσιάζονται οι οδηγίες εγκατάστασης της εφαρμογής.

ΚΕΦΑΛΑΙΟ 1

Οδηγοί επαγγελματών / ειδικοτήτων

1.1. Εισαγωγή

Η επιλογή επαγγέλματος είναι μία από τις σημαντικότερες αποφάσεις που παίρνει ένας άνθρωπος στην ζωή του, ιδιαίτερα στις μέρες μας, που η ανεργία είναι υψηλή και υπάρχει ρευστότητα στην αγορά εργασίας και στις εργασιακές σχέσεις. Οι νέοι καλούνται επίμονα σε μια κρίσιμη περίοδο της ζωής τους, στα χρόνια της εφηβείας, έχοντας μικρή γνώση του εαυτού τους και του κόσμου, να κάνουν επιλογές σε σχέση με το μέλλον τους. Η οικονομική ρευστότητα και η τεχνολογική μεταβλητότητα της εποχής μας έχει σαν αποτέλεσμα πολλοί εργαζόμενοι να αναγκάζονται να αλλάξουν επάγγελμα δύο και τρεις φορές στη διάρκεια της σταδιοδρομίας τους. Παλιότερα, η επαγγελματική πορεία μετά την κατάρτιση σε ένα αντικείμενο ήταν κάτι στατικό, ενώ σήμερα ορίζεται ως μια σειρά από συνεχείς, προσωπικές επιλογές, οι οποίες παρακολουθούν ανάγκες που μεταβάλλονται συνεχώς.

Μέσα σε αυτό το δύσκολο τοπίο οι νέοι άνθρωποι ζητούν πληροφορίες για τις επαγγελματικές επιλογές που έχουν στη διάθεσή τους. Οι οδηγοί επαγγελματών / ειδικοτήτων προσπαθούν να καλύψουν αυτό το κενό. Δίνουν συνοπτικά και με απλό τρόπο τις βασικές επαγγελματικές επιλογές που ανοίγονται μπροστά σε κάθε νέο. Η αξιόπιστη ενημέρωση σχετικά με τις υπάρχουσες δυνατότητες της αγοράς εργασίας και τα χαρακτηριστικά των επαγγελμάτων μπορούν να βοηθήσουν τους υποψήφιους φοιτητές ή πτυχιούχους να κάνουν την σωστή επαγγελματική επιλογή.

Στην εφαρμογή καλύπτεται η μεγάλη πλειοψηφία των ειδικοτήτων που παρέχονται από τα ελληνικά ΤΕΙ. Πιο συγκεκριμένα αναλύονται 86 ειδικότητες αποφοίτων ΤΕΙ από τις συνολικά 96, σε ποσοστό ~ 90%. Η κάλυψη των αντίστοιχων τμημάτων ΤΕΙ είναι ακόμη μεγαλύτερη: οι οδηγοί που παρουσιάζονται αντιστοιχούν σε 185 τμήματα ΤΕΙ από τα συνολικά 195 υπάρχοντα, δηλαδή κάλυψη σε ποσοστό 95%. Απουσιάζουν μόνο 10 ειδικότητες (που αντιστοιχούν σε ισάριθμα «μοναδικά» τμήματα), για τις οποίες δεν εκπονήθηκαν πρωτογενείς οδηγοί.

Στον παρακάτω πίνακα παρουσιάζονται οι ειδικότητες που καλύπτονται από την εφαρμογή.

Πίνακας 1-1 : Οδηγοί επαγγελμάτων / ειδικοτήτων

ΑΙΣΘΗΤΙΚΗΣ ΚΑΙ ΚΟΣΜΗΤΟΛΟΓΙΑΣ
ΑΝΑΚΑΙΝΙΣΗΣ & ΑΠΟΚΑΤΑΣΤΑΣΗΣ ΚΤΙΡΙΩΝ
ΑΝΘΟΚΟΜΙΑΣ & ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ
ΑΥΤΟΜΑΤΙΣΜΟΥ
ΒΙΒΛΙΟΘΗΚΟΝΟΜΙΑΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΒΙΟΜΗΧΑΝΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ
ΒΡΕΦΟΝΗΠΙΟΚΟΜΙΑΣ
ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΟΠΟΓΡΑΦΙΑΣ
ΓΕΩΠΟΝΟΥ ΖΩΪΚΗΣ ΠΑΡΑΓΩΓΗΣ
ΓΕΩΠΟΝΟΥ ΦΥΤΙΚΗΣ ΠΑΡΑΓΩΓΗΣ
ΓΕΩΡΓΙΚΗΣ ΜΗΧΑΝΟΛΟΓΙΑΣ ΚΑΙ ΥΔΑΤΙΝΩΝ ΠΟΡΩΝ
ΓΕΩΤΕΧΝΟΛΟΓΙΑΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ
ΓΡΑΦΙΣΤΙΚΗΣ
ΔΑΣΠΟΝΙΑΣ & ΔΙΑΧΕΙΡΙΣΗΣ ΦΥΣΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ
ΔΗΜΟΣΙΑΣ ΥΓΙΕΙΝΗΣ
ΔΗΜΟΣΙΩΝ ΣΧΕΣΕΩΝ & ΕΠΙΚΟΙΝΩΝΙΑΣ
ΔΙΑΚΟΣΜΗΤΙΚΗΣ
ΔΙΑΤΡΟΦΗΣ & ΔΙΑΙΤΟΛΟΓΙΑΣ
ΔΙΑΙΤΟΛΟΓΟΣ - ΔΙΑΤΡΟΦΟΛΟΓΟΣ
ΔΙΑΧΕΙΡΙΣΤΗ ΕΡΓΩΝ
ΔΙΕΘΝΟΥΣ ΕΜΠΟΡΙΟΥ
ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ
ΔΙΟΙΚΗΣΗΣ ΓΕΩΡΓΙΚΩΝ ΕΚΜΕΤΑΛΛΕΥΣΕΩΝ
ΔΙΟΙΚΗΣΗΣ ΜΟΝΑΔΩΝ ΥΓΕΙΑΣ & ΠΡΟΝΟΙΑΣ
ΔΟΜΙΚΩΝ ΕΡΓΩΝ ΑΣΠΑΙΤΕ
ΕΜΠΟΡΙΑΣ & ΔΙΑΦΗΜΙΣΗΣ
ΕΜΠΟΡΙΑΣ & ΠΟΙΟΤΙΚΟΥ ΕΛΕΓΧΟΥ ΑΓΡΟΤΙΚΩΝ ΠΡΟΪΟΝΤΩΝ
ΕΝΕΡΓΕΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΕΠΙΣΚΕΠΤΩΝ-ΤΡΙΩΝ ΥΓΕΙΑΣ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ & ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΕΡΓΟΘΕΡΑΠΕΙΑΣ
ΕΡΓΩΝ ΥΠΟΔΟΜΗΣ ΑΣΠΑΙΤΕ
ΕΦΑΡΜΟΓΩΝ ΞΕΝΩΝ ΓΛΩΣΣΩΝ ΣΤΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΣΤΟ ΕΜΠΟΡΙΟ
ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΔΙΟΙΚΗΣΗ ΚΑΙ ΣΤΗΝ ΟΙΚΟΝΟΜΙΑ
ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ
ΗΛΕΚΤΡΟΛΟΓΙΑΣ ΑΣΠΑΙΤΕ
ΗΛΕΚΤΡΟΛΟΓΟΥ ΜΗΧΑΝΙΚΟΥ
ΗΛΕΚΤΡΟΝΙΚΗΣ

ΗΛΕΚΤΡΟΝΙΚΗΣ ΑΣΠΑΙΤΕ

ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΘΕΡΜΟΚΗΠΙΑΚΩΝ ΚΑΛΛΙΕΡΓΕΙΩΝ & ΑΝΘΟΚΟΜΙΑΣ

ΙΑΤΡΙΚΩΝ ΕΡΓΑΣΤΗΡΙΩΝ

ΙΧΘΥΟΚΟΜΙΑΣ & ΑΛΙΕΙΑΣ

ΚΛΩΣΤΟΨΦΑΝΤΟΥΡΓΙΑΣ

ΚΟΙΝΩΝΙΚΗΣ ΕΡΓΑΣΙΑΣ

ΛΑΪΚΗΣ ΚΑΙ ΠΑΡΑΔΟΣΙΑΚΗΣ ΜΟΥΣΙΚΗΣ

ΛΟΓΙΣΤΙΚΗΣ

ΛΟΓΟΘΕΡΑΠΕΥΤΗ

ΜΑΙΕΥΤΙΚΗΣ

ΜΗΧΑΝΙΚΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΗΧΑΝΙΚΩΝ ΕΡΓΩΝ ΥΠΟΔΟΜΗΣ

ΜΗΧΑΝΟΛΟΓΙΑΣ

ΜΗΧΑΝΟΛΟΓΙΑΣ

ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ & ΑΚΟΥΣΤΙΚΗΣ

ΝΑΥΠΗΓΙΚΗΣ

ΝΟΣΗΛΕΥΤΙΚΗ

ΟΔΟΝΤΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΟΙΝΟΛΟΓΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΠΟΤΩΝ

ΟΠΤΙΚΗΣ & ΟΠΤΟΜΕΤΡΙΑΣ

ΟΧΗΜΑΤΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΟΛΙΤΙΚΩΝ ΔΟΜΙΚΩΝ ΕΡΓΩΝ

ΡΑΔΙΟΛΟΓΙΑΣ & ΑΚΤΙΝΟΛΟΓΙΑΣ

ΣΤΕΛΕΧΩΝ ΣΥΝΕΤΑΙΡΙΣΤΙΚΩΝ ΟΡΓΑΝΩΣΕΩΝ ΚΑΙ ΕΚΜΕΤΑΛΛΕΥΣΕΩΝ

ΣΥΝΤΗΡΗΤΕΣ ΑΡΧΑΙΟΤΗΤΩΝ & ΕΡΓΩΝ ΤΕΧΝΗΣ

ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΑΡΑΓΩΓΗΣ ΕΝΔΥΜΑΤΩΝ

ΣΧΕΔΙΑΣΤΗ & ΤΕΧΝΟΛΟΓΟΥ ΞΥΛΟΥ & ΕΠΙΠΛΟΥ

ΤΕΧΝΟΛΟΓΙΑΣ ΑΛΙΕΙΑΣ ΚΑΙ ΥΔΑΤΟΚΑΛΛΙΕΡΓΕΙΩΝ

ΤΕΧΝΟΛΟΓΙΑΣ ΓΕΩΡΓΙΚΩΝ ΠΡΟΪΟΝΤΩΝ

ΤΕΧΝΟΛΟΓΙΑΣ ΓΡΑΦΙΚΩΝ ΤΕΧΝΩΝ

ΤΕΧΝΟΛΟΓΙΑΣ ΙΑΤΡΙΚΩΝ ΟΡΓΑΝΩΝ

ΤΕΧΝΟΛΟΓΙΑΣ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΦΥΣΙΚΩΝ ΠΟΡΩΝ

ΤΕΧΝΟΛΟΓΙΑΣ ΤΡΟΦΙΜΩΝ

ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΟΙΚΗΣΗΣ

ΤΟΠΙΚΗ ΑΥΤΟΔΙΟΙΚΗΣΗ

ΤΟΠΟΓΡΑΦΙΑΣ

ΤΟΥΡΙΣΤΙΚΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΤΥΠΟΠΟΙΗΣΗΣ ΚΑΙ ΔΙΑΚΙΝΗΣΗΣ ΠΡΟΪΟΝΤΩΝ ΕΦΟΔΙΑΣΤΙΚΗΣ

ΦΥΣΙΚΟΘΕΡΑΠΕΙΑ

ΦΩΤΟΓΡΑΦΙΑΣ & ΟΠΤΙΚΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΩΝ

ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ & ΕΛΕΓΚΤΙΚΗ

ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ & ΑΣΦΑΛΙΣΤΙΚΗΣ

ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

1.2. Πεδία οδηγών επαγγελματών / ειδικοτήτων

Τα πεδία που περιέχουν οι οδηγοί επαγγελματών / ειδικοτήτων είναι τα ακόλουθα:

- Κωδικοποίηση επαγγέλματος
- Περιγραφή Επαγγέλματος
- Πρόγραμμα Σπουδών
- Επαγγελματικά Δικαιώματα
- Συναφείς Ειδικότητες και Τμήματα ΤΕΙ
- Μεταπτυχιακά στην Ελλάδα
- Συμπράξεις
- Μεταπτυχιακά στο Εξωτερικό
- Υποτροφίες
- Επαγγελματικοί και Επιστημονικοί Φορείς
- Εργασία στον Δημόσιο Τομέα
- Εργασία στον Ιδιωτικό Τομέα
- Αυτοαπασχόληση
- Ασφαλιστικά Θέματα

1.3. Επίλογος

Όπως αναφέρθηκε παραπάνω οι οδηγοί επαγγελματών / ειδικοτήτων περιέχουν σημαντικές πληροφορίες ώστε κάποιος να ενημερωθεί σωστά για κάποιο επάγγελμα. Αυτός είναι και ο στόχος της εφαρμογής να ενημερώσει σωστά όσους ενδιαφέρονται έτσι ώστε να κάνουν την καλύτερη επιλογή επαγγέλματος / ειδικότητας. Στο επόμενο κεφάλαιο ξεκινά η παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 2

HTML

2.1. Εισαγωγή

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα **HTML** που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα. Η HTML είναι ακρωνύμιο των λέξεων **HyperText Markup Language** (Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου) και ο ορισμός της βασίζεται στην SGML (Standard Generalized Markup Language) που αποτελεί ένα πρότυπο για ορισμό άλλων γλωσσών σήμανσης κειμένου. Η HTML δεν είναι γλώσσα προγραμματισμού αλλά γλώσσα μορφοποίησης (markup language) και έχει τυποποιηθεί από τον παγκόσμιο οργανισμό τυποποίησης τεχνολογιών του Web, τον World Wide Web Consortium (W3C).

2.2. World Wide Web Consortium (W3C).

Η Κοινοπραξία του Παγκόσμιου Ιστού, γνωστή ως World Wide Web Consortium (W3C) έχει ως στόχο να οδηγήσει τον Παγκόσμιο Ιστό στο μέγιστο των δυνατοτήτων του, αναπτύσσοντας πρότυπα τεχνολογιών. Το W3C συστάθηκε το 1994 από τον εφευρέτη του Παγκόσμιου Ιστού, Sir Tim Berners-Lee, στο Πανεπιστήμιο MIT της Μασαχουσέτης των Ηνωμένων Πολιτειών.

Το W3C διοικείται από μία κεντρική Ομάδα και έρχεται σε επαφή με τις κατά τόπους κοινότητες διεθνώς, μέσω των Γραφείων του. Στην ανάπτυξη των τεχνολογιών του συνεισφέρουν τα εγγεγραμμένα Μέλη του, καθώς και το ευρύ κοινό. Το όραμα του W3C για τον Ιστό περιλαμβάνει τη συμμετοχή, τη διανομή της γνώσης, και με αυτόν τον τρόπο τη δόμηση της εμπιστοσύνης σε μια παγκόσμια κλίμακα.

2.3. Εξέλιξη της HTML μέχρι σήμερα

Το 1990, ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδους αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό

ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσας HTML.

Μέχρι το 1993 η HTML (τότε στην αρχική της έκδοση) ήταν ένα περιβάλλον δημιουργίας υπερκειμένων (hypertext). Το 1994, αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό IETF (Internet Engineering Task Force) που δίνει νέες δυνατότητες στην HTML. Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996) που θεωρήθηκε σταθμός. Σήμερα η HTML βρίσκεται στην 4^η έκδοση της και το W3C δεν θα συνεχίσει να αναπτύσσει την HTML. Η μελλοντική δουλειά του W3C θα επικεντρωθεί στην XHTML.

2.4. Στοιχεία της HTML

Η HTML χρησιμοποιεί ειδικές **σημάνσεις** (tags) για να δώσει τις απαραίτητες οδηγίες στον περιηγητή Ιστού (Web browser). Οι σημάνσεις είναι εντολές προς τον browser για τον τρόπο εμφάνισης του κειμένου και για την εμφάνιση πολυμεσικού περιεχομένου (εικόνες, video, ήχος). Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Οι οδηγίες είναι case insensitive, δηλαδή, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία και ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

Η βασική δομή ενός HTML αρχείου είναι η παρακάτω:

```
<html> Όλος ο κώδικας περιέχεται στην βασική ετικέτα <HTML>
```

```
<head> Εντολές προς τον υπολογιστή. Δεν εμφανίζονται στο χρήστη.
```

```
</head>
```

```
<body>
```

```
    Περιεχόμενο που εμφανίζεται στο χρήστη.
```

```
</body>
```

```
</html>
```

Σε ένα έγγραφο HTML υπάρχουν αρκετές δεκάδες σημάνσεις οι οποίες είναι γραμμένες απευθείας σε μορφή απλού κειμένου στο HTML έγγραφο όπου μπορούν να διερμηνευτούν από το λογισμικό του υπολογιστή. Όταν ένας browser ανοίγει ένα αρχείο HTML ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή. Οι σημάνσεις αυτές καθ' εαυτές δεν παρουσιάζονται στον browser και είναι ξεχωριστές από το περιεχόμενο που περικλείουν. Επίσης, κάθε σήμανση (εκτός λίγων εξαιρέσεων π.χ.
) έχει ένα σημείο που ανοίγει (π.χ. <HTML>) και ένα σημείο που κλείνει (π.χ. </HTML>). Το κλείσιμο της σήμανσης περιέχει ένα / μετά το αρχικό <. Οι σημάνσεις μπορεί να περιέχουν ιδιότητες και κάθε ιδιότητα τιμές.

Η βασική δομή μιας σήμανσης είναι η παρακάτω:

```
< σήμανση [ιδιότητα ="τιμή"]>
```

περιεχόμενο

```
</ σήμανση >
```

Παράδειγμα:

```
<body bgcolor="green" background="picture.jpg">
```

Μια απλή σελίδα

```
</body>
```

2.5. Χρήση της HTML στην εφαρμογή

Παρακάτω παρουσιάζονται οι βασικότερες σημάνσεις που χρησιμοποιήθηκαν στην εφαρμογή καθώς και κάποια σχόλια για τον τρόπο εφαρμογής τους.

2.5.1. Σήμανση <html>

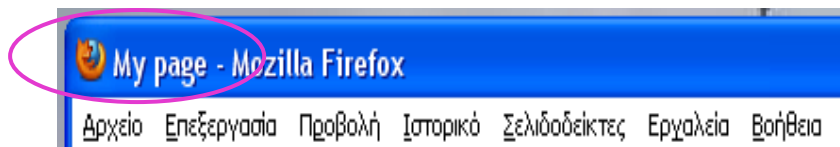
Το στοιχείο αυτό είναι το πρώτο το οποίο χρησιμοποιείται στο έγγραφο και περικλείει όλα τα υπόλοιπα μέσα σε αυτό. Με αυτό τον τρόπο πληροφορούμε το φυλλομετρητή ότι οι γραμμές που περικλείονται μέσα σε αυτές τις δύο σημάνσεις είναι κώδικας γραμμένος σε γλώσσα HTML. Επίσης, είναι γνωστό ως στοιχείο ρίζα (root element).

2.5.2. Σήμανση <head>

Είναι το κομμάτι κώδικα που παραμένει κρυφό στο χρήστη και περιέχει πληροφορίες σχετικά με το HTML έγγραφο. Περιλαμβάνει λέξεις κλειδιά που περιγράφουν την ιστοσελίδα καθώς άλλα και συνδέσμους με αρχεία τα οποία χρησιμοποιούνται από το έγγραφο όπως αρχεία CSS και JavaScript. Επίσης, μέσα στη σήμανση head περιέχεται και ο τίτλος του εγγράφου <title>.

2.5.3. Σήμανση <title>

Ο τίτλος του εγγράφου που εμφανίζεται στη γραμμή τίτλου του φυλλομετρητή. π.χ. <title> My page <title>



Εικόνα 2-1 : Τίτλος σελίδας

2.5.4. Σήμανση <body>

Μέσα στη σήμανση <body> περιλαμβάνεται όλο το κείμενο και οποιοδήποτε άλλο περιεχόμενο, όπως σύνδεσμοι, εικόνες κ.ά.

2.5.5. Σήμανση

Η σήμανση
 αναγκάζει τον browser να σταματήσει να τυπώνει στην τρέχουσα γραμμή και έτσι το αμέσως επόμενο κείμενο ξεκινά από το αριστερό περιθώριο της επόμενης γραμμής. Το
 δεν προσθέτει επιπλέον χώρο πάνω ή κάτω από τη νέα γραμμή και δεν αλλάζει τη γραμματοσειρά ή το στυλ του κειμένου.

2.5.6. Σήμανση <table>

Το στοιχείο ενός πίνακα ξεκινά με την σήμανση `<table>` και τελειώνει με τη σήμανση `</table>`. Χρησιμοποιείται για να οργανώσουμε και να παρουσιάσουμε πληροφορίες σε μορφή πίνακα. Παράλληλα χρησιμοποιείται πάρα πολύ για τη μορφοποίηση των ιστοσελίδων αλλά αυτό έχει περιοριστεί αρκετά καθώς οι περισσότεροι πια κάνουν χρήση των CSS. Μέσα σε ένα πίνακα με τις σημάνσεις `<tr>` `</tr>` και `<td>` `</td>` καθορίζουμε την αρχή και το τέλος μιας γραμμής και μιας στήλης του πίνακα αντίστοιχα.

```
<table border="1" style="border-color:fuchsia">
  <tr>
    <td> Επικεφαλίδα στήλης 1 </td>
    <td> Επικεφαλίδα στήλης 2</td>
  </tr>
  <tr>
    <td> δεδομένα</td>
    <td> δεδομένα</td>
  </tr>
</table>
```

Το παραπάνω κομμάτι κώδικα παράγει τον παρακάτω πίνακα:

Πίνακας 2-1 : Χρήση της σήμανσης `<table>`

Επικεφαλίδα στήλης 1	Επικεφαλίδα στήλης 2
δεδομένα	δεδομένα

2.5.7. Σήμανση `<input>`

Με τη σήμανση `<input>` ορίζουμε στοιχεία μιας φόρμας όπως text boxes, check boxes, buttons και radio buttons. Το είδος του στοιχείου που θα οριστεί στη φόρμα καθορίζεται από την ιδιότητα `type`, αν δεν υπάρχει η προκαθορισμένη τιμή είναι `text`. Μια άλλη σημαντική ιδιότητα της `<input>` είναι η `name` με την οποία μπορούμε να κάνουμε αναφορά στο πεδίο. Επίσης, σημαντικές ιδιότητες της `<input>` είναι οι `checked`, `disabled` και `maxlength`. Με την `checked`, μπορούμε να δώσουμε αρχική

τιμή για τα checkbox και τα radio buttons. Η disabled απενεργοποιεί το αντίστοιχο στοιχείο (η τιμή ενός απενεργοποιημένου στοιχείου δεν αποστέλλεται για επεξεργασία). Η ιδιότητα maxlength καθορίζει πόσους χαρακτήρες μπορεί να δεχτεί ένα πεδίο κειμένου. Για παράδειγμα ο παρακάτω κώδικας

```
<input name="text" type="text" size="10" maxlength="30">
```

ορίζει ένα πεδίο κειμένου μεγέθους 10 χαρακτήρων (το μέρος που φαίνεται) που δέχεται το πολύ 30 χαρακτήρες.

2.5.8. Σήμανση <script>

Με τη σήμανση <script> εισάγουμε κάποιο script στην σελίδα το οποίο είτε είναι γραμμένο με JavaScript είτε με κάποια άλλη γλώσσα.

2.5.9. Σήμανση <p>

Με την σήμανση <p> διαχωρίσουμε προτάσεις κειμένου κατά την εμφάνισή τους από τον browser.

2.5.10. Σήμανση <center>

Με τη σήμανση <center> μπορούμε να κεντράρουμε το περιεχόμενο της HTML που βρίσκεται μέσα στην σήμανση. Η σήμανση είναι συντόμευση της σήμανσης <div align=center> και λειτουργούν το ίδιο. Η σήμανση <center> τοποθετείται πριν από το κείμενο που θέλουμε να κεντράρουμε και η </center> μετά το τέλος του κειμένου.

```
<center>
```

```
<h1> Αυτή η επικεφαλίδα θα είναι στο κέντρο της web σελίδας</h1>
```

```
</center>
```

**2.5.11. Σήμανση **

Με την σήμανση μπορούμε να ρυθμίσουμε την γραμματοσειρά του περιεχομένου της HTML που βρίσκεται μέσα στην σήμανση. Με την ιδιότητα size ορίζουμε το μέγεθος των γραμμάτων. Η μπορεί να πάρει τιμές από 1 μέχρι 7 αλλά και σχετικές τιμές όπως (από -3 έως +4) που καθορίζουν το μέγεθος της σε

σχέση με την προκαθορισμένη γραμματοσειρά που έχει οριστεί στον browser. Αν θέλουμε μπορούμε να σώσουμε ξεχωριστά χρώματα σε μεμονωμένες λέξεις ή χαρακτήρες με την ιδιότητα color.

2.5.12. Σήμανση <a>

Για να δημιουργήσουμε ένα σύνδεσμο χρειαζόμαστε δύο πράγματα:

Το όνομα του αρχείου στον τοπικό δίσκο ή το URL του αρχείου, για το οποίο θέλουμε να δημιουργήσουμε το σύνδεσμο.

Το κείμενο που θα εμφανίζεται στη σελίδα και στο οποίο εάν κάνουμε κλικ θα ακολουθήσουμε τον σύνδεσμο.

Για να δημιουργηθεί ο σύνδεσμος χρησιμοποιείται η σήμανση <a> και έχει την εξής μορφή:

```
<a name="link_1" href="http://www.google.gr" title="link">
```

Οι **ιδιότητες** name, href και title περιγράφουν τον ίδιο σύνδεσμο. Η href (**H**ypertext **R**eference, αναφορά υπέρ-κειμένου) ορίζει το όνομα ή το URL του αρχείου στον οποίο δείχνει ο σύνδεσμος και είναι υποχρεωτική. Οι ιδιότητες name, title μπορούν να παραληφθούν. Το κείμενο ανάμεσα στη σήμανση αρχής <a> και τέλους είναι υπογραμμισμένο (εκτός και αν έχει καθορισθεί διαφορετικά) και πατώντας πάνω του ο τελικός χρήστης ενεργοποιεί το σύνδεσμο. Εκτός από κείμενο για την ενεργοποίηση του συνδέσμου μπορεί να υπάρχει και κάποιο γραφικό.

**2.5.13. Σήμανση **

Με τη σήμανση μπορούμε να προσθέσουμε γραφικά σε ένα HTML αρχείο. Η βασική ιδιότητα της σήμανσης είναι η src με την οποία καθορίζουμε τη διεύθυνση του αρχείου γραφικών.

2.5.14. Σήμανση <style>

Με την ετικέτα <style> μπορούμε να δημιουργήσουμε ένα ενσωματωμένο στυλ στο έγγραφο.

**2.5.15. Σήμανση <div> & **

Αυτές οι δύο σημάνσεις ομαδοποιούν το περιεχόμενο της HTML με σκοπό να δοθούν κοινά χαρακτηριστικά σε αυτό ή να αλλάξει αργότερα χαρακτηριστικά με την χρήση δυναμικής HTML και CSS.

Η διαφορά των δύο σημάνσεων είναι ότι η είναι στοιχείο το οποίο μπορεί να υπάρχει οπουδήποτε στη ροή του κειμένου χωρίς να προκαλεί αλλαγή γραμμής σε αντίθεση με την <div> που προκαλεί αλλαγή γραμμής.

2.5.16. Σήμανση <form>

Με την σήμανση <form> έχουμε τη δυνατότητα να στείλουμε δεδομένα στον εξυπηρετητή (web server). Οι φόρμες λειτουργούν ως εξής: ο χρήστης συμπληρώνει με δεδομένα τα πεδία μιας φόρμας και πατώντας το κουμπί υποβολής της φόρμας τα δεδομένα στέλλονται στον web server. Ο web server δέχεται τα δεδομένα και συνήθως τα επεξεργάζεται ή τα αποθηκεύει κάπου.

Μια φόρμα έχει δύο βασικές ιδιότητες, τις method και action. Με την action καθορίζεται η url διεύθυνση του αρχείου που θα κληθεί για να λάβει τα δεδομένα που θα αποσταλούν, ενώ με την method η οποία μπορεί να πάρει δύο τιμές (post και get) καθορίζεται ο τρόπος με τον οποίο τα δεδομένα στον web browser στέλνονται στον web server.

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

Προσωπικά Στοιχεία:

Παρακαλώ εισάγετε τα προσωπικά σας στοιχεία στα ακόλουθα πεδία.

ΟΝΟΜΑ:	<input type="text"/>	* ΕΠΩΝΥΜΟ:	<input type="text"/>	*
ΟΔΟΣ:	<input type="text"/>	* ΠΟΛΗ:	<input type="text"/>	*
ΤΗΛΕΦΩΝΟ:	<input type="text"/>	* FAX:	<input type="text"/>	
ΑΡ.ΔΕΛ.ΤΑΥΤΟΤΗΤΑΣ:	<input type="text"/>	* ΑΦΜ:	<input type="text"/>	*
Email:	<input type="text"/>	* ΔΟΥ:	<input type="text"/>	*

Στοιχεία Εισόδου:

Παρακαλώ εισάγετε τα επιθυμητά στοιχεία εισόδου στα ακόλουθα πεδία. Αν είστε πολίτης του δήμου επιλέξτε κατηγορία χρήστη 'Πολίτης Δήμου', αν είστε προμηθευτής επιλέξτε 'Προμηθευτής' σε διαφορετική περίπτωση επιλέξτε 'Απλός Χρήστης'.

Επιθυμητός Κωδικός Χρήστη (UserName)	<input type="text"/>	*
Κατηγορία Χρήστη	<input type="text" value="Επιλέξτε"/>	*

Όλα τα πεδία με * είναι υποχρεωτικά

Εικόνα 2-2 : Φόρμα υποβολής στοιχείων

2.5.17. Σημάνσεις <select> και <option>

Με τις σημάνσεις <select> και <option> μπορούμε να δημιουργήσουμε ένα μενού επιλογών από τις οποίες ο χρήστης μπορεί να επιλέξει μία που αντιστοιχεί σε μία τιμή. Μπορούμε να δημιουργήσουμε απλά drop-down list boxes όπου ο χρήστης επιλέγει μία από τις διαθέσιμες επιλογές, αλλά και multi-select list boxes στα οποία ο χρήστης έχει την δυνατότητα να επιλέξει πάνω από μία επιλογή. Οι επιλογές από τις οποίες διαλέγει ο χρήστης βρίσκονται μέσα στην σήμανση <option>.

Στο παρακάτω απόσπασμα κώδικα παρουσιάζεται ένα παράδειγμα χρήσεις των σημάνσεων:

Επέλεξε ένα στοιχείο

```
<select>
```

```
<option> Επιλογή 1
```

```
</option>
```



```
<option> Επιλογή 2      </option>

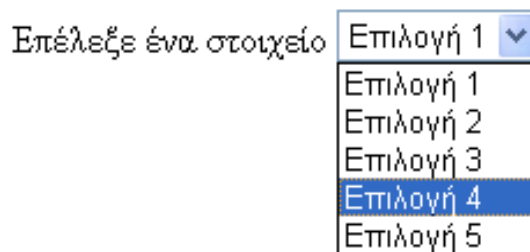
<option> Επιλογή 3      </option>

<option> Επιλογή 4      </option>

<option> Επιλογή 5      </option>

</select>
```

Το αποτέλεσμα του παραπάνω κώδικα είναι:



Εικόνα 2-3 : Ένα drop-down list box

2.5.18. Σήμανση `<link>`

Με τη σήμανση `link` μπορούμε να συνδέσουμε ένα έγγραφο HTML με εξωτερικά αρχεία. Η πιο κοινή του χρήση είναι να συνδέει εξωτερικά φύλλα στυλ (external style sheets) στο έγγραφο.

2.5.19. Σήμανση `<textarea>`

Με την σήμανση `textarea` δημιουργούμε ένα πεδίο πολλών γραμμών σε περίπτωση που το κείμενο προς εισαγωγή είναι πολύ μεγάλο για να χωρέσει σε ένα πεδίο μίας γραμμής κειμένου. Με τις ιδιότητες `cols` και `rows` μπορούμε να καθορίσουμε τις γραμμές και τις στήλες του `textarea`. Οι πιθανές τιμές που μπορεί να πάρει η `textarea` είναι:

Soft (Default). Το κείμενο εμφανίζεται με αναδίπλωση κειμένου και όταν υποβάλλεται δεν υπάρχουν χαρακτήρες αλλαγής γραμμής.

Hard. Το κείμενο εμφανίζεται με αναδίπλωση κειμένου και όταν υποβάλλεται υπάρχουν χαρακτήρες αλλαγής γραμμής.

Off. Το κείμενο δεν εμφανίζεται με αναδίπλωση κειμένου αλλά ακριβώς όπως υποβάλλεται από τον χρήστη. Αν ο χρήστης πατήσει το πλήκτρο Enter μόνο τότε θα γίνει αλλαγή γραμμής.

2.5.20. Σχόλια στην HTML

Τα σχόλια χρησιμοποιούνται για να γράφουμε σημειώσεις μέσα στον κώδικα. Δεν εμφανίζονται στην οθόνη του χρήστη. Ένα σχόλιο αρχίζει με το `<!--` και τελειώνει με το `-->`

π.χ. `<!-- Αυτό είναι ένα σχόλιο -->`

2.6. Επίλογος

Οι ετικέτες της HTML σχεδιάστηκαν αρχικά για να ορίσουν το περιεχόμενο ενός εγγράφου. Η δουλειά τους ήταν να λένε "Αυτή είναι μια επικεφαλίδα", "Αυτή είναι μια παράγραφος", "Αυτός είναι ένας πίνακας", χρησιμοποιώντας ετικέτες όπως `<h1>`, `<p>`, `<table>` κοκ. Η διάταξη (layout) του εγγράφου ήταν υπόθεση του φυλλομετρητή, χωρίς τη χρήση ετικετών μορφοποίησης (formatting tags).

Καθώς οι δύο κύριοι φυλλομετρητές, ο Netscape και ο Internet Explorer, συνέχισαν να προσθέτουν νέες HTML ετικέτες και χαρακτηριστικά (attributes), όπως η ετικέτα ``, στις αρχικές προδιαγραφές της HTML, γινόταν ολοένα και δυσκολότερη η δημιουργία Web γιατί οι σημάνσεις μορφοποίησης "μπλέκονταν" με τις σημάνσεις ορισμού κειμένου. Για να μπορέσει να λύσει αυτό το πρόβλημα, το W3C δημιούργησε τα ΣΤΥΛ (STYLES) σαν προσθήκη στην HTML 4.0.

ΚΕΦΑΛΑΙΟ 3

CSS

3.1. Εισαγωγή

Τα **CSS (Cascading Style Sheets)** δηλαδή Διαδοχικά Φύλλα Στυλ δημιουργήθηκαν με σκοπό τη διαμόρφωση της παρουσίασης - εμφάνισης μιας ιστοσελίδας έτσι ώστε αυτή να διαχωριστεί τελείως από την HTML. Χρησιμοποιούνται κυρίως σε εφαρμογές γραμμένες σε HTML και XHTML αλλά μπορεί να χρησιμοποιηθούν επίσης και σε οποιοδήποτε έγγραφο της XML για τη μορφοποίησή του. Ο διαχωρισμός αυτός επιτρέπει παράλληλα σε πολλά έγγραφα να μοιράζονται την ίδια μορφοποίηση με αποτέλεσμα τη μείωση της πολυπλοκότητας και της επανάληψης κώδικα. Για την δημιουργία και επεξεργασία εγγράφων CSS, μπορεί να χρησιμοποιηθεί ένας επεξεργαστής (editor) απλού κειμένου ή κάποιο πρόγραμμα ειδικό για CSS, που προσφέρει αρκετές ευκολίες στην διαμόρφωση του κώδικα CSS. Ένα πρόγραμμα (open source, δωρεάν διαθέσιμο για download και χρήση) για την επεξεργασία εγγράφων CSS, είναι το CSSED.

3.2. Πλεονεκτήματα CSS

Τα πλεονεκτήματα από τη χρήση style sheets είναι αρκετά κ σημαντικά. Κατ' αρχήν ο προγραμματιστής μπορεί εύκολα να τροποποιήσει παράλληλα την εμφάνιση πολλών διαφορετικών HTML εγγράφων τροποποιώντας μόνο ένα αρχείο στυλ. Έτσι, οι ενδεχόμενες αλλαγές, γίνονται μόνο σε ένα έγγραφο και αυτόματα εφαρμόζονται σε όλες τις σελίδες που το χρησιμοποιούν. Αν έχουμε ποτέ προσπαθήσει να αλλάξουμε τη γραμματοσειρά (font) ή το χρώμα (color) όλων των επικεφαλίδων (headings) σ' όλες τις ιστοσελίδες μας, θα καταλάβουμε γιατί τα CSS μπορούν να μας γλυτώσουν από πολλή δουλειά. Επίσης, ο συνολικός κώδικας είναι σημαντικά λιγότερος με τη χρήση CSS με αποτέλεσμα το αρχείο να είναι πιο μικρό και ελαφρύ "κατεβαίνοντας" έτσι γρηγορότερα.

3.3. Σύνταξη των CSS

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

επιλογέας {ιδιότητα: τιμή}

selector {property: value}

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

body {color: red}

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

p { font-family: "sans serif" }

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

p { text-align: center; color: green }

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

p { text-align: center;

color: black;

font-family: arial

}

3.4. Ιδιότητες φόντου των CSS

Οι ιδιότητες φόντου (background properties) μάς δίνουν τη δυνατότητα να ελέγξουμε το χρώμα φόντου ενός στοιχείου, να ορίσουμε μια εικόνα σαν φόντο, να

επαναλάβουμε μια εικόνα φόντου κατακόρυφα ή οριζόντια και να τοποθετήσουμε μια εικόνα σε μια σελίδα.

Πίνακας 3-1 : Ιδιότητες φόντου των CSS

Ιδιότητα	Περιγραφή	Τιμές
background	Ορίζει όλες τις ιδιότητες φόντου σε μια δήλωση	background-color background-image background-repeat background-attachment background-position
background-attachment	Ορίζει αν μια εικόνα φόντου είναι σταθερή (fixed) ή κυλάει μαζί με την υπόλοιπη σελίδα (scroll)	scroll, fixed
background-color	Ορίζει το χρώμα φόντου ενός στοιχείου	Color-rgb, color-hex, colorname, transparent
background-image	Ορίζει μια εικόνα σαν φόντο	url, none
background-position	Ορίζει τη θέση εκκίνησης μια εικόνας φόντου	top left, top center, top right, center left, center center, center right, bottom left bottom center, bottom right
background-repeat	Ορίζει αν και πώς θα επαναλαμβάνεται μια εικόνα φόντου	repeat, repeat-x, no-repeat

3.5. Ιδιότητες κειμένου των CSS

Οι ιδιότητες κειμένου (text properties) μάς δίνουν τη δυνατότητα να ελέγξουμε την εμφάνιση του κειμένου. Είναι δυνατό να αλλάξουμε το χρώμα ενός κειμένου, να αυξήσουμε ή να ελαττώσουμε το διάστημα ανάμεσα στους χαρακτήρες ενός

κειμένου, να ευθυγραμμίσουμε ένα κείμενο, να διακοσμήσουμε ένα κείμενο, να δημιουργήσουμε εσοχή στην πρώτη γραμμή ενός κειμένου κ.ά.

Πίνακας 3-3 : Ιδιότητες κειμένου των CSS

Ιδιότητα	Περιγραφή	Τιμές
color	Ορίζει το χρώμα του κειμένου	color
direction	Ορίζει την κατεύθυνση του κειμένου	ltr, rtl
letter-spacing	Αυξάνει ή ελαττώνει το διάστημα ανάμεσα στους χαρακτήρες	normal, length
text-align	Ευθυγραμμίζει το κείμενο σ' ένα στοιχείο	Left, right, center, justify
text-decoration	Προσθέτει διακόσμηση στο κείμενο	none, underline, overline line-through, blink
text-indent	Δημιουργεί εσοχή στην πρώτη γραμμή του κειμένου ενός στοιχείου	length
text-transform	Ελέγχει τα γράμματα σ' ένα στοιχείο	none, capitalize, uppercase, lowercase
white-space	Ορίζει το πώς θα αντιμετωπίζεται το λευκό κενό (white space) μέσα σ' ένα στοιχείο	normal, pre, nowrap
word-spacing	Αυξάνει ή ελαττώνει το διάστημα ανάμεσα στις λέξεις	normal, length

3.6. Ιδιότητες γραμματοσειράς των CSS

Οι ιδιότητες γραμματοσειρών μάς δίνουν τη δυνατότητα να αλλάξουμε την οικογένεια της γραμματοσειράς (font family), το έντονο (boldness), το μέγεθος (size) και το στυλ (style) ενός κειμένου.

Πίνακας 3-3 : Ιδιότητες γραμματοσειράς των CSS

Ιδιότητα	Περιγραφή	Τιμές
font	Ορίζει όλες τις ιδιότητες μιας γραμματοσειράς σε μια δήλωση	font-style, font-variant font-weight,font-size/line-height, fontfamily, caption, icon, menu, message-box, small-caption, status-bar
font-family	Μια λίστα προτεραιότητας οικογενειών γραμματοσειράς και/ή generic ονομάτων οικογένειας για ένα στοιχείο	family-name, generic-family
font-size	Ορίζει το μέγεθος μιας γραμματοσειράς	xx-small, x-small, small, medium, large,x-large xx-large, smaller, larger, length
font-size-adjust	Καθορίζει μια τιμή άποψης (aspect value) για ένα στοιχείο που θα διατηρήσει το x-height της πρώτης επιλεγμένης γραμματοσειράς	None, number

font-stretch	Συμπυκνώνει ή επεκτείνει την τρέχουσα οικογένεια γραμματοσειράς	Normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded
font-variant	Εμφανίζει το κείμενο με μικρά κεφαλαία (small-caps) ή κανονικά (normal)	normal, small-caps
font-weight	Ορίζει το βάρος (weight) μιας γραμματοσειράς	normal, bold, bolder, lighter 100, 200, ...,900

3.7. Ιδιότητες περιγραμμάτων των CSS

Οι ιδιότητες περιγραμμάτων μάς δίνουν τη δυνατότητα να καθορίσουμε το στυλ, το χρώμα και το πλάτος του περιγράμματος ενός στοιχείου. Στην HTML χρησιμοποιούμε πίνακες για να δημιουργούμε περιγράμματα γύρω από ένα κείμενο, αλλά με τις ιδιότητες περιγραμμάτων μπορούμε να δημιουργήσουμε περιγράμματα με ωραία εφέ και τα οποία μπορούν να εφαρμοσθούν σ' ένα οποιοδήποτε στοιχείο.

Πίνακας 4-4 : Ιδιότητες περιγραμμάτων των CSS

Ιδιότητα	Περιγραφή	Τιμές
border	Ορίζει όλες τις ιδιότητες για τα τέσσερα περιγράμματα σε μια δήλωση	border-width, border-style border-color
border-bottom	Ορίζει όλες τις ιδιότητες του κάτω περιγράμματος σε μια δήλωση	border-bottom-width, border-style, border-color
border-bottom-color	Ορίζει το χρώμα του κάτω περιγράμματος	border-color
border-bottom-style	Ορίζει το στυλ του κάτω	border-style

	περιγράμματος	
border-bottom-width	Ορίζει το πλάτος του κάτω περιγράμματος	thin, medium, thick, length
border-color	Ορίζει το χρώμα των τεσσάρων περιγραμμάτων και μπορεί να έχει τιμή από ένα έως τέσσερα χρώματα	color
border-left	Ορίζει όλες τις ιδιότητες του αριστερού περιγράμματος σε μια δήλωση	border-left-width, border-style, border-color
border-left-color	Ορίζει το χρώμα του αριστερού περιγράμματος	border-color
border-left-style	Ορίζει το στυλ του αριστερού περιγράμματος	border-style
border-left-width	Ορίζει το πλάτος του αριστερού περιγράμματος	thin, medium, thick, length
border-right	Ορίζει όλες τις ιδιότητες του δεξιού περιγράμματος σε μια δήλωση	border-right-width border-style, border-color
border-right-color	Ορίζει το χρώμα του δεξιού περιγράμματος	border-color
border-right-style	Ορίζει το στυλ του δεξιού περιγράμματος	border-style
border-right-width	Ορίζει το πλάτος του δεξιού περιγράμματος	thin, medium, thick, length
border-style	Ορίζει το στυλ των τεσσάρων περιγραμμάτων και μπορεί να έχει τιμή από ένα έως τέσσερα στυλ	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
border-top	Ορίζει όλες τις ιδιότητες του πάνω περιγράμματος σε μια δήλωση	border-top-width border-style, border-color
border-top-color	Ορίζει το χρώμα του πάνω	border-color

	περιγράμματος	
border-top-style	Ορίζει το στυλ του πάνω περιγράμματος	border-style
border-width	Ορίζει το πλάτος των τεσσάρων περιγραμμάτων σε μια δήλωση και μπορεί να έχει από μία έως τέσσερις τιμές	thin, medium, thick, length
border-top-width	Ορίζει το πλάτος του πάνω περιγράμματος	thin, medium, thick, length

3.8 Ιδιότητες περιθωρίου των CSS

Οι ιδιότητες περιθωρίου ορίζουν το διάστημα γύρω από τα στοιχεία. Μπορούμε να χρησιμοποιήσουμε αρνητικές τιμές για να επικαλύψουμε (overlap) το περιεχόμενο. Το πάνω, δεξιά, κάτω και αριστερό περιθώριο μπορούν να αλλάξουν ανεξάρτητα χρησιμοποιώντας ξεχωριστές ιδιότητες. Μπορούμε να χρησιμοποιήσουμε την ιδιότητα margin για να αλλάξουμε όλα τα περιθώρια μονομιάς.

Πίνακας 3-5 : Ιδιότητες περιθωρίου των CSS

Ιδιότητα	Περιγραφή	Τιμές
margin	Ορίζει τις ιδιότητες περιθωρίου σε μια δήλωση	margin-top, margin-right, margin-bottom, margin-left
margin-bottom	Ορίζει το κάτω περιθώριο ενός στοιχείου	auto, length, %
margin-left	Ορίζει το αριστερό περιθώριο ενός στοιχείου	auto, length, %
margin-right	Ορίζει το δεξιό περιθώριο ενός στοιχείου	auto, length, %

margin-top	Ορίζει το πάνω περιθώριο ενός στοιχείου	auto, length, %
------------	---	-----------------

3.9. Εφαρμογή CSS

Τα Φύλλα Στυλ επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο <head> μιας HTML σελίδας ή σ' ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά Φύλλα Στυλ μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο. Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν σ' ένα νέο εικονικό (virtual) Φύλλο Στυλ σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα :

1. Προεπιλογή του φυλλομετρητή.
2. Εξωτερικό Φύλλο Στυλ (External Style Sheet).
3. Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header του εγγράφου.
4. Inline Style, μέσα στο HTML στοιχείο.

3.10. Ενσωματωμένα Στυλ

Τα **ενσωματωμένα στυλ** (embedded styles) ορίζονται μέσα στη σήμανση <head> ανάμεσα στη σήμανση <style>. Από τη στιγμή που θα οριστεί ένα style για μια σήμανση κάθε εμφάνιση αυτής της σήμανσης θα μορφοποιείται σύμφωνα με το συγκεκριμένο style. Το μειονέκτημα των ενσωματωμένων στυλ είναι ότι εφαρμόζονται μόνο σε μία σελίδα, έτσι αν θέλουμε την ίδια μορφοποίηση σε μία άλλη σελίδα τότε πρέπει να ξαναγράψουμε τον ίδιο κώδικα.

Παράδειγμα ενός τέτοιου στυλ είναι το παρακάτω:

```
<html>
  <head>
    <title> Embedded CSS Example</title>
    <style type="text /css">
      p { color: red; }
```

```
a { color: green; }  
</style>  
</head>  
<html>
```

Το παραπάνω κομμάτι κώδικα θα κάνει όλες τις παραγράφους p της HTML ιστοσελίδας κόκκινες και όλους τους συνδέσμους a μπλε.

3.11. Εξωτερικά Στυλ

Τα **εξωτερικά φύλλα μορφοποίησης** (External Style Sheets) είναι πολύ χρήσιμα όταν θέλουμε να μορφοποιήσουμε ένα ολόκληρο site. Τα εξωτερικά φύλλα μορφοποίησης αποθηκεύονται σε ένα ξεχωριστό αρχείο ASCII (με κατάληξη css) και είναι της μορφής:

```
p { bold; color : red }  
  
h1 { font-size : 16pt; color : green }
```

Το παραπάνω κομμάτι κώδικα θα κάνει όλες τις παραγράφους p της HTML ιστοσελίδας με έντονα και κόκκινα γράμματα και όλους τις επικεφαλίδες με πράσινα γράμματα μεγέθους 16.

3.11.1. Σύνδεση εξωτερικού στυλ με την σελίδα

Αν το αρχείο CSS έχει όνομα style.css η κάθε σελίδα πρέπει να έχει έναν δεσμό (link) προς το φύλλο στυλ, ο οποίος βρίσκεται μέσα στο τμήμα head, ως εξής :

```
<html>  
  
<head>  
  
    <title> External CSS Example</title>  
  
    <link href="style.css" rel="stylesheet" type="text/css">  
  
</head>  
  
<html>  
  
<body> ...
```

Έτσι με αυτό τον τρόπο μπορούμε να συνδέσουμε πολλές HTML σελίδες με ένα CSS αρχείο. Το CSS θα διαμορφώσει ομοιόμορφα όλες τις σελίδες.

Επίσης, πολλά CSS αρχεία μπορούν να ενσωματωθούν σε μία HTML σελίδα όπως για παράδειγμα:

```
<html>
  <head>
    <title> External CSS Example</title>
    <link href="style.css" rel="stylesheet" type="text/css">
    <link href="style_2.css" rel="stylesheet" type="text/css">
  </head>
</html>
<body> ...
```

Ακόμη ένα CSS μπορεί να κάνει εισαγωγή (import) ένα άλλο CSS αρχείο.

```
@import: url (style_2.css)
```

Τέλος, μπορούμε να ομαδοποιήσουμε τους επιλογείς. Ξεχωρίζουμε τον κάθε επιλογή με κόμμα. Στο παρακάτω παράδειγμα έχουμε ομαδοποιήσει όλα τα στοιχεία επικεφαλίδας (header elements). Το κάθε στοιχείο επικεφαλίδας θα είναι πράσινο :

```
h1, h2, h3, h4, h5, h6 { color: green }
```

3.12. Ένθετα Στυλ

Ένα **ένθετο στυλ** (inline style) χάνει πολλά από τα πλεονεκτήματα των φύλλων στυλ αναμειγνύοντας το περιεχόμενο με την παρουσίαση. Πρέπει να χρησιμοποιούμε αυτή τη μέθοδο με προσοχή, όπως όταν ένα στυλ πρόκειται να εφαρμοσθεί σε μία μοναδική εμφάνιση ενός στοιχείου.

Για να χρησιμοποιήσουμε τα inline styles χρησιμοποιούμε το χαρακτηριστικό (attribute) style στο σχετικό tag. Το χαρακτηριστικό style μπορεί να περιέχει οποιαδήποτε ιδιότητα CSS. Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να αλλάξουμε το χρώμα και το αριστερό περιθώριο μιας παραγράφου :

```
<p style="color: sienna; margin-left: 20px">
```

Αυτή είναι μια παράγραφος

```
</p>
```

3.13. Τα Πολλαπλά Φύλλα Στυλ

Αν μερικές ιδιότητες έχουν ορισθεί για τον ίδιο επιλογέα (selector) σε διαφορετικά φύλλα στυλ, οι τιμές θα κληρονομηθούν από το γενικότερο φύλλο στυλ. Για παράδειγμα, ένα εξωτερικό φύλλο στυλ έχει αυτές τις ιδιότητες για τον επιλογέα h3 :

```
h3 { color: red; text-align: left; font-size: 8pt }
```

Και ένα εσωτερικό φύλλο στυλ έχει αυτές τις ιδιότητες για τον επιλογέα h3 :

```
h3 { text-align: right; font-size: 20pt }
```

Αν η σελίδα με το εσωτερικό φύλλο στυλ συνδέεται επίσης στο εξωτερικό φύλλο στυλ, οι ιδιότητες του h3 θα είναι οι εξής :

```
color: red;
```

```
text-align: right;
```

```
font-size: 20pt
```

Το χρώμα κληρονομείται από το εξωτερικό φύλλο στυλ και η στοίχιση κειμένου και το μέγεθος γραμματοσειράς αντικαθίστανται από το εσωτερικό φύλλο στυλ.

3.14. Υποστήριξη των CSS από τους Browsers

Τα CSS υποστηρίζονται μόνο από τις τελευταίες εκδόσεις των browsers. Τόσο ο Internet Explorer 4 (υποστήριξε πρώτος, από την έκδοση 3, τα CSS) όσο και ο Netscape Navigator 4 υποστηρίζουν πλήρως τα CSS. Όσο περίεργο και αν ακούγεται, και οι δύο browsers είναι –σχεδόν- πλήρως συμβατοί. Είναι γνωστό σε όλους ότι οι δύο browsers σχεδόν ποτέ δεν συμβαδίζουν, με αποτέλεσμα να χρειάζεται άλλος τρόπος συγγραφής για τον πρώτο και άλλος για τον δεύτερο. Στα CSS όμως δεν ισχύει αυτός ο κανόνας. Ως αποτέλεσμα, τα CSS συχνά καλούνται ως η «νεκρή

ζώνη» για τους browsers τέταρτης γενιάς. Τι συμβαίνει όμως με τα CSS σε παλιότερους browsers; Ορισμένοι, όπως ο Netscape 2.0 και 3.0, είναι αρκετά «έξυπνοι» ώστε να αγνοήσουν style sheets που βρίσκονται μεταξύ του tag <STYLE>. Ορισμένοι ακόμα πιο παλιοί, όπως ο Internet Explorer 2 και ο Netscape 1, θα αγνοήσουν μεν το <STYLE> tag, αλλά θα εμφανίσουν τα περιεχόμενα μεταξύ των tags. Τα περιεχόμενα αυτά θα τυπωθούν στη κορυφή της σελίδας σε μία γραμμή – και αυτό δεν είναι καθόλου καλό. Για να λυθεί και αυτό το πρόβλημα, η λύση είναι να κάνουμε σχόλιο τα περιεχόμενα του tag με τη χρήση των <!-- --> , όπως στο ακόλουθο παράδειγμα:

```
<html>
  <head>
    <title> Κρύβοντας σε CSS σε παλιότερους browsers</title>
    <style type="text /css">
      <!--
        2στοixeio {
          position: absolute;
          z-index: 1;
          left: 30px;
          top: 30px; }
      -->
    </style>
  </head>

  <body>
    <div id=στοixeio>
      
    </div>
  </body>
</html>
```

Με αυτή την τεχνική, μπορούμε να είμαστε σίγουροι ότι δεν θα συμβεί κάποιο «ατύχημα» σε παλαιότερο browser.

3.15. Επίλογος

Ένα βασικό κομμάτι μιας σύγχρονης ιστοσελίδας είναι η αλληλεπίδραση του χρήστη με την σελίδα. Όταν ο χρήστης εκτελεί κάποια ενέργεια για την οποία έχει καθοριστεί μια λειτουργία, αντί να φορτώνεται από τη αρχή η σελίδα, εκτελείται η λειτουργία τοπικά και δίνεται η εντύπωση στο χρήστη της άμεσης ανταπόκρισης. Καθώς οι περισσότερες λειτουργίες είναι απλές, είναι άσκοπο για τον εξυπηρετητή να εκτελούνται όλες από αυτόν αν σκεφτούμε ότι σε μία σελίδα μπορεί να είναι ταυτόχρονα συνδεδεμένοι εκατοντάδες ή ακόμη και χιλιάδες χρήστες.

Έτσι, στα μέσα της δεκαετίας του 1990 άρχισε η ανάπτυξη κάποιων γλωσσών συγγραφής σεναρίων (script languages) προκειμένου να προστεθεί στις ιστοσελίδες διαδραστικότητα (interactivity). Η πρώτη client-side γλώσσα προγραμματισμού που εμφανίστηκε είναι η JavaScript και στη συνέχεια ακολούθησαν και άλλες όπως η Jscript, η VBScript κ.α. Στην εφαρμογή που αναπτύχθηκε χρησιμοποιήθηκε η JacaScript για τα client-side scripts.

ΚΕΦΑΛΑΙΟ 4

JavaScript

4.1. Εισαγωγή

Η JavaScript δημιουργήθηκε το 1995 από τον Brendan Eich, έναν μηχανικό της Netscape και εκδόθηκε με τον Netscape 2 στις αρχές του 1996. Το αρχικό της όνομα ήταν LiveScript αλλά πιθανώς για λόγους μάρκετινγκ, καθώς η εξάπλωση της Java ήταν μεγάλη, μετονομάστηκε σε JavaScript. Είναι μία γλώσσα προγραμματισμού επηρεασμένη από αντικειμενοστραφείς γλώσσες προγραμματισμού όπως η C++ και Java, αλλά η ίδια δεν είναι αντικειμενοστραφής γλώσσα προγραμματισμού.

Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεστεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript.

4.2. Java Vs JavaScript

Το όνομα της JavaScript αποτελεί μόνιμη πηγή σύγχυσης. Αν και ακούγονται ίδιες, η Java και η JavaScript δεν έχουν καμία απολύτως σχέση μεταξύ τους, ούτε στη σύνταξή τους σαν γλώσσες προγραμματισμού ούτε και στις εφαρμογές που χρησιμοποιούνται.

Πίνακας 5-1 : Διαφορές Java - JavaScript

Η Java είναι περίπλοκη στην χρήση	Η JavaScript είναι σχετικά εύκολη στην χρήση
Οι εφαρμογές σε Java είναι σχετικές αργές.	Η JavaScript εκτελείται πολύ γρήγορα

Η Java χρειάζεται το JDK (Java Developer's Kit)	Η JavaScript χρειάζεται μόνο ένας φυλλομετρητή συμβατός με JavaScript
Τα προγράμματα στην Java μεταγλωττίζονται σε εκτελέσιμα αρχεία και ενσωματώνονται στις σελίδες της html με την ετικέτα <applet>. Οι φυλλομετρητές φορτώνουν και τρέχουν το αρχείο.	Στην JavaScript τα προγράμματα ενσωματώνονται στις σελίδες της html με την μορφή σεναρίου και δεν χρειάζονται μεταγλώττιση ή διερμήνευση.
Η Java είναι πλήρης γλώσσα προγραμματισμού με πολλές δυνατότητες.	Η JavaScript είναι πιο κατάλληλη για απλές χρήσεις.
Η Java είναι προσανατολισμένη στο αντικείμενο και είναι καθαρά αντικειμενοστραφής γλώσσα προγραμματισμού.	Η JavaScript είναι βασισμένη στο αντικείμενο αλλά δεν είναι αληθινή αντικειμενοστραφής γλώσσα προγραμματισμού.
Η Java έχει αυστηρούς κανόνες στη χρήση των μεταβλητών.	Η JavaScript χαρακτηρίζεται από χαλαρότητα ως προς τις μεταβλητές.

4.3. Πως εισάγεται η JavaScript

Η JavaScript εισάγεται σε μια σελίδα HTML με την σήμανση <script> και το χαρακτηριστικό language ως εξής:

```
<script language="JavaScript">
```

Ανάμεσα στις σημάνσεις <script> και </script> γράφουμε τον κώδικα της JavaScript. Η εντολή για να εμφανίσουμε κάποιο κείμενο σε μια σελίδα είναι η document.write για παράδειγμα:

```
document.write ("Καλημέρα ! ")
```

Το σύμβολο ; είναι προαιρετικό στο τέλος των εντολών.

4.4. Που εισάγεται η JavaScript

Μπορούμε να τοποθετήσουμε ένα script στο τμήμα head ή στο τμήμα body μιας HTML σελίδας ή να γράψουμε ένα script σε ξεχωριστό αρχείο το οποίο στη συνέχεια θα ενσωματωθεί στην σελίδα.

4.4.1. Script στο τμήμα head

Τα scripts που πρέπει να εκτελεσθούν όταν κληθούν ή όταν προκαλείται ένα **συμβάν** (event) τοποθετούνται στο τμήμα head. Όταν τοποθετούμε το script στο τμήμα head τότε είναι σίγουρο ότι αυτό θα φορτωθεί πριν χρησιμοποιηθεί.

```
<html>

  <head>
    <script language="JavaScript">
      ...εντολές...
    </script>
  </head>
  <body>
    ...
  </body>

</html>
```

4.4.2. Script στο τμήμα body

Τα scripts που πρέπει να εκτελεσθούν όταν φορτώνεται η σελίδα τοποθετούνται στο τμήμα body. Όταν τοποθετούμε ένα script στο τμήμα body τότε αυτό συμμετέχει στη διαμόρφωση του περιεχομένου της σελίδας.

```
<html>

  <head>
  </head>
  <body>
    <script language="JavaScript">
      ...εντολές...
```

```
</script>  
  
...  
</body>  
</html>
```

4.4.3. Script σε ξεχωριστά αρχεία

Τα scripts εκτός από το να γράφονται απευθείας μέσα στην HTML μπορούν να γραφτούν σε ξεχωριστό με κατάληξη .js, τα οποία καλούνται και εκτελούνται από τη σελίδα HTML με τη βοήθεια της ιδιότητας src .

```
<html>  
  <head>  
  </head>  
  <body>  
    <script language="JavaScript" src="script.js">  
    </script>  
  
    ...  
  </body>  
</html>
```

4.5. Χειριστήρια συμβάντος της JavaScript

Τα **χειριστήρια συμβάντος** (event handlers) είναι αυτά απ' όπου αντλεί τη δύναμή της η JavaScript. Παρακολουθώντας τις ενέργειες του χρήστη, μπορεί να γίνει επιβεβαίωση κάποιας ενέργειας του ανάλογα με τις καταχωρήσεις ή τις κινήσεις του. Οι ενέργειες του χρήστη σε μια ιστοσελίδα (HTML page) περιορίζονται στην κίνηση του ποντικιού και στις καταχωρήσεις στοιχείων στα πεδία των φορμών.

- **onBlur** - Το συμβάν αυτό συμβαίνει όταν χάνεται η εστίαση (lost focus), δηλ. όταν απομακρυνόμαστε είτε με κλικ με το ποντίκι είτε πατώντας το πλήκτρο tab σ' ένα πεδίο κειμένου (text field) ή σε μια περιοχή κειμένου (text area) ή και σ' ένα πλαίσιο λίστας (select) μιας φόρμας.

- **onChange** - Το συμβάν αυτό συμβαίνει όταν τροποποιείται το περιεχόμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (text area) ή και ενός πλαισίου λίστας (select) μιας φόρμας, πριν ακόμα απομακρυνθούμε από το στοιχείο αυτό.
- **onClick** - Το συμβάν αυτό συμβαίνει όταν κάνουμε κλικ με το ποντίκι σ' ένα αντικείμενο, όπως είναι για παράδειγμα ένα πλήκτρο εντολής (button) ή ένα πλαίσιο ελέγχου (check box).
- **onFocus** - Το συμβάν αυτό συμβαίνει όταν κάνουμε εστίαση (focus) μέσα σ' ένα πεδίο μιας φόρμας. Μπορούμε να εστιάσουμε (receive focus) σ' ένα στοιχείο (πεδίο) μιας φόρμας είτε πατώντας επανειλημμένα το πλήκτρο tab μέχρι να φθάσουμε στο επιθυμητό πεδίο της φόρμας ή κάνοντας κλικ με το ποντίκι μέσα στο συγκεκριμένο πεδίο. Η επιλογή κάποιου κειμένου μέσα σ' ένα πεδίο δημιουργεί το event select.
- **onLoad** - Το συμβάν αυτό συμβαίνει όταν τελειώσει το φόρτωμα μιας ιστοσελίδας σ' ένα παράθυρο ή όταν τελειώσει το φόρτωμα όλων των πλαισίων (frames) που βρίσκονται μέσα σ' ένα tag <frameset>.
- **onMouseOver** - Το συμβάν αυτό συμβαίνει χώρα όταν ο δείκτης του ποντικιού τοποθετείται πάνω από ένα αντικείμενο ή και από έναν σύνδεσμο (link).
- **onSelect** - Το συμβάν αυτό συμβαίνει όταν επιλέγουμε ένα μέρος ή όλο το κείμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (textarea).
- **onSubmit** - Το συμβάν αυτό συμβαίνει όταν κάνουμε κλικ με το ποντίκι σ' ένα πλήκτρο υποβολής (submit button) για να υποβάλλουμε μια φόρμα. Το χειριστήριο συμβάντος πρέπει να επιστρέφει την τιμή true για να μπορέσουν να υποβληθούν (σταλούν) τα περιεχόμενα της φόρμας στον server. Αν επιστρέφει την τιμή false, δεν θα μπορέσει να γίνει η υποβολή των στοιχείων της φόρμας.
- **onUnload** - Το συμβάν αυτό συμβαίνει όταν φεύγουμε από ένα έγγραφο (ιστοσελίδα). Κανονικά θα πρέπει να τοποθετείται μέσα σ' ένα από τα tags <body> ή <frameset>

4.6. Συναρτήσεις της JavaScript

Μία συνάρτηση ομαδοποιεί ένα σύνολο εντολών υπό μια ονομαζόμενη υπορουτίνα. Αυτό επιτρέπει στο χρήστη να καλεί τη συνάρτηση αυτή όταν χρειαστεί τις λειτουργίες της. Οι συναρτήσεις είναι ένας βασικός τρόπος προγραμματισμού στα περισσότερα προγράμματα σε JavaScript. Φυσικά πριν την κλήση θα πρέπει να έχει γίνει ο ορισμός της συνάρτησης. Η χρήση των συναρτήσεων μπορεί να χωριστεί σε δύο λογικές κατηγορίες: ορισμός συναρτήσεων και κλήση συναρτήσεων. Η JavaScript παρέχει τη δυνατότητα δημιουργίας συναρτήσεων, οι οποίες, ουσιαστικά, είναι σενάρια τα οποία εκτελούνται όταν κληθούν με το όνομά τους, από κάποιο σημείο του HTML κώδικα, ή από κάποιο άλλο σενάριο JavaScript. Οι συναρτήσεις της JavaScript ενσωματώνονται ανάμεσα στις ετικέτες <HEAD> και </HEAD> και η δήλωσή τους γίνεται με την εντολή

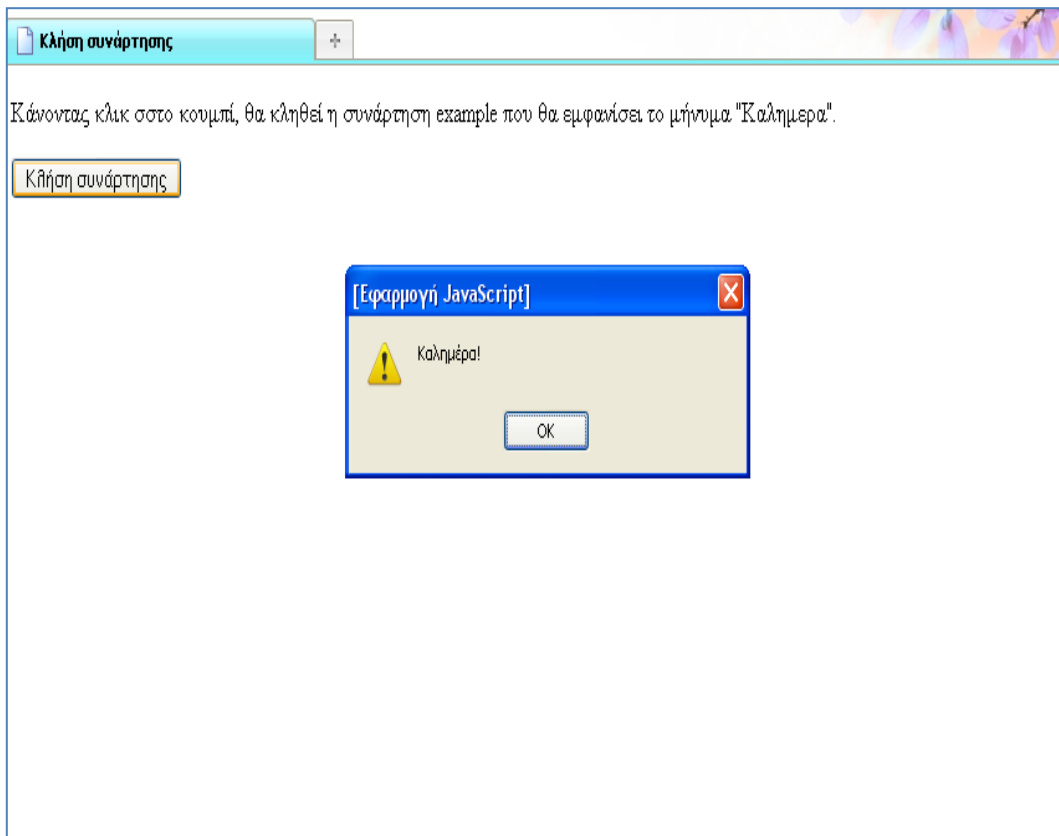
```
function Όνομα_Συνάρτησης (Ορισμα ή Ορίσματα)
```

Το όρισμα της συνάρτησης είναι προαιρετικό και παρέχει τη δυνατότητα μεταβίβασης συγκεκριμένων παραμέτρων στη συνάρτηση κατά την κλήση της. Στο ακόλουθο παράδειγμα, ορίζεται η συνάρτηση test(), η οποία στη συνέχεια καλείται πατώντας ένα button που έχει δημιουργηθεί με κώδικα HTML.

```
<html>
  <head>
    <title>Κλήση συνάρτησης</title>
    <script language="javascript">
      function test(){
        alert ("Καλημέρα!")
      }
    </script>
  </head>
  <body>
    <p>Κάνοντας κλικ στο κουμπί, θα κληθεί η συνάρτηση example που θα εμφανίσει το μήνυμα "Καλημερα".</p>
    <input type="button" value="Κλήση συνάρτησης" onClick="test()">
  </body>
```

</html>

Αποθηκεύοντας το παραπάνω κείμενο ως αρχείο HTML και ανοίγοντας στη συνέχεια το αρχείο αυτό με το φυλλομετρητή, εάν πιάσουμε το button «Κλήση συνάρτησης», καλείται η συνάρτηση example του κώδικα Javascript η οποία εμφανίζει ένα μήνυμα, όπως φαίνεται στο παρακάτω στιγμιότυπο:



Εικόνα 4-1 : Εμφάνιση μηνύματος με JavaScript

4.7. Εντολές της JavaScript

Για να εισάγουμε **σχόλια** (comments) στην JavaScript, υπάρχουν δύο τρόποι :

// αυτό είναι ένα σχόλιο μίας γραμμής

/* αυτό είναι ένα σχόλιο που μπορεί να συνεχίσει σε πολλές γραμμές και που πρέπει να ανοίξει και να κλείσει με τους ειδικούς χαρακτήρες */

Με την εντολή **break** έχουμε την δυνατότητα να κάνουμε διακοπή της εκτέλεσης ενός βρόχου πριν αυτός ολοκληρωθεί και να συνεχίσουμε την εκτέλεση των εντολών με τις εντολές που ακολουθούν μετά τον βρόχο.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() {
    var i
    var result
    for (i=1; i<20; i++) {
        result = confirm("Πατήστε το Cancel για Break")
        if (!result) {
            break;
        } // end if
    } // end for
    alert("Μόλις ο βρόχος ολοκληρώθηκε, το i είναι ίσο με : " + i)
} // end function
```

Με την εντολή **continue** μπορούμε να συνεχίσουμε την εκτέλεση ενός βρόχου με την επόμενη επανάληψη και να αγνοήσουμε έτσι τις υπόλοιπες εντολές του βρόχου.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() {
    var i = 0
    var result
    while (i<20) {
        result = confirm("Πατήστε το Cancel για Continue")
        if (!result) {
            continue;
        } // end if
        i++
    } // end while
    alert("Ο βρόχος ολοκληρώθηκε, το i είναι ίσο με : " + i)
} // end function
```


Με την εντολή **for** μπορούμε να δημιουργήσουμε ένα βρόχο, οι εντολές που ακολουθούν εκτελούνται τόσες φορές όσες ορίζεται στη συνθήκη.

Ακολουθεί ένα παράδειγμα.

```
for (i = 0; i <= 5; i++) {  
    document.write("Αριθμός : " + i)  
    document.write("<br>")  
} // end for
```

Με την εντολή **for ... in** έχουμε έναν πολύ χρήσιμο τρόπο πρόσβασης σ' όλες τις ιδιότητες ενός αντικειμένου ή σ' όλα τα στοιχεία ενός πίνακα.

Ακολουθεί ένα παράδειγμα.

```
for (i in navigator) {  
    document.write("Ιδιότητα : "+i+"<br>");  
    document.write("Τιμή : "+navigator[i]+ "<br><br>");  
} // end for
```

Με την εντολή **while** μπορούμε να δημιουργήσουμε ένα βρόχο που θα επαναλαμβάνεται όσο μια συνθήκη είναι αληθής. Ο έλεγχος της συνθήκης γίνεται στην αρχή του βρόχου και έτσι υπάρχει περίπτωση να μην εκτελεστούν καθόλου οι εντολές που περιέχει ο βρόχος αν η συνθήκη είναι από την αρχή ψευδής.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() {  
    var word = " "  
    while (word != "Exit") {  
        word = prompt("Γράψτε Exit για έξοδο", "")  
    } // end while  
    alert("Ο βρόχος τελείωσε")  
} // end function
```

Με την εντολή **do ... while** μπορούμε να δημιουργήσουμε ένα βρόχο που θα επαναλαμβάνεται όσο μια συνθήκη είναι αληθής. Ο έλεγχος της συνθήκης γίνεται στο

τέλος του βρόχου και αφού έχουν εκτελεσθεί οι εντολές που περιέχει ο βρόχος μία τουλάχιστον φορά.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() {  
    var word = " "  
    do {  
        word = prompt("Γράψτε Exit για έξοδο", "")  
    } while (word != "Exit")  
    alert("Ο βρόχος ολοκληρώθηκε")  
} // end function
```

Με την εντολή **if ... else** γίνεται εκτέλεση μιας εντολής ή μια ομάδας εντολών αν είναι αληθής μια συνθήκη και μια άλλη εντολή ή μια άλλη ομάδα εντολών αν είναι ψευδής η ίδια συνθήκη. Η χρήση της λέξης κλειδί else είναι προαιρετική.

Ακολουθεί ένα παράδειγμα.

```
var d = new Date()  
var time = d.getHours()  
if (time < 10) {  
    document.write("<b> Καλημέρα </b>")  
}  
else {  
    document.write("<b> Καλό Μεσημέρι </b>")  
}
```

Με τον **τριαδικό τελεστή σύγκρισης** μπορούμε να αποφύγουμε τη χρήση της εντολής if ... else και να έχουμε το ίδιο αποτέλεσμα αλλά μ' έναν πιο σύντομο τρόπο.

Ακολουθεί ένα παράδειγμα.

```
var sText = (nSeconds > 30) ? "περισσότερο" : "λιγότερο"
```

Στην παραπάνω εντολή, αν η τιμή της μεταβλητής nSeconds είναι μεγαλύτερη από 30, τότε θα καταχωρηθεί η τιμή (συμβολοσειρά) "περισσότερο" στη μεταβλητή

sText, ενώ αν η τιμή της μεταβλητής nSeconds είναι μικρότερη ή ίση από 30, τότε θα καταχωρηθεί η τιμή (συμβολοσειρά) "λιγότερο" στη μεταβλητή sText.

Με την εντολή **switch** μπορούμε να έχουμε ένα σύνολο από εντολές που θα εκτελεστούν ανάλογα με την τιμή μιας μεταβλητής.

Ακολουθεί ένα παράδειγμα.

```
switch (theDay) {
    case 4:
        document.write("Σήμερα είναι Πέμπτη!")
    break
    case 6:
        document.write("Σήμερα είναι Σάββατο!")
    break
    case 0:
        document.write("Σήμερα είναι Κυριακή!")
    break
    default:
        document.write("Περιμένω το Σαββατοκύριακο!")
} // end switch
```

Στο παραπάνω παράδειγμα αν η τιμή της μεταβλητής theDay είναι ίση με 4 θα εμφανισθεί ένα συγκεκριμένο μήνυμα, αν είναι ίση με 6 θα εμφανισθεί ένα άλλο μήνυμα, αν είναι ίση με 0 θα εμφανισθεί ένα άλλο μήνυμα, ενώ αν έχει μια οποιαδήποτε άλλη τιμή, θα εμφανισθεί το μήνυμα που υπάρχει στο τμήμα default της εντολής switch. Η χρήση της εντολής break μέσα σε κάθε κομμάτι της εντολής switch είναι απαραίτητη αν θέλουμε να εκτελεσθούν μόνο οι εντολές που ανήκουν στο συγκεκριμένο κομμάτι και όχι και οι υπόλοιπες.

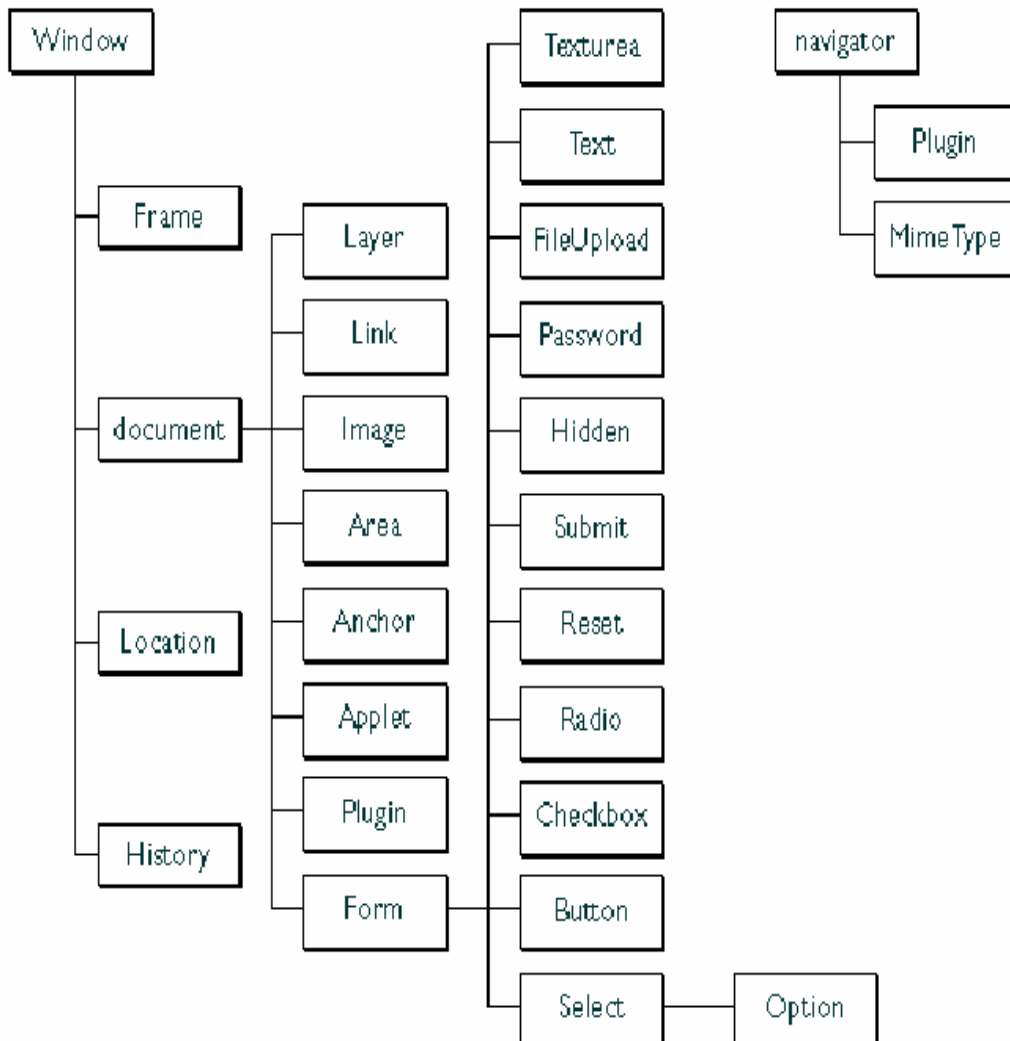
Με την εντολή **return** έχουμε τη δυνατότητα να τερματίσουμε την εκτέλεση μιας συνάρτησης και να επιστρέψουμε, αν θέλουμε, μια τιμή στο όνομα της συνάρτησης. Η εντολή χρησιμοποιείται μόνο μέσα στον κώδικα των συναρτήσεων (functions).

Με την πρόταση **var** μπορούμε να δηλώσουμε μια μεταβλητή και να της αποδώσουμε και τιμή προαιρετικά. Αν και η χρήση της πρότασης var δεν είναι

υποχρεωτική στη JavaScript, είναι καλό να τη χρησιμοποιούμε τουλάχιστον στις δηλώσεις των τοπικών μεταβλητών μιας συνάρτησης, για να μην υπάρξει έτσι σύγχυση με τις καθολικές μεταβλητές που δηλώνονται στο κυρίως σώμα ενός script.

4.8. Document Object Model (DOM)

Για να μπορέσουμε με τη χρήση της JavaScript (ή οποιαδήποτε άλλης γλώσσας) να κάνουμε αλλαγές σε μία HTML σελίδα θα πρέπει με κάποιο τρόπο τα διάφορα στοιχεία που αποτελούν την σελίδα να είναι προσπελάσιμα. Για αυτόν το λόγο υπάρχει το DOM, είναι δηλαδή ένα μοντέλο με βάση το οποίο κάθε στοιχείο της HTML σελίδας και ενός browser μπορεί να γίνει προσπελάσιμο σε μία γλώσσα όπως η JavaScript ως ένα αντικείμενο.



Εικόνα 4-2 : Τα αντικείμενα του DOM

4.8.1. Αντικείμενα του DOM, μέθοδοι & ιδιότητες

Μερικά από τα πιο βασικά αντικείμενα του DOM, οι μέθοδοι και οι ιδιότητες που σχετίζονται με αυτά είναι τα παρακάτω:

Window - Το αντικείμενο window είναι το μεγαλύτερο αντικείμενο του DOM (top-level object) και με αυτό γίνεται αναφορά στο παράθυρο του browser. Δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια σελίδα, είναι προκαθορισμένο αντικείμενο και δεν χρειάζεται να αναφέρουμε το όνομα του παραθύρου (window) όταν αναφερόμαστε στα αντικείμενά του, στις ιδιότητές του ή στις μεθόδους του. Για παράδειγμα, οι επόμενες δύο γραμμές έχουν το ίδιο αποτέλεσμα, δηλαδή την εμφάνιση ενός μηνύματος στη γραμμή κατάστασης (status line) :

```
status("Hallo!")
```

```
window.status("Hallo!")
```

Μπορούμε να ανοίξουμε ένα καινούργιο παράθυρο με τη μέθοδο open και να κλείσουμε με την μέθοδο close, για παράδειγμα :

```
NewWindow = window.open("http://www.it.teithe.gr/", "height=400",  
"width=400")
```

```
Window1 = window.close()
```

Επίσης μπορούμε να εμφανίσουμε μήνυμα με την μέθοδο alert.

```
window.alert("Καλημέρα!")
```

Document - Το αντικείμενο document δημιουργείται από τον browser όταν φορτώνεται μια σελίδα και περιέχει όλες πληροφορίες για το τρέχον έγγραφο, όπως είναι ο τίτλος του (title), το χρώμα φόντου (bgcolor) και οι φόρμες που περιέχει. Αυτές οι ιδιότητες ορίζονται μέσα στις σημάνσεις της HTML. Το αντικείμενο document είναι πολύ χρήσιμο διότι μέσω αυτού σχεδόν όλα τα στοιχεία μίας σελίδας είναι προσπελάσιμα.

History - Το αντικείμενο history περιέχει το ιστορικό επισκέψεων του browser με τις ιστοσελίδες που έχουμε επισκεφτεί νωρίτερα. Μερικές μέθοδοι είναι οι back, forward και go. Το παρακάτω παράδειγμα μας δείχνει τη χρήση των μεθόδων back και

forward με τις οποίες έχουμε τη δυνατότητα να μετακινηθούμε στην προηγούμενη-επόμενη σελίδα.

```
window.history.back()
```

```
window.history.forward()
```

Screen - Το αντικείμενο screen μας δίνει τη δυνατότητα να αναφερθούμε και να ανακτήσουμε τις ιδιότητες της οθόνης. Για παράδειγμα το παρακάτω κομμάτι κώδικα μας εμφανίζει το πλάτος της οθόνης.

```
<script type="text/javascript">  
  
    document.write("Total Width: " + screen.width);  
  
</script>
```

Form - Το αντικείμενο form είναι ιδιότητα του αντικειμένου document. Κάθε φόρμα σε ένα έγγραφο (document) είναι και ένα ξεχωριστό αντικείμενο (object) και μέσω κώδικα JavaScript μπορούμε να προσπελάσουμε τα στοιχεία της φόρμας. Το αντικείμενο form είναι ένας πίνακας (array) που δημιουργείται καθώς ορίζονται οι φόρμες (forms) μέσω των HTML tags. Αν η πρώτη φόρμα σ' ένα έγγραφο ονομασθεί form1, τότε θα μπορούμε να αναφερόμαστε σ' αυτήν ως document.form1 ή και ως document.forms[0]. Αποτελεί ιδιότητα του αντικειμένου document και σχετικές ιδιότητες είναι οι action, elements, encoding, forms, method, name και target. Σχετική μέθοδος είναι η submit και σχετικό χειριστήριο συμβάντος είναι το onSubmit.

Navigator - Το αντικείμενο αυτό περιέχει πληροφορίες για την τρέχουσα έκδοση του browser που χρησιμοποιεί ο πελάτης. Σχετικές ιδιότητες είναι οι appName, appCodeName, appVersion και userAgent. Για παράδειγμα:

```
<script type="text/javascript">  
  
    document.write("appName: " + navigator.appName);  
  
    document.write("appVersion: " + navigator.appVersion);  
  
    document.write("userAgent: " + navigator.userAgent);  
  
</script>
```

Location - Το αντικείμενο location περιέχει πλήρεις πληροφορίες URL για το τρέχον έγγραφο (document), ενώ η κάθε ιδιότητά του αναφέρεται σ' ένα διαφορετικό κομμάτι του URL. Αποτελεί ιδιότητα του αντικειμένου document. Μερικές ιδιότητες είναι οι hash, host, hostname, href, location, pathname, port, protocol, search και target. Με την χρήση της ιδιότητας href μπορούμε να αλλάξουμε τη τρέχουσα σελίδα, δηλαδή να μεταφερθούμε σε μία νέα σελίδα.

```
window.location.href = "http://www.it.teithe.gr"
```

String - Είναι μια σειρά από χαρακτήρες (characters) που ορίζεται με διπλά ή απλά εισαγωγικά, όπως για παράδειγμα :

```
firstname = "Nikos"
```

```
lastname = "Papadopoulos"
```

Η παραπάνω εντολή επιστρέφει δύο αντικείμενα string με όνομα firstname και lastname με τιμές "Nikos", "Papadopoulos". Μπορούμε να χρησιμοποιήσουμε μεθόδους που να επιστρέφουν ένα κομμάτι του string, όπως για παράδειγμα η μέθοδος `firstname.toUpperCase()` που επιστρέφει την τιμή "NIKOS". Περιέχει επίσης μεθόδους που επιστρέφουν HTML παραλλαγές του string, όπως σε μορφή **bold** και *italics*. Σχετική ιδιότητα είναι η `length` και σχετικές μέθοδοι είναι οι `anchor`, `big`, `blink`, `bold`, `charAt`, `fixed`, `fontcolor`, `fontsize`, `indexOf`, `italics`, `lastIndexOf`, `link`, `small`, `strike`, `sub`, `substring`, `sup`, `toLowerCase` και `toUpperCase`.

Math - Το αντικείμενο αυτό περιέχει ιδιότητες για μαθηματικές σταθερές και μεθόδους που χρησιμοποιούνται σε συναρτήσεις. Με αυτό το αντικείμενο μπορούμε να κάνουμε μαθηματικούς υπολογισμούς. Για παράδειγμα, η δήλωση `Math.PI` αντιπροσωπεύει την τιμή του π σε μια εξίσωση ενώ με την μέθοδο `sin()` μπορούμε να υπολογίσουμε τα ημίτονα. Σχετικές ιδιότητες είναι οι `E`, `LN10`, `LN2`, `PI`, `SQRT1_2`, `SQRT2` και σχετικές μέθοδοι είναι οι `abs`, `acos`, `asin`, `atan`, `ceil`, `cos`, `exp`, `floor`, `log`, `max`, `min`, `pow`, `random`, `round`, `sin`, `sqrt` και `tan`.

Date - Αντικαθιστά έναν κανονικό τύπο ημερομηνίας (date type). Αν και δεν έχει καθόλου ιδιότητες (properties), διαθέτει πολλές μεθόδους (methods). Οι μέθοδοι για να λαμβάνουμε και να ορίζουμε πληροφορίες για την ημερομηνία και την ώρα

χωρίζονται σε τέσσερις κατηγορίες : set, get, to και parse/UTC. Εκτός από τις ημέρες του μήνα, όλες οι άλλες αριθμητικές εκφράσεις ημερομηνιών ξεκινούν από το 0. Οι μήνες ξεκινούν από το 0 (Ιανουάριος) έως και το 11 (Δεκέμβριος). Σχετικές μέθοδοι είναι οι getDate, getDay, getHours, getMinutes, getMonth, getSeconds, getTime, getTimezoneOffset, getYear, parse, setDate, setHours, setMinutes, setMonth, setSeconds, setTime, setYear, toGMTString, toLocaleString και toString.

Button - Είναι ένα αντικείμενο που αποτελεί στοιχείο μιας φόρμας και μπορεί να χρησιμοποιηθεί για να εκτελεσθεί μια ενέργεια (action). Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι η name και η value, σχετική μέθοδος είναι η click και σχετικό χειριστήριο συμβάντος είναι το onClick.

4.8.2. Δημιουργία Αντικειμένου

Για να δημιουργήσουμε ένα δικό μας αντικείμενο, υπάρχουν οι εξής λέξεις κλειδιά :

- **new**, δημιουργούμε ένα καινούργιο αντικείμενο.
- **this**, αναφερόμαστε στο τρέχον αντικείμενο.
- **with**, κάνουμε προεπιλεγμένο ένα αντικείμενο σε μια ομάδα εντολών για να μην είμαστε αναγκασμένοι να το γράφουμε συνέχεια.

Για να μπορέσουμε να δημιουργήσουμε ένα δικό μας αντικείμενο, γράφουμε πρώτα μια συνάρτηση δημιουργίας αντικειμένων (constructor), με την οποία δίνουμε τιμές στις ιδιότητες του αντικειμένου με τη λέξη κλειδί this, ως εξής :

```
function FullName (firstname, lastname) {  
  
    this.firstname = firstname;  
  
    this.lastname = lastname;  
  
}
```

Μετά, για να μπορέσουμε να δημιουργήσουμε ένα στιγμιότυπο (instance) αυτού του αντικειμένου, χρησιμοποιούμε λέξη κλειδί new, ως εξής :

```
John01 = new FullName("Nikos", "Papadopoulos");
```


4.8.3. Τροποποίηση Αντικειμένου

Με τη λέξη κλειδί prototype μπορούμε να προσθέσουμε κι άλλες ιδιότητες στο πρωτότυπο ενός αντικειμένου, δηλ. στον ορισμό του, και έτσι όλα τα αντικείμενα αυτού του τύπου που θα δημιουργήσουμε θα περιλαμβάνουν τις ιδιότητες αυτές.

Για παράδειγμα, για να προσθέσουμε την ιδιότητα fathersname στο αντικείμενο FullName που δημιουργήσαμε προηγουμένως, γράφουμε την εξής εντολή :

```
FullName.prototype.fathersname = "George";
```

4.9 Συνδυάζοντας HTML, JavaScript και CSS

Έχουμε δει παραπάνω ότι η JavaScript, και τα διαδοχικά φύλλα στυλ μπορούν να ενσωματωθούν σε ένα αρχείο HTML είτε απευθείας είτε ως εξωτερικά αρχεία. Παρακάτω θα δούμε ένα ολοκληρωμένο παράδειγμα το οποίο συνδυάζει την HTML, την JavaScript και τα διαδοχικά φύλλα στυλ.

Το αποτέλεσμα είναι μία σελίδα η οποία καλύπτει τις εξής απαιτήσεις:

Η σελίδα έχει φόντο κίτρινο, η προεπιλεγμένη γραμματοσειρά είναι Monotype Corsiva και το μέγεθος των γραμμάτων είναι x-large.

Η σελίδα περιλαμβάνει 2 εικόνες σε μικρογραφίες. Εάν γίνει κλικ στην αριστερή εικόνα το μέγεθος της αυξάνεται και μετά από 2 δευτερόλεπτα επιστρέφει στο αρχικό της μέγεθος.

Η σελίδα δίνει τη δυνατότητα αλλαγής του χρώματος φόντου με την τοποθέτηση του ποντικιού σε ανάλογα χρωματισμένο κελί ενός πίνακα.

Η σελίδα περιλαμβάνει μια φράση η οποία θα κινείται προς τα δεξιά και θα γίνεται κόκκινη όσο μετακινούμε το ποντίκι πάνω της.

Η σελίδα περιλαμβάνει μία φόρμα με ένα drop-down list από το οποίο ο χρήστης μπορεί να επιλέξει την πόλη κατοικίας του. Μόλις ο χρήστης επιλέξει η επιλογή του εμφανίζεται με alert box.

Ζητούμενο 1) Το ζητούμενο μπορεί να υλοποιηθεί χρησιμοποιώντας τις ιδιότητες φόντου των CSS. Δημιουργώντας ένα εσωτερικό φύλλο στυλ, αρκεί να ενσωματώσουμε στην ιστοσελίδα το παρακάτω τμήμα:

```
<style>

    body {

        background-color:#FFFFCC;

        font-family:"Monotype Corsiva";

        font-size:x-large;

    }

</style>
```

Ζητούμενο 2) Για αρχή εισάγουμε στην ιστοσελίδα τις δύο εικόνες. Εφόσον όμως απαιτείται η αλλαγή της μιας από τις εικόνες με το κλικ του ποντικιού, θα χρειαστεί να δώσουμε σε αυτές ένα μοναδικό χαρακτηριστικό ώστε να μπορέσουμε στη συνέχεια με τη βοήθεια του συμβάντος onclick να καλέσουμε μια συνάρτηση JavaScript η οποία θα αλλάζει το μέγεθος της αριστερής εικόνας. Οι δύο εικόνες εισάγονται στην σελίδα με το ακόλουθο τμήμα κώδικα:

```



```

Γίνεται χρήση του χειριστηρίου συμβάντων onclick το οποίο παραπέμπει στη συνάρτηση changeSize() της JavaScript, την οποία θα χρειαστεί να δημιουργήσουμε. Η συνάρτηση JavaScript αλλάζει το μέγεθος της πρώτης εικόνας και καλεί μέσω της συνάρτησης setTimeout την συνάρτηση returnSize() με καθυστέρηση 2000 milliseconds η οποία επαναφέρει την εικόνα στο αρχικό της μέγεθος.

Ο κώδικας των παραπάνω συναρτήσεων παρουσιάζεται παρακάτω:

```
function changeSize(obj) {

    document.getElementById("1").height=500;
```

```
var t=setInterval("returnSize()", 2000 );  
  
}  
  
function returnSize() {  
  
    document.getElementById("1").height=70;  
  
}
```

Ζητούμενο 3) Για αρχή χρειαζόμαστε έναν πίνακα με τρία πεδία, το καθένα με διαφορετικό χρώμα, έστω πράσινο, ροζ και γαλάζιο. Στη συνέχεια, σε κάθε πεδίο με το χειριστήριο συμβάντος onmouseover καλούμε τη JavaScript συνάρτηση bgChange, η οποία θα πρέπει να δέχεται ως όρισμα το χρώμα που θέλουμε να παίρνει το φόντο. Ο πίνακας ορίζεται ως εξής:

```
<table border="1" width="100%" height="34" bgcolor="silver">  
  
    <tr>  
  
        <td onmouseover="bgChange('#99FF99')"  
onmouseout="bgChange('#FFFFCC)"    bgcolor="#99FF99"  
width="33%" align="center" height="28"></td>  
  
        <td onmouseover="bgChange('#FF99FF')"  
onmouseout="bgChange('#FFFFCC)"    bgcolor="#FF99FF"  
width="33%" align="center" height="28"></td>  
  
        <td onmouseover="bgChange('#CCFFFF')"  
onmouseout="bgChange('#FFFFCC)"    bgcolor="#CCFFFF"  
width="34%" align="center" height="28"></td>  
  
    </tr>  
  
</table>
```

Η JavaScript συνάρτηση bgChange είναι μία απλή συνάρτηση, όπως φαίνεται στο παρακάτω πεδίο.

```
function bgChange(bg){
```

```
document.body.style.background=bg;  
  
}
```

Ζητούμενο 4) Για αρχή ορίζεται μια φράση στην οποία θα δώσουμε την ονομασία Mytext και συνδέεται με το συμβάν onmouseover με την JavaScript συνάρτηση moveright. Ο HTML ορισμός της φράσης γίνεται:

`<h3 id="Mytext" onmousemove="moveright()">Μετακινείτε το ποντίκι πάνω απ' αυτό το κείμενο για να μετακινηθεί </h3>`. Η ακόλουθη JavaScript συνάρτηση χρησιμοποιεί τις ιδιότητες position και left του αντικειμένου Mytext που ορίσαμε, αλλάζοντας συνέχεια την τιμή της ιδιότητας left, η οποία αυξάνει κατά 1 κάθε φορά που κινούμε το ποντίκι πάνω από τη φράση.

```
var i=1  
  
function moveright(){  
  
    Mytext.style.position="relative"  
  
    Mytext.style.left=i  
  
    i++  
  
}
```

Ζητούμενο 5) Για αρχή πρέπει να εισάγουμε την φόρμα με το drop-down list στην σελίδα. Αυτό γίνεται με τον παρακάτω κώδικα:

```
<form> Διάλεξε την πόλη κατοικίας σου  
  
    <select name="selection" onChange="writeselection(this)">  
        <option value ="Αθήνα"> Αθήνα</option>  
        <option value ="Θεσσαλονίκη"> Θεσσαλονίκη</option>  
        <option value ="Ρόδος"> Ρόδος</option>  
        <option value ="Λάρισα"> Λάρισα</option>  
        <option value ="Ηράκλειο"> Ηράκλειο</option>  
        <option value ="Κατερίνη"> Κατερίνη</option>  
        <option value ="Καλαμάτα"> Καλαμάτα</option>
```

```
<option value ="Ξάνθη"> Ξάνθη</option>
</select>
</form>
```

Όταν ο χρήστης επιλέξει κάποια από τις διαθέσιμες επιλογές μέσω του χειριστηρίου συμβάντος onChange καλείται η συνάρτηση writeselection η οποία εμφανίζει το alert box με την επιλογή του χρήστη.

Ο τελικός κώδικας της σελίδας

Συνδυάζοντας όλα τα τμήματα κώδικα των ζητημάτων 1 – 4, προκύπτει ο παρακάτω συνολικός κώδικας:

```
<html>
<head>
<title>Συνδυάζοντας HTML, CSS και JavaScript</title>

<style>
    body {
        background-color:#FFFFCC;
        font-family:"Monotype Corsiva";
        font-size:x-large;
    }
</style>

<script language="JavaScript">
function changeSize(obj) {
    document.getElementById("1").height=500;
    var t=setInterval("returnSize()", 2000 );
}
function returnSize() {
    document.getElementById("1").height=70;
}
function bgChange(bg){
    document.body.style.background=bg;
}
function writeselection(obj){
    alert(' Διάλεξες την ' + obj.value);
}
var i=1;
function moveright() {
    Mytext.style.color ="red";
```

```
        Mytext.style.position="relative";
        Mytext.style.left=i;
        i++;
    }
</script>

</head>

<body>


<p align="center">Κινώντας το ποντίκι στα κελιά του πίνακα αλλάζει το
Χρώμα Φόντου</p>
<table border="1" width="100%" height="34" bgcolor="silver">
    <tr>
        <td onmouseover="bgChange('#99FF99')"
            onmouseout="bgChange('#FFFFCC')" bgcolor="#99FF99"
            width="33%" align="center" height="28"></td>
        <td onmouseover="bgChange('#FF99FF')"
            onmouseout="bgChange('#FFFFCC')" bgcolor="#FF99FF"
            width="33%" align="center" height="28"></td>
        <td onmouseover="bgChange('#CCFFFF')"
            onmouseout="bgChange('#FFFFCC')" bgcolor="#CCFFFF"
            width="34%" align="center" height="28"></td>
    </tr>
</table>

<h3 id="Mytext" onmousemove="moveright()">Μετακινείστε το ποντίκι
πάνω απ' αυτό το κείμενο για να μετακινηθεί</h3>

<form> Διάλεξε την πόλη κατοικίας σου
<select name="selection" onChange="writeselection(this)">
    <option value="Αθήνα">Αθήνα</option>
    <option value="Θεσσαλονίκη">Θεσσαλονίκη</option>
    <option value="Ρόδος">Ρόδος</option>
    <option value="Λάρισα">Λάρισα</option>
    <option value="Ηράκλειο">Ηράκλειο</option>
    <option value="Κατερίνη">Κατερίνη</option>
    <option value="Καλαμάτα">Καλαμάτα</option>
    <option value="Ξάνθη">Ξάνθη</option>
</select>
</form>

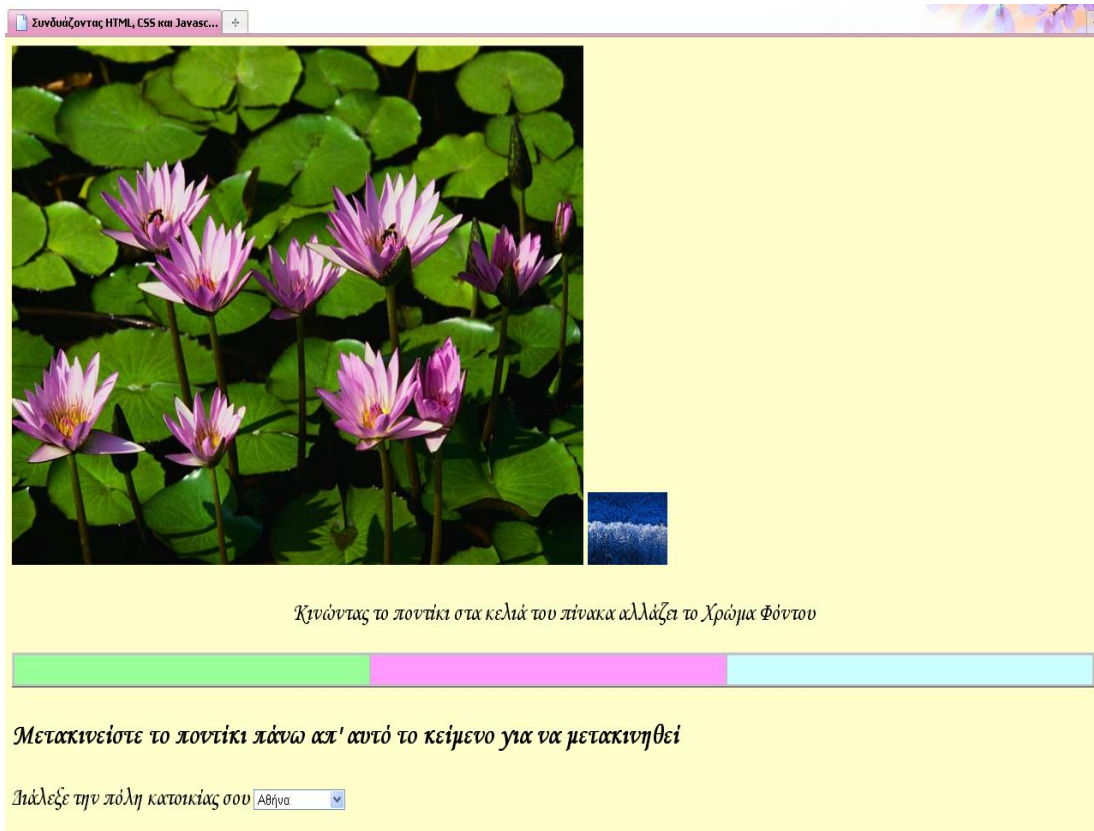
</body>
```

</html>

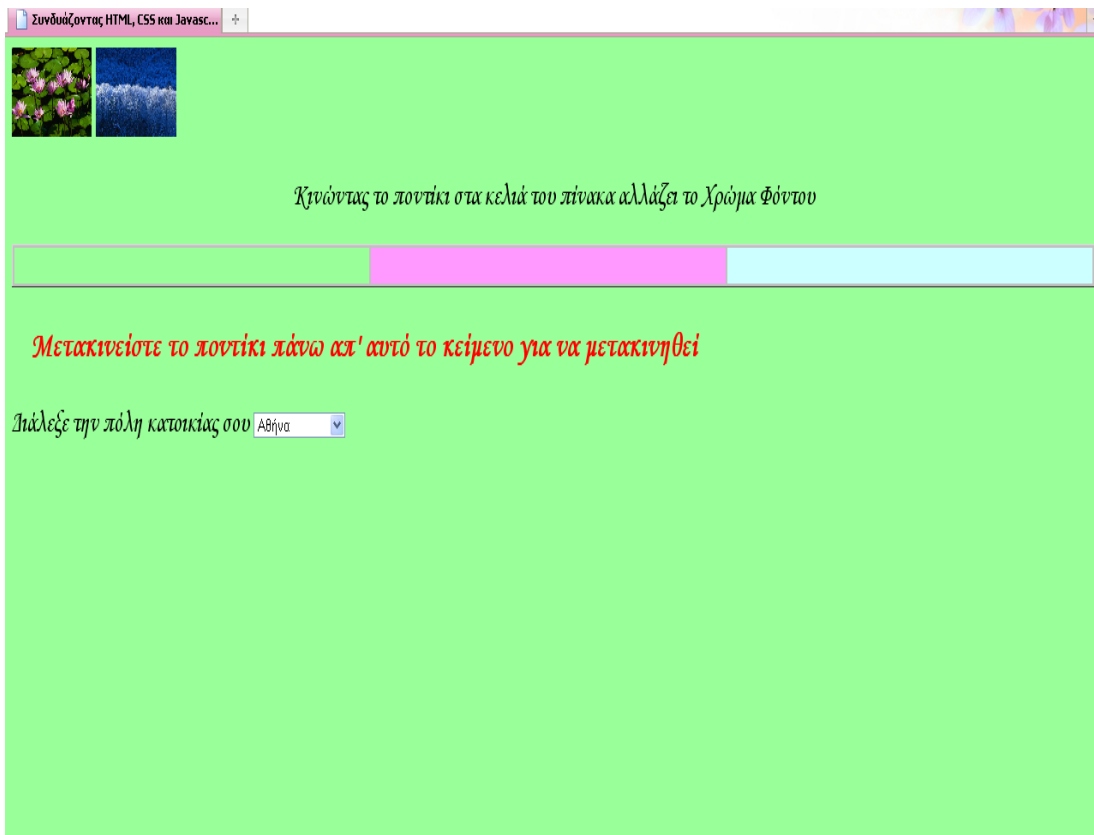
Σώζοντας τον παραπάνω κώδικα ως αρχείο .htm στον ίδιο φάκελο με τις εικόνες Water lilies.jpg, Winter.jpg, μπορούμε να πειραματιστούμε με τις λειτουργίες της σελίδας μας χρησιμοποιώντας κάποιον browser. Κάποια χαρακτηριστικά στιγμιότυπα είναι τα ακόλουθα:



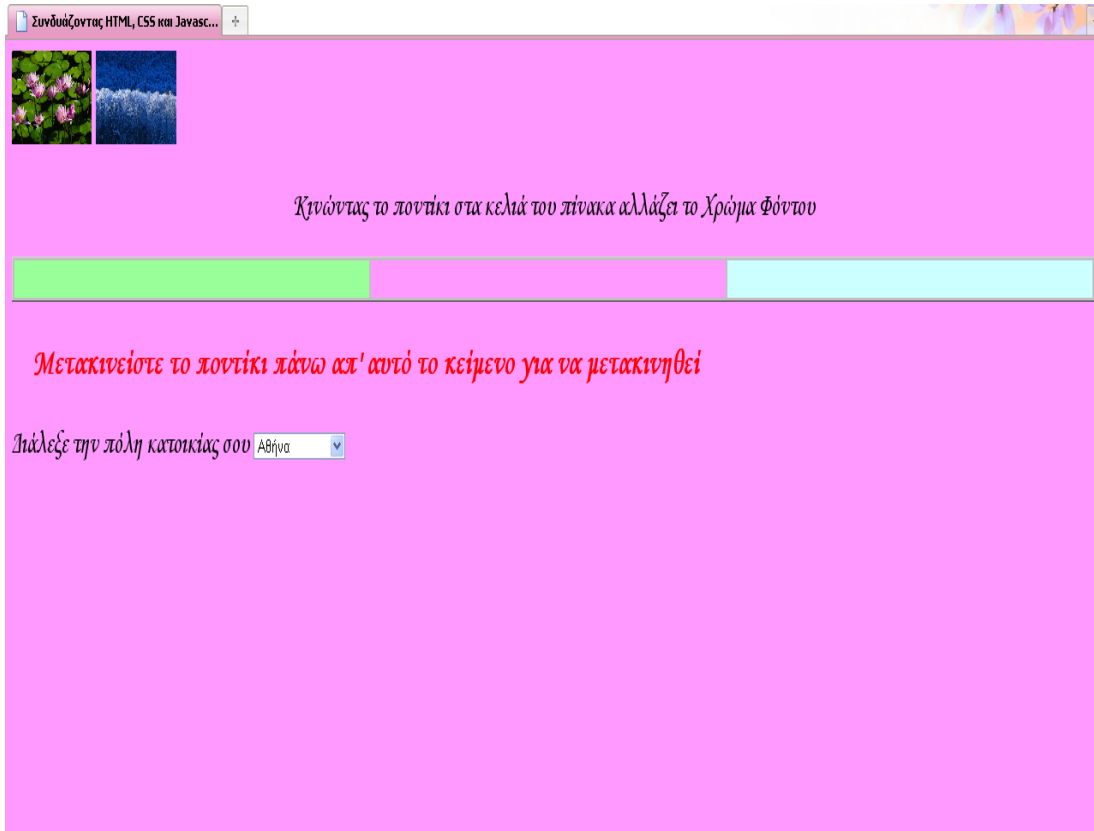
Εικόνα 4-3 : Η αρχική σελίδα



Εικόνα 4-4 : Κάνοντας κλικ στην εικόνα



Εικόνα 4-5 : Αλλαγή φόντου



Εικόνα 4-6 : Αλλαγή φόντου2



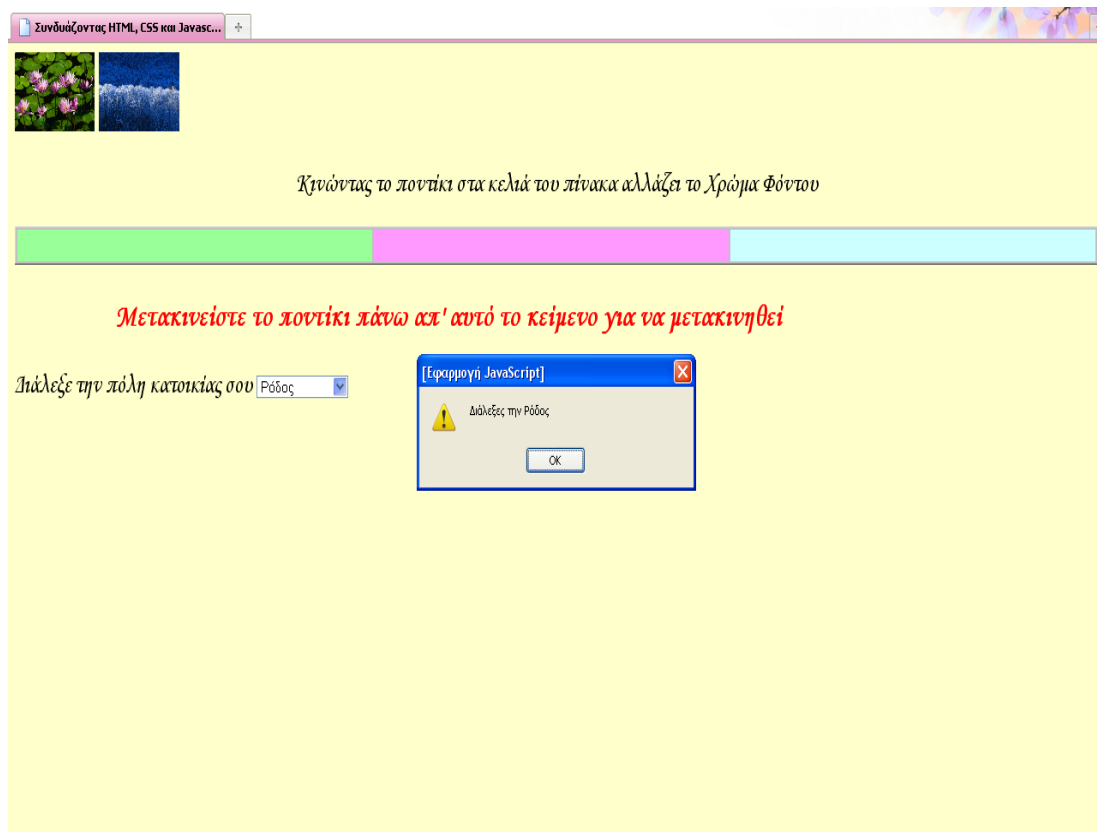
Εικόνα 4-7 : Αλλαγή φόντου3



Εικόνα 4-8 : Μετακίνηση κειμένου



Εικόνα 4-9 : Drop down list



Εικόνα 4-10 : Εμφάνιση μηνύματος

4.10. Επίλογος

Σήμερα στο Web, αυτό που κυριαρχεί είναι το περιεχόμενο. Μία σελίδα για να τραβάει το ενδιαφέρον των επισκεπτών συνεχώς, πρέπει να περιέχει περιεχόμενο που ανανεώνεται συχνά. Χρησιμοποιώντας την HTML, την JavaScript, τη Δυναμική HTML, μπορούμε να δημιουργήσουμε μια ιστοσελίδα με εντυπωσιακή εμφάνιση, όταν όμως πρέπει να προσθέσουμε στη σελίδα πληροφορίες η αντιγραφή αρχεία του Word μέσα σε σημάνσεις της HTML δεν είναι αποτελεσματικός τρόπος. Η λύση σ' αυτό το πρόβλημα είναι το database-driven site design. Πετυχαίνοντας τον πλήρη διαχωρισμό ανάμεσα στη σχεδίαση του site και το περιεχόμενο που θέλουμε να παρουσιάσουμε μέσα σ' αυτό, μπορούμε να δουλέψουμε με το ένα χωρίς να μας απασχολεί το άλλο.

Για να δημιουργήσουμε ένα database-driven Web site, χρησιμοποιούμε μια γλώσσα συγγραφής σεναρίων στην πλευρά του διακομιστή, μια σχεσιακή βάση δεδομένων και έναν εξυπηρετητή ιστού.

Κεφάλαιο 5

Apache, MySQL & PHP

5.1. Εισαγωγή

Όπως αναφέρθηκε παραπάνω για να δημιουργήσουμε ένα database-driven Web site, χρησιμοποιούμε μια γλώσσα συγγραφής σεναρίων στην πλευρά του διακομιστή, μια σχεσιακή βάση δεδομένων και έναν διακομιστή. Στην εφαρμογή χρησιμοποιήθηκε ως διακομιστής ο Apache HTTP Web Server, ως σχεσιακή βάση δεδομένων η MySQL και τέλος η PHP ως η γλώσσα συγγραφής σεναρίων από την πλευρά του διακομιστή.

5.2. Apache HTTP Web server

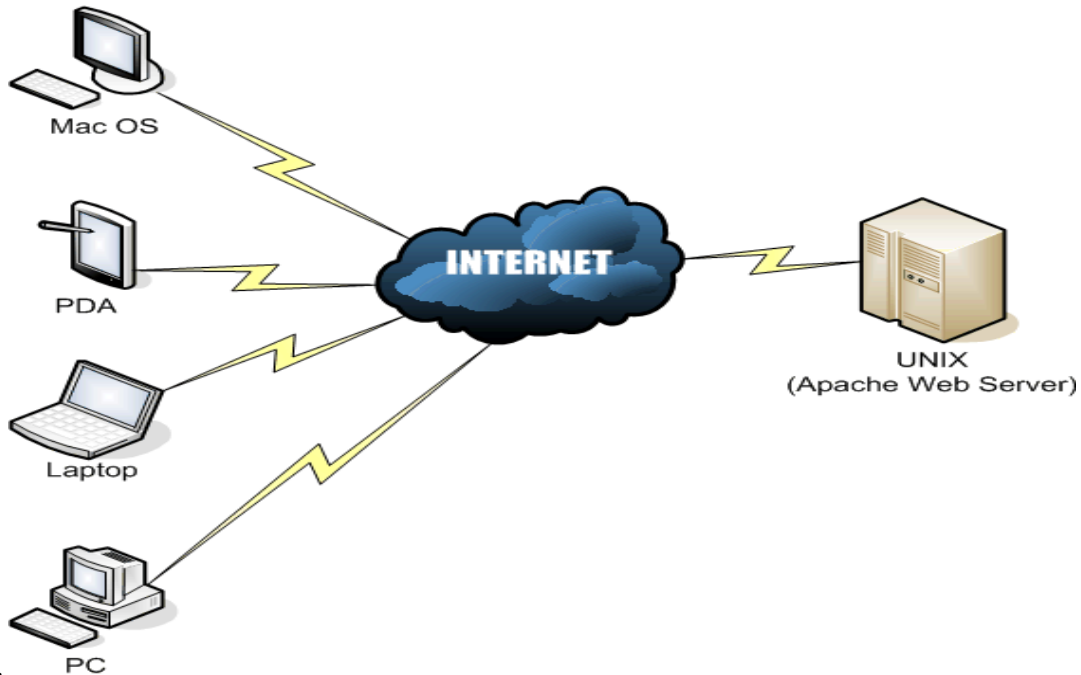
5.2.1. Τι είναι ένας Web Server

Κεντρικός υπολογιστής ή αλλιώς εξυπηρετητής ή **διακομιστής** (Web server) είναι ο υπολογιστής που ρόλος του είναι να παρέχει διάφορες υπηρεσίες σε υπολογιστές πελατών στον παγκόσμιο Ιστό. Είναι πιο αναβαθμισμένος όσον αφορά το λογισμικό και υλικό του εξοπλισμό, σε σχέση με κοινούς υπολογιστές.

Τα κύρια χαρακτηριστικά του είναι:

- Υποστηρίζει επεξεργαστές που χρησιμοποιούνται για την επεξεργασία των πολλαπλών δεδομένων που δέχεται.
- Υποστηρίζει μεγάλης ταχύτητας σκληρούς δίσκους
- Υποστηρίζει ταχύτερες μνήμες.

Οι κεντρικοί υπολογιστές συνήθως συνδέονται με συστήματα διπλής τροφοδοσίας ή παροχή αδιάλειπτης ενέργειας (UPS) για μεγαλύτερη αξιοπιστία και σιγουριά στις παρεχόμενες υπηρεσίες.



Εικόνα 5-1 : Η λειτουργία ενός web server

5.2.2. Hypertext Transfer Protocol (HTTP)

Το πρωτόκολλο HTTP καθορίζει τον τρόπο επικοινωνίας στο διαδίκτυο μεταξύ των διακομιστών και των πελατών (servers - clients). Είναι ένα γενικό, αντικειμενοστραφές πρωτόκολλο που μεταβιβάζει πληροφορία μεταξύ των διακομιστών και των πελατών. Ξεκίνησε από την έκδοση HTTP/0.9 κατά τη διάρκεια της πρώιμης ανάπτυξης του διαδικτύου και ακολούθησε η έκδοση HTTP/1.0 το 1995. Η πιο πρόσφατη έκδοσή του, HTTP/1.1, παρέχει περισσότερη λειτουργικότητα και υποστηρίζει πολλαπλές συναλλαγές μεταξύ πελάτη και διακομιστή κατά τη διάρκεια της ίδιας αίτησης.

5.2.3. Apache HTTP Web server

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται έναν ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με ένα διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους, 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 π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET . Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

5.3. MySQL

5.3.1. Είδη βάσεων δεδομένων

Υπάρχουν δύο ειδών βάσεων δεδομένων:

➤ Αντικειμενοστραφείς βάσεις δεδομένων

Πολύπλοκες δομές δεδομένων, οι οποίες όταν αναπαρασταθούν γραφικά παρομοιάζονται σαν δίκτυα ή σαν οντολογίες.

➤ Σχεσιακές βάσεις δεδομένων

Οι δομές δεδομένων αναπαριστώνται με δισδιάστατους πίνακες. Δύο ή περισσότεροι πίνακες μπορούν να συσχετισθούν μεταξύ τους ώστε να προκύψει μια νέα δομή δεδομένων.

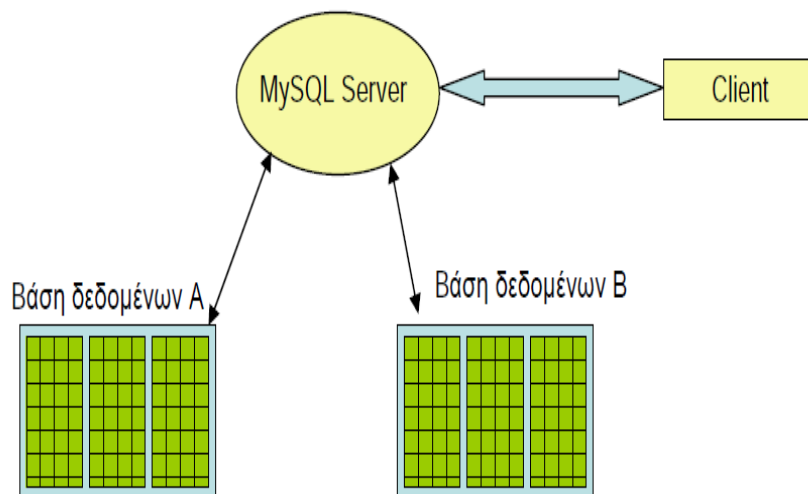
Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (relational database management system) ή RDBMS. Με τη χρήση της MySQL είναι εύκολη η πρόσβαση σ' αυτές τις πληροφορίες χρησιμοποιώντας μια γλώσσα

συγγραφής σεναρίων στην πλευρά του διακομιστή (server-side scripting languages), όπως είναι η PHP.

Η MySQL αναπτύσσεται, διανέμεται, και υποστηρίζεται από την εταιρεία MySQL AB, η οποία είναι μια εμπορική επιχείρηση ανάπτυξης ανοικτού λογισμικού. Ανοικτό λογισμικό, σημαίνει ότι είναι δυνατό για οποιονδήποτε να χρησιμοποιήσει και να τροποποιήσει το λογισμικό, χωρίς καμία οικονομική επιβάρυνση. Ο διακομιστής MySQL ελέγχει την πρόσβαση στα δεδομένα μας για να διασφαλίσει ότι πολλοί χρήστες θα μπορούν να δουλεύουν ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και για να διασφαλίσει ότι μόνο οι πιστοποιημένοι χρήστες θα μπορούν να έχουν πρόσβαση.

5.3.2. Χαρακτηριστικά της MySQL

Η MySQL λειτουργεί με το μοντέλο Πελάτη/Εξυπηρετητή (Client/Server). Ο εξυπηρετητής δέχεται αιτήματα από τους πελάτες σχετικά με τη διαχείριση μίας ή περισσότερων βάσεων δεδομένων και προβαίνει στις απαιτούμενες ενέργειες, όπως φαίνεται παρακάτω.



Εικόνα 5-2 : Το μοντέλο Πελάτη/Εξυπηρετητή

Οι κύριες λειτουργίες διαχείρισης βάσεων δεδομένων τις οποίες υλοποιεί ο εξυπηρετητής μετά από αίτηση του πελάτη, δε διαφέρουν σημαντικά από τις αντίστοιχες λειτουργίες των άλλων εφαρμογών βάσεων δεδομένων και περιλαμβάνουν τη δημιουργία-διαγραφή βάσεων δεδομένων, την εισαγωγή-

τροποποίηση-διαγραφή πινάκων (tables) και πεδίων (fields), την εισαγωγή-τροποποίηση-διαγραφή εγγραφών (records) και τέλος την ανάκτηση δεδομένων από τη βάση με τη χρήση συγκεκριμένων κριτηρίων.

Για την υλοποίηση των παραπάνω λειτουργιών, η MySQL χρησιμοποιεί την SQL (Δομημένη Γλώσσα Ερωτήσεων – Structured Query Language). Η SQL είναι η ευρύτερα χρησιμοποιούμενη τυποποιημένη γλώσσα πρόσβασης στις βάσεις δεδομένων και αποτελείται από εντολές οι οποίες επιτρέπουν την ανάκτηση και ενημέρωση δεδομένων σε μια βάση. Εκτός από τη MySQL, η γλώσσα SQL συνεργάζεται με άλλα προγράμματα βάσεων δεδομένων όπως είναι η Access, Microsoft SQL Server, Oracle, Sybase και κ.α.

5.3.3. Οι βάσεις δεδομένων της MySQL

Μια βάση δεδομένων της MySQL αποτελείται από έναν ή περισσότερους πίνακες, ο καθένας από τους οποίους περιέχει μια λίστα από δεδομένα. Αν, για παράδειγμα, θελήσουμε να δημιουργήσουμε μια βάση δεδομένων για να αποθηκεύσουμε έναν τηλεφωνικό κατάλογο, θα μπορούσαμε να κατασκευάσουμε έναν πίνακα με το όνομα *ΕΠΑΦΕΣ*. Η μορφή που θα μπορούσε να έχει αυτός ο πίνακας φαίνεται παρακάτω:

Πίνακας 6-1 : Μορφή πίνακα βάσης δεδομένων

Επαφές			
id	Όνομα	Επώνυμο	Τηλέφωνο
1	Γιάννη	Παπαδόπουλος	555555
2	Βούλα	Ζαφειρίου	222222

5.3.4. Τύποι δεδομένων

Κάθε πίνακας αποτελείται από πεδία, τα οποία αντιπροσωπεύουν τα γνωρίσματα. Για κάθε πεδίο πρέπει να οριστεί ο τύπος τους. Θα πρέπει πάντα να χρησιμοποιείται ο τύπος δεδομένων που χρειάζεται για να αποθηκευτούν οι τιμές που μπορεί να πάρει

το γνώρισμα. Επίσης πρέπει να ορίζεται και το μέγεθος του πεδίου, σύμφωνα με τις ανάγκες.

Αριθμητικοί

- **int** - ακέραιος μήκους 11 ψηφίων
- **tinyint** - ακέραιος μήκους 4 ψηφίων
- **smallint** - ακέραιος μήκους 5 ψηφίων
- **mediumint** - ακέραιος μήκους 9 ψηφίων
- **bigint** - ακέραιος μήκους 20 ψηφίων
- **float(m,d)** - κινητής υποδιαστολής m ψηφίων και d δεκαδικών (μέχρι 24), είναι πάντα προσημασμένος
- **double(m,d)** - διπλής ακρίβειας m ψηφίων και d δεκαδικών (μέχρι 53), είναι πάντα προσημασμένος

Ημερομηνίες & Ώρες

- **date** - Τιμή ημερομηνίας με μορφή: EEEE-MM-HH
- **datetime** - Ημερομηνία και ώρα με μορφή: EEEE-MM-HH ΩΩ:ΛΛ:ΔΔ
- **time** - Ώρα με μορφή: ΩΩ:ΛΛ:ΔΔ
- **year(m)** - Αποθηκεύει έτος σε διψήφια ή τετραψήφια μορφή. Στη διψήφια μορφή περιλαμβάνονται τα έτη από 1970-2069. Στην τετραψήφια από 1901-2155.
- **timestamp** - Χρονική ένδειξη με μορφές: 14(EEEEMMHHΩΩΛΛΔΔ) 12(EEMMHHΩΩΛΛΔΔ) 8(EEEEMMHH) 6(EEMMHH)

Αλφαριθμητικά

- **char(m)** - Αλφαριθμητικό σταθερού μεγέθους μεταξύ 1-255 χαρακτήρες. Αν η τιμή δεν φτάνει το μέγεθος συμπληρώνεται με κενά ως το προσδιοριζόμενο μήκος.
- **varchar(ma)** - Αλφαριθμητικό μεταβλητού μεγέθους, από 1-255 χαρακτήρες.
- **text** - Πεδίο κειμένου με μέγιστο μήκος 65535
- **tinytext** - Πεδίο κειμένου με μέγιστο μήκος 255
- **mediumtext** - Πεδίο κειμένου με μέγιστο μήκος 16777215
- **longtext** - Πεδίο κειμένου με μέγιστο μήκος 4294967295

- **enum** – Απαρίθμηση, δηλαδή λίστα. Όταν ορίζεται ένας τύπος enum δημιουργείται μία λίστα στοιχείων από τα οποία μπορεί να επιλεγθεί μία τιμή (ή μπορεί να είναι και null).
- **set** – Παρόμοιος τύπος με τον enum, με την διαφορά ότι η τιμή ενός γνωρίσματος μπορεί να είναι ένα σύνολο με περισσότερες από μία τιμές του πεδίου ορισμού.

5.4. PHP

Η PHP είναι μια γλώσσα προγραμματισμού, που σχεδιάστηκε με σκοπό τη δημιουργία δυναμικών σελίδων και είναι επισήμως γνωστή ως: HyperText preprocessor. Η PHP είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων. Με τον όρο δυναμική εννοείται μια ιστοσελίδα, που αλλάζει αυτόματα, ανάλογα με τα στοιχεία του χρήστη όπως το λειτουργικό του σύστημα, η διεύθυνση ip του κ.ά. Ανταγωνιστικές ως προς την PHP γλώσσες προγραμματισμού είναι οι εξής: ASP (Active Server Pages) της εταιρείας Microsoft, CFML (ColdFusion Markup Language) της εταιρείας Allaire και JSP (JavaServer Pages) της εταιρείας Sun.

5.4.1. Εξέλιξη της PHP μέχρι σήμερα

Η Php δημιουργήθηκε το 1994 από έναν φοιτητή, τον Rasmus Lerdorf ο οποίος δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα PHP.cgi και το ενσωμάτωσε στην προσωπική του ιστοσελίδα. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter και το κύριο χαρακτηριστικό αυτής της έκδοσης ήταν ότι ο PHP/FI parser ήταν γραμμένος κυρίως με το χέρι και έτσι προέκυπταν συχνά λάθη. Ο όρος parser (αναλυτής) αναφέρεται στον μηχανισμό ο οποίος δέχεται ένα script και το μετατρέπει σε κάτι που μπορεί να κατανοήσει ο υπολογιστής.

Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ

αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

5.4.2. Ενσωμάτωση της PHP

Οι περισσότερες PHP σελίδες έχουν PHP και HTML κώδικα μαζί. Όταν ο server διαβάζει μια PHP σελίδα ψάχνει να δει από πού θα αρχίσει να διαβάζει την σελίδα ως PHP και από πού ως HTML. Η σελίδα διαβάζεται ως PHP όταν συναντήσει τις σημάνσεις `<? PHP και ?>`. Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Ένα απλό παράδειγμα που δείχνει την εισαγωγή της PHP μέσα στην HTML είναι το παρακάτω:

```
<html>

<head>

<title> Παράδειγμα </title>

</head>

<body>

<?PHP echo "Καλημέρα!"; ?>

</body>
```

</html>

Γράφουμε ένα HTML script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο. Ο κώδικας της PHP περικλείεται με ειδικές σημάνσεις αρχής και τέλους. Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον server. Αυτό σημαίνει με απλά λόγια ότι οι τελικοί χρήστες δεν μπορούν ποτέ να δουν τον πηγαίο κώδικα (source code) της PHP.

Τα scripts της PHP αποθηκεύονται με την κατάληξη .PHP και κάθε φορά που ο Web server πρέπει να στείλει ένα αρχείο που τελειώνει σε .PHP, πρώτα το στέλνει στον διερμηνευτή (interpreter) της PHP, ο οποίος εκτελεί τον κώδικα της PHP που υπάρχει στο script πριν επιστρέψει το παραγόμενο αρχείο στον τελικό χρήστη.

5.4.3. PHPMYAdmin

Η PHPMYAdmin είναι ένα εργαλείο με γραφικό περιβάλλον που αποτελείται από ένα σύνολο PHP scripts με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Η PHPMYAdmin μπορεί να διαχειριστεί ένα ολόκληρο mysql server ή ακόμα και απλές βάσεις δεδομένων όπου ο κάθε χρήστης έχει ένα λογαριασμό και μπορεί να δημιουργήσει και να διαχειριστεί τις δικές του βάσεις δεδομένων. Ότι δηλαδή μπορεί να γίνει από την γραμμή εντολών της MySQL, μπορεί να γίνει πολύ πιο εύκολα από την PHPMYAdmin.

Συνοπτικά, με την PHPMYAdmin μπορούμε:

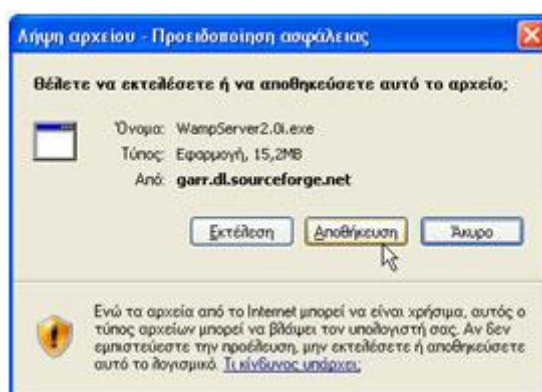
- να δημιουργήσουμε και να διαγράψουμε βάσεις δεδομένων
- να δημιουργήσουμε, να διαγράψουμε και να χειριστούμε πίνακες
- να διαγράψουμε, να προσθέσουμε και να επεξεργαστούμε πεδία πινάκων
- να εκτελέσουμε ερωτήματα και να εξάγουμε τα αποτελέσματα σε οποιαδήποτε μορφή (text, sql, κτλ)
- να εξαγάσουμε (back-up) πεδία, πίνακες αλλά και ολόκληρες βάσεις δεδομένων

5.5. Εγκατάσταση Apache, MySQL, PHP & PHPMyAdmin

Η εγκατάσταση των εργαλείων PHP, MySQL, Apache και PHPMyAdmin γίνεται εύκολα κατεβάζοντας ένα και μόνο αρχείο από την ιστοσελίδα: <http://www.wampserver.com/en/download.php>

Κάνουμε κλικ στο σύνδεσμο **DOWNLOAD WampServer 2**. Η εγκατάσταση του **Wamp** είναι μια απλή διαδικασία και διαρκεί μερικά λεπτά, παρακάτω ακολουθούν μερικά στιγμιότυπα της διαδικασίας εγκατάστασης.

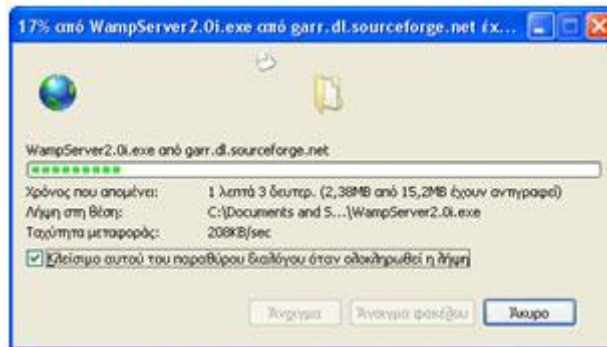
Εμφανίζεται το παράθυρο διαλόγου “**Λήψη αρχείου – Προειδοποίηση ασφαλείας**”. Κάνουμε κλικ στο κουμπί **Αποθήκευση**.



Εικόνα 5-3 : Αποθήκευση αρχείου

Εμφανίζεται το παράθυρο διαλόγου “**Αποθήκευση ως**”. Στο παράθυρο αυτό μπορούμε να επιλέξουμε τη θέση αποθήκευσης. Συγκεκριμένα, θα αποθηκεύσουμε το αρχείο στην **Επιφάνεια εργασίας**. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Αποθήκευση**.

Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Το παράθυρο αυτό μας δείχνει την πρόοδο “λήψης” του αρχείου.



Εικόνα 5-4 : Λήψη αρχείου

Μετά το τέλος της λήξης του αρχείου κάνουμε κλικ στο κουμπί **Κλείσιμο** (εάν στη διαδικασία της λήξης του αρχείου δεν έχουμε επιλέξει το πλαίσιο ελέγχου “**Κλείσιμο αυτού του παραθύρου διαλόγου όταν ολοκληρωθεί η λήψη**”). Στην επιφάνεια εργασίας εμφανίζεται το παρακάτω εικονίδιο.



Εικόνα 5-5 : Εικονίδιο εγκατάστασης

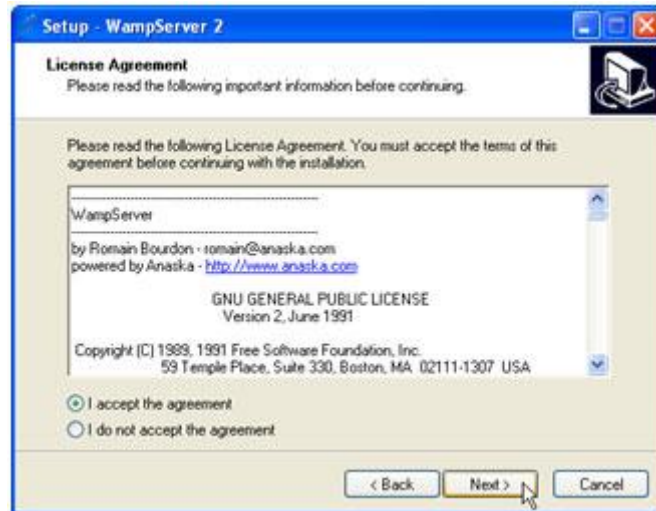
Κάνουμε διπλό κλικ πάνω στο εικονίδιο για να ξεκινήσει η εγκατάσταση του προγράμματος.

Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Κάνουμε κλικ στο κουμπί **Next**.



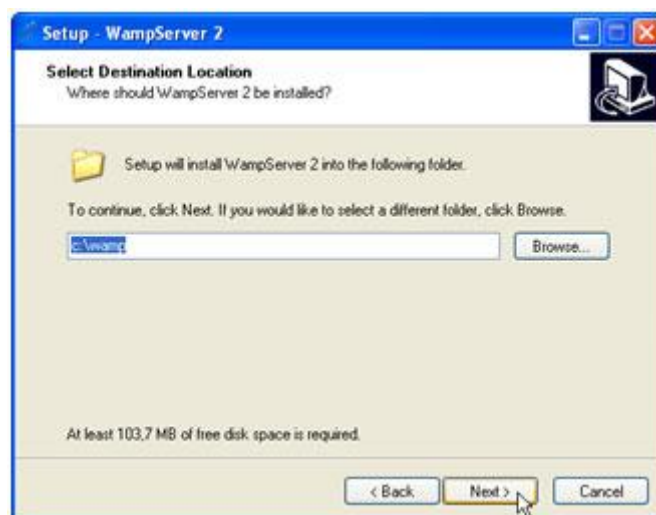
Εικόνα 5-6 : Εγκατάσταση

Εμφανίζεται το παράθυρο διαλόγου “Setup – WampServer 2”. Κάνουμε κλικ στο κουμπί επιλογής **I accept the agreement**. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Next**.



Εικόνα 5-7 : Εγκατάσταση 2

Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Σε αυτό το σημείο δημιουργείτε ένας φάκελος με την ονομασία wamp μέσα στον οποίο θα γίνει η εγκατάσταση του WAMP και θα περιέχει όλα τα υποπρογράμματα που θα συμπεριλάβουμε στην εφαρμογή μας (addons, templates κτλ). Κάνουμε κλικ στο κουμπί **Next**.



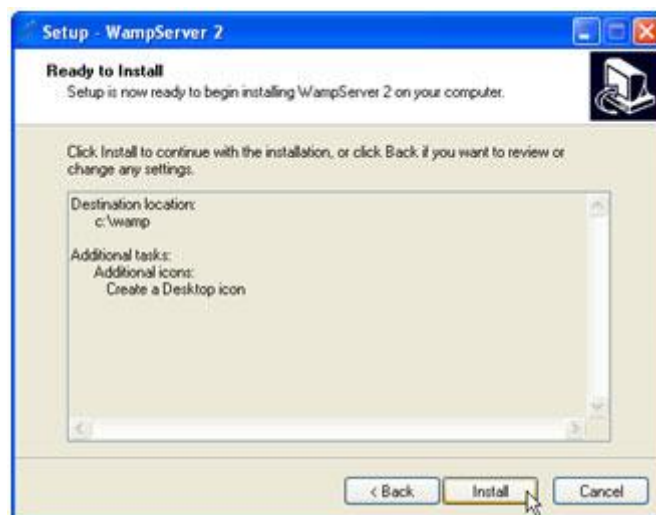
Εικόνα 5-8 : Εγκατάσταση 3

Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Επιλέγουμε το πλαίσιο ελέγχου **Create a Desktop icon**, για να δημιουργηθεί το αντίστοιχο εικονίδιο στην επιφάνεια εργασίας. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Next**.



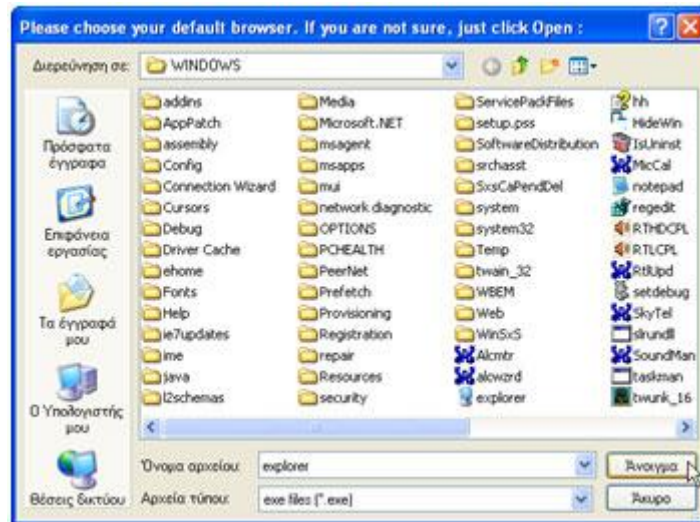
Εικόνα 5-9 : Εγκατάσταση 4

Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Install**.



Εικόνα 5-10 : Εγκατάσταση 5

Μετά το τέλος της εγκατάστασης, εμφανίζεται το παρακάτω παράθυρο διαλόγου. Στο πλαίσιο κειμένου **Όνομα αρχείου** πληκτρολογούμε **explorer**. Έτσι, δηλώνουμε τον **Internet Explorer** ως τον προεπιλεγμένο **Web Browser**. Στην συνέχεια, κάνουμε κλικ στο κουμπί **Άνοιγμα**.



Εικόνα 5-11 : Εγκατάσταση 6


Στο παράθυρο που εμφανίζεται κάνουμε κλικ στο κουμπί **Next**.

Εάν εμφανιστεί το παράθυρο που μας προειδοποιεί για την ασφάλεια των Windows κάνουμε κλικ στο κουμπί **Άρση αποκλεισμού**.

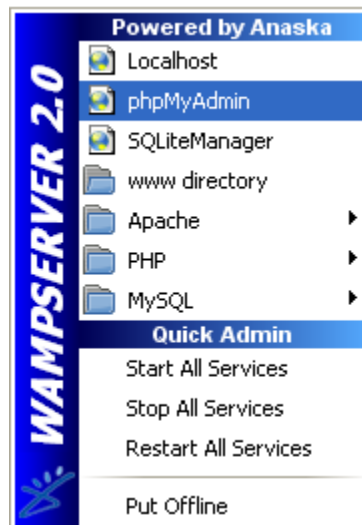
Εμφανίζεται το παρακάτω παράθυρο διαλόγου. Αφήνουμε επιλεγμένο το πλαίσιο ελέγχου **Launch Wampserver 2 now**, για να ξεκινήσει άμεσα η λειτουργία του **Server**. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Finish**.



Εικόνα 5-12 : Τέλος εγκατάστασης

Στην κάτω δεξιά γωνία της γραμμής εργασιών της επιφάνειας εργασίας, εμφανίζεται το εικονίδιο του **Wamp** . Κάνοντας αριστερό κλικ στο παραπάνω εικονίδιο μας

εμφανίζεται το μενού της παρακάτω εικόνας στο οποίο βρίσκονται οι διάφορες λειτουργίες που μπορούμε να κάνουμε.



Εικόνα 5-13 : Λειτουργίες WampServer

5.6. Επίλογος

Σε αυτό το σημείο ολοκληρώθηκε η παρουσίαση των τεχνολογιών και των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Στο επόμενο κεφάλαιο ακολουθεί η ανάλυση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 6

Υλοποίηση

6.1. Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται η υλοποίηση της εφαρμογής με τον απαραίτητο κώδικα και σχολιασμό του.

6.2. Κανονικοποίηση

Τα προβλήματα που είναι πιθανό να παρουσιασθούν κατά τη διαδικασία της υλοποίησης του σχεδιασμού μιας βάσης δεδομένων είναι η περιττή επανάληψη πληροφοριών, καθώς και δυσκολίες στην ενημέρωση της βάσης δεδομένων. Για να αντιμετωπιστούν με επιτυχία, θα πρέπει να διασπάσουμε τις μεγάλες σχέσεις σε μικρότερες. Αυτό γίνεται με τη διαδικασία της κανονικοποίησης, έτσι ώστε η βάση δεδομένων να είναι έτοιμη για καταχώριση στοιχείων.

Στο σχεσιακό περιβάλλον βάσης δεδομένων, οι πίνακες στις βάσεις δεδομένων πρέπει να είναι κανονικοποιημένοι τουλάχιστον σε πρώτο βαθμό.

Η κανονικοποίηση (normalization) είναι μια τεχνική που ασχολείται με την ανάλυση των σχέσεων (συσχετίσεων) σε μια βάση δεδομένων, όπου κάνουμε μετατροπή των αρχικών μεγάλων σχέσεων σε μικρότερες. Κατά τη διαδικασία της κανονικοποίησης διαιρούνται οι πίνακες μιας βάσης δεδομένων και γίνεται ένωση- συσχέτιση των πινάκων, που θα προκύψουν βάση κοινών τιμών.

6.2.1. Μη κανονική μορφή (MKM)

Στην μη κανονική μορφή δεν έχουμε ακόμη πίνακα με την σχεσιακή έννοια. Τα πεδία τοποθετούνται το ένα δίπλα στο άλλο με μόνο επιπλέον στοιχείο πληροφορίας την ομαδοποίηση πεδίων που αντιστοιχούν σε χαρακτηριστικά που παίρνουν πολλαπλές τιμές. Η ομαδοποίηση των τελευταίων αναπαριστάται με ζεύγη εσωτερικών παρενθέσεων στην περιγραφή της δομής του αντίστοιχου πίνακα.

MKM : (id, name (id, name) (id, name) (id, code, name, related_schools, description, public_op, private_op, self_op, post_gr, post_en, job_rights, carriers, study_programms, payment, cooperation_abroad, scholarships, insurance) (id, tei_depts_id, name, program, job_rights, payment, post_gr, cooperation_abroad, post_en, scholarships, carriers, public_op, private_op, self_op, insurance))

6.2.2. 1^η κανονική μορφή (1NF)

Για να έρθει ένας πίνακας σε πρώτη κανονική μορφή, αφαιρούνται οι ομάδες δεδομένων που επαναλαμβάνονται. Δηλαδή δεν επιτρέπεται να υπάρχουν πεδία στον πίνακα που να παίρνουν πολλαπλές τιμές.

1NF : (id, name (id, name) (id, name) (id, code, name, related_schools, description, public_op, private_op, self_op, post_gr, post_en, job_rights, carriers, study_programms, payment, cooperation_abroad, scholarships, insurance) (id, tei_depts_id, name, program, job_rights, payment, post_gr, cooperation_abroad, post_en, scholarships, carriers, public_op, private_op, self_op, insurance))

6.2.3. 2^η κανονική μορφή (2NF)

Η δεύτερη κανονική μορφή προκύπτει από την πρώτη κανονική μορφή εάν μετασχηματιστεί η δομή του πίνακα με τέτοιο τρόπο ώστε να απομακρυνθούν όλες οι μερικές συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία του. Ένας πίνακας βρίσκεται σε δεύτερη κανονική μορφή όταν όλα τα πεδία που δεν ανήκουν στο πρωτεύον κλειδί του πίνακα εξαρτώνται μόνο από τα πεδία του πρωτεύοντος κλειδιού και μάλιστα

2NF : (id, name (id, name) (id, name) (id, code, name, related_schools, description, public_op, private_op, self_op, post_gr, post_en, job_rights, carriers, study_programms, payment, cooperation_abroad, scholarships, insurance) (id, tei_depts_id, name, program, job_rights, payment, post_gr, cooperation_abroad, post_en, scholarships, carriers, public_op, private_op, self_op, insurance))

6.2.4. 3^η κανονική μορφή (3NF)

Για να μετασχηματιστεί ένας πίνακας από 2NF σε 3NF, ανιχνεύεται η πιθανή ύπαρξη έμμεσων συναρτησιακών εξαρτήσεων. Αν υπάρχουν, όπως και παραπάνω

μετακινούνται σε άλλο πίνακα. Οι πίνακες στην τρίτη κανονική μορφή της εφαρμογής παραμένουν ίδιοι όπως και στη δεύτερη κανονική μορφή.

6.3. Οντότητες και συσχετίσεις

Το μοντέλο χρησιμοποιείται για να παραστήσει τις έννοιες ή τη δομή μιας βάσης δεδομένων είναι το **Μοντέλο Οντοτήτων–Συσχετίσεων** (Entity Relationship model, ER model). Οι βασικές έννοιες του μοντέλου είναι:

- Οντότητες
- Ιδιότητες ή Χαρακτηριστικά
- Συσχετίσεις

Το μοντέλο οντοτήτων–συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα δεδομένα (εννοιολογικό), χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.

Οι Οντότητες

Με τον όρο οντότητα (entity) αναφερόμαστε σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση που έχει μια ανεξάρτητη ύπαρξη. Είναι κάτι που ξεχωρίζει και μπορούμε να συγκεντρώσουμε πληροφορίες (στοιχεία) γι' αυτό. Η οντότητα είναι αντίστοιχη με την έννοια της εγγραφής που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια του αντικειμένου στις σύγχρονες αντικειμενοστραφείς γλώσσες προγραμματισμού.

Οι Ιδιότητες (Χαρακτηριστικά) των Οντοτήτων


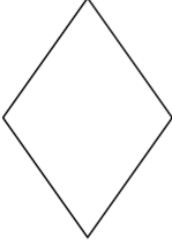


Με τον όρο ιδιότητες (properties) ή χαρακτηριστικά (attributes) αναφερόμαστε στα συστατικά (δομικά) στοιχεία που προσδιορίζουν (αποτελούν) μια οντότητα. Η ιδιότητα είναι αντίστοιχη με την έννοια του πεδίου που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια της μεταβλητής στις γλώσσες προγραμματισμού.

Απ' όλες τις ιδιότητες μιας οντότητας, υπάρχει μία μόνο ιδιότητα, και σπανιότερα ένας συνδυασμός δύο ή και περισσότερων ιδιοτήτων, η τιμή της οποίας είναι μοναδική και προσδιορίζει την κάθε εμφάνιση (στιγμιότυπο) της οντότητας και αποκαλείται **πρωτεύον κλειδί** (primary key).

Οι Συσχετίσεις Μεταξύ Οντοτήτων

Ο σωστός σχεδιασμός και προσδιορισμός των οντοτήτων και των ιδιοτήτων τους αποτελούν τα θεμελιώδη βήματα για τη σωστή σχεδίαση και υλοποίηση μιας βάσης δεδομένων.

Οι βασικές έννοιες του μοντέλου είναι Μοντέλο Οντοτήτων–Συσχετίσεων παριστάνονται ως εξής:

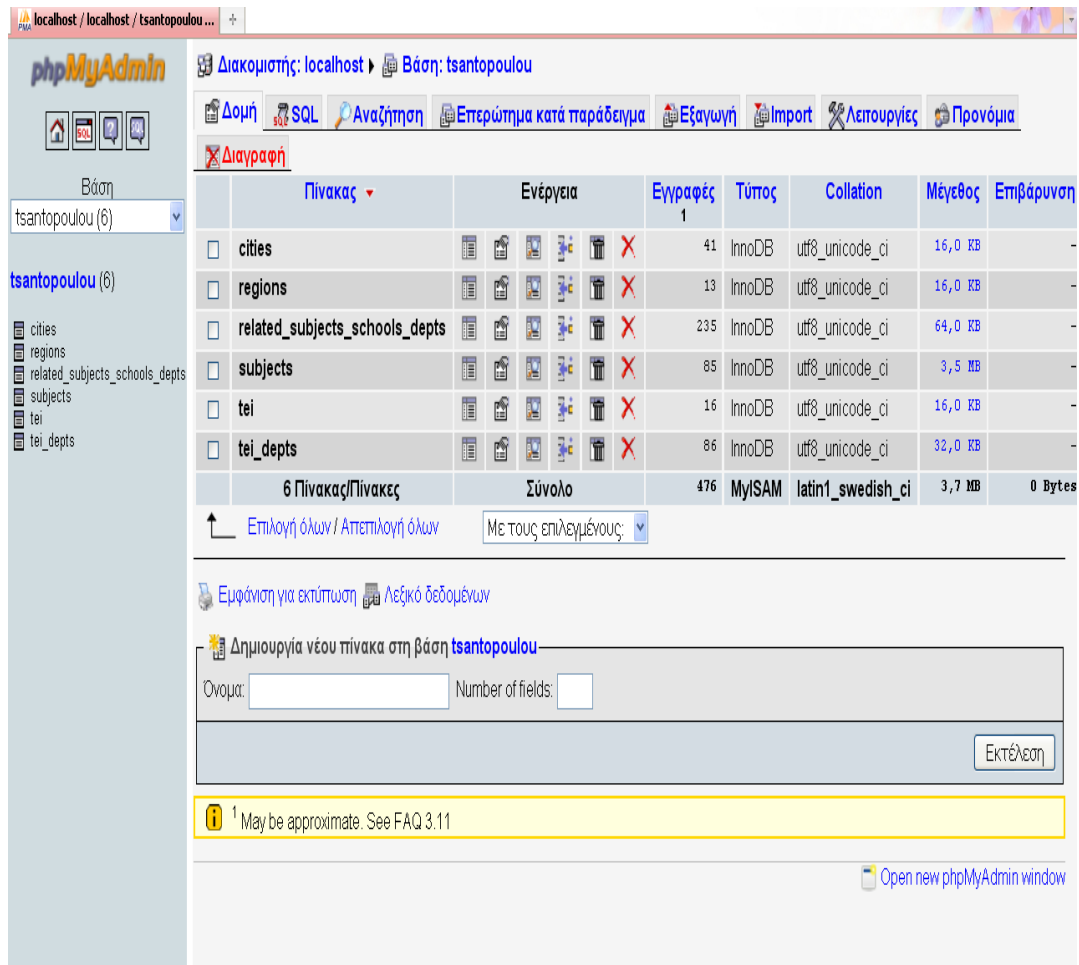
	ΟΝΤΟΤΗΤΑ (ENTITY)
	ΣΥΣΧΕΤΙΣΗ (RELATIONSHIP)
	ΠΕΔΙΟ (ATTRIBUTE)
	ΠΕΔΙΟ ΚΛΕΙΔΙ (KEY ATTRIBUTE)

Εικόνα 6-1 : Σύμβολα σχεδίασης

6.4. Τεκμηρίωση Πινάκων

Στη εφαρμογή χρησιμοποιούνται 6 πίνακες, οι οποίοι είναι: cities, regions, tei, subjects, tei_depts και related_sunjects_schools_depts.

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη



Εικόνα 6-2 : Οι πίνακες της εφαρμογής

Πίνακας cities

cities (id, regions_id, name)

Πίνακας 7-1 : Τα πεδία του πίνακα cities

id	int(11)
regions_id	int(11)
name	varchar(40)

Πίνακας regions

regions (id, name)

Πίνακας 6-2 : Τα πεδία του πίνακα regions

id	int(11)
name	varchar(40)

Πίνακας tei

tei (id, name)

Πίνακας 6-3 : Τα πεδία του πίνακα tei

id	int(11)
name	varchar(100)

Πίνακας subjects

subjects (id, code, name, related_schools, description, public_op, private_op, self_op, post_gr, post_en, job_rights, carriers, study_programms, payment, cooperation_abroad, scholarships, insurance)

Πίνακας 6-4 : Τα πεδία του πίνακα subjects

id	int(11)
code	int(11)
name	varchar(100)
related_schools	text

description	text
public_op	text
private_op	text
self_op	text
post_gr	text
post_en	text
job_rights	text
carriers	text
study_programms	text
payment	text
cooperation_abroad	text
scholarships	text
insurance	text

Πίνακας tei_depts

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

tei_depts (id, tei_id, tei_depts_id, name, regions_id, cities_id, program, job_rights, payment, post_gr, cooperation_abroad, post_en, scholarships, carriers, public_op, private_op, self_op, insurance)

Πίνακας 6-5 : Τα πεδία του πίνακα tei_depts

id	int(11)
tei_id	int(11)
tei_depts_id	int(11)
name	varchar(100)
regions_id	int(11)
cities_id	enum('y','n')
program	enum('y','n')
job_rights	enum('y','n')
payments	enum('y','n')
post_gr	enum('y','n')
cooperation_abroad	enum('y','n')
post_en	enum('y','n')
scholarships	enum('y','n')
carriers	enum('y','n')
public_op	enum('y','n')

private_op	enum('y','n')
self_op	enum('y','n')
insurance	enum('y','n')

Πίνακας related_subjects_schools_depts

related_subjects_schools_depts (id, subjects_id, tei_id, tei_depts_id)

Πίνακας 6-6 : Τα πεδία του πίνακα related_subjects_schools_depts

id	int(11)
subjects_id	int(11)
tei_id	int(11)
tei_depts_id	int(11)

MySQL κώδικας

Πίνακας cities

```
CREATE TABLE IF NOT EXISTS `cities` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `regions_id` int(11) NOT NULL,  
  `name` varchar(40) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=42 ;
```

Πίνακας regions

```
CREATE TABLE IF NOT EXISTS `regions` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
`name` varchar(40) COLLATE utf8_unicode_ci NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=14 ;
```

Πίνακας tei

```
CREATE TABLE IF NOT EXISTS `tei` (  
  
`id` int(11) NOT NULL AUTO_INCREMENT,  
  
`name` varchar(100) COLLATE utf8_unicode_ci NOT NULL,  
  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=17 ;
```

Πίνακας subjects

```
CREATE TABLE IF NOT EXISTS `subjects` (  
`id` int(11) NOT NULL AUTO_INCREMENT,  
`code` int(11) DEFAULT NULL,  
`name` varchar(100) COLLATE utf8_unicode_ci NOT NULL,  
`related_schools` text COLLATE utf8_unicode_ci,  
`description` text COLLATE utf8_unicode_ci,  
`public_op` text COLLATE utf8_unicode_ci,  
`private_op` text COLLATE utf8_unicode_ci,  
`self_op` text COLLATE utf8_unicode_ci,  
`post_gr` text COLLATE utf8_unicode_ci,  
`post_en` text COLLATE utf8_unicode_ci,  
`job_rights` text COLLATE utf8_unicode_ci,  
`carriers` text COLLATE utf8_unicode_ci,  
`study_programms` text COLLATE utf8_unicode_ci,  
`payment` text COLLATE utf8_unicode_ci,  
`cooperation_abroad` text COLLATE utf8_unicode_ci,  
`scholarships` text COLLATE utf8_unicode_ci,
```

```
`insurance` text COLLATE utf8_unicode_ci,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=87 ;
```

Πίνακας tei_depts

```
CREATE TABLE IF NOT EXISTS `tei_depts` (  
  
`id` int(11) NOT NULL AUTO_INCREMENT,  
`tei_id` int(11) NOT NULL,  
`tei_depts_id` int(11) NOT NULL,  
`name` varchar(100) COLLATE utf8_unicode_ci NOT NULL,  
`regions_id` int(11) DEFAULT NULL,  
`cities_id` int(11) DEFAULT NULL,  
`program` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`job_rights` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`payment` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`post_gr` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`cooperation_abroad` enum('y','n') COLLATE utf8_unicode_ci NOT NULL  
DEFAULT 'y',  
`post_en` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`scholarships` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`carriers` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`public_op` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`private_op` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`self_op` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
`insurance` enum('y','n') COLLATE utf8_unicode_ci NOT NULL DEFAULT 'y',  
PRIMARY KEY (`id`),  
KEY `id` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=89 ;
```

Πίνακας related_sunjects_schools_depts

```
CREATE TABLE IF NOT EXISTS `related_subjects_schools_depts` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `subjects_id` int(11) NOT NULL,  
  `tei_id` int(11) DEFAULT NULL,  
  `tei_depts_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `subjects_id` (`subjects_id`,`tei_id`,`tei_depts_id`),  
  KEY `tei_id` (`tei_id`),  
  KEY `tei_depts_id` (`tei_depts_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=237 ;
```

6.5. Σύνδεση στη Βάση Δεδομένων

Πριν μπορέσουμε να πάρουμε το περιεχόμενο μιας βάσης δεδομένων της MySQL για να το εισάγουμε σε μια ιστοσελίδα, πρέπει πρώτα να δούμε πώς θα δημιουργήσουμε σύνδεση (connection) με την MySQL. Αυτό γίνεται με κλήση της συνάρτησης `mysql_connect()` η οποία δέχεται τρία strings: το address, ένα username και ένα password. Παράδειγμα:

```
mysql_connect(<address>, <username>, <password>);
```

Όπου το `<address>` είναι η IP διεύθυνση ή το hostname του υπολογιστή στον οποίο εκτελείται το λογισμικό του MySQL server, αν εκτελείται στον ίδιο υπολογιστή με το λογισμικό του Web server τότε είναι localhost, και τα `<username>` και `<password>` είναι τα ίδια με τα user name και password της MySQL που χρησιμοποιήσαμε για να συνδεθούμε στον MySQL server.

Το παραπάνω κομμάτι κώδικα χρησιμοποιεί τη `mysql_connect()` για να γίνει σύνδεση με τον διακομιστή της βάσης δεδομένων MySQL :

```
$connection = mysql_connect(SERVER, USER, MY_PASSWORD);
```

```
if (!$connection) {  
    die("Database connection failed ");  
}
```

6.6. Επιλογή της Βάσης Δεδομένων

Αφού έχουμε συνδεθεί με το διακομιστή της βάσης δεδομένων πρέπει να επιλέξουμε τη βάση δεδομένων με την οποία θέλουμε να δουλέψουμε. Αυτή τη δυνατότητα μας

τη δίνει η `mysql_select_db()`. Η συγκεκριμένη μέθοδος παίρνει ως ορίσματα το όνομα μιας βάσης δεδομένων και προαιρετικά έναν αναγνωριστή συνδέσμου. Η `mysql_select_db()` επιστρέφει `true` αν η βάση δεδομένων υπάρχει και μπορούμε να την προσπελάσουμε.

```
$db_select = mysqli_select_db(MY_DATABASE, db_connection());  
  
if (!$db_select) {  
  
    die("Database selection failed ");
```

6.7. Διαχείριση λαθών

Στις παραπάνω περιπτώσεις εάν οι επιστρεφόμενες τιμές από τις συναρτήσεις που καλούσαμε παρουσίαζαν κάποιο πρόβλημα καλούσαμε την `die()` για τερματισμό της εκτέλεσης του script. Αν θέλουμε όμως να τυπώσουμε περισσότερες πληροφορίες στον browser σχετικές με το λάθος τότε μπορούμε να χρησιμοποιήσουμε τις `mysql_error()`.

Παρακάτω ακολουθεί ένα σχετικό παράδειγμα:

```
$db_select = mysqli_select_db(MY_DATABASE, db_connection());  
  
if (!$db_select) {  
  
    die("Database selection failed " .mysql_error());
```

6.8. Εκτέλεση SQL ερωτημάτων

Αφού συνδεθούμε με την βάση δεδομένων μπορούμε να γράψουμε ερωτήματα (queries) και να δούμε τα αποτελέσματα. Αυτό γίνεται με την συνάρτηση της PHP `mysql_query()`.

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

Όπου το `<query>` είναι ένα string με τις εντολές της SQL που θα εκτελεστούν και η παράμετρος `connection identifier` (αναγνωριστικό 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.

6.9. Κώδικας εφαρμογής

6.9.1. *index.php*

```
<h1 style="text-align:center;text-shadow: 0 0 0.2em #F87" > Ψηφιακός χάρτης  
ειδικοτήτων ανώτατης τεχνολογικής εκπαίδευσης </h1>
```

```
<center style="position:relative; top:10%; left:25%; right:20%; border:5px  
outset gray; width: 650px;" class="style1">
```

```
<object data="~map.swf" height="650" type="application/x-  
shockwave-flash" width="650">
```

```
<param name="movie" value="map.swf" />
```

```
</object>
```

```
</center>
```

```
<a href="index2.php" style="position:relative; left:90%">  </a>
```

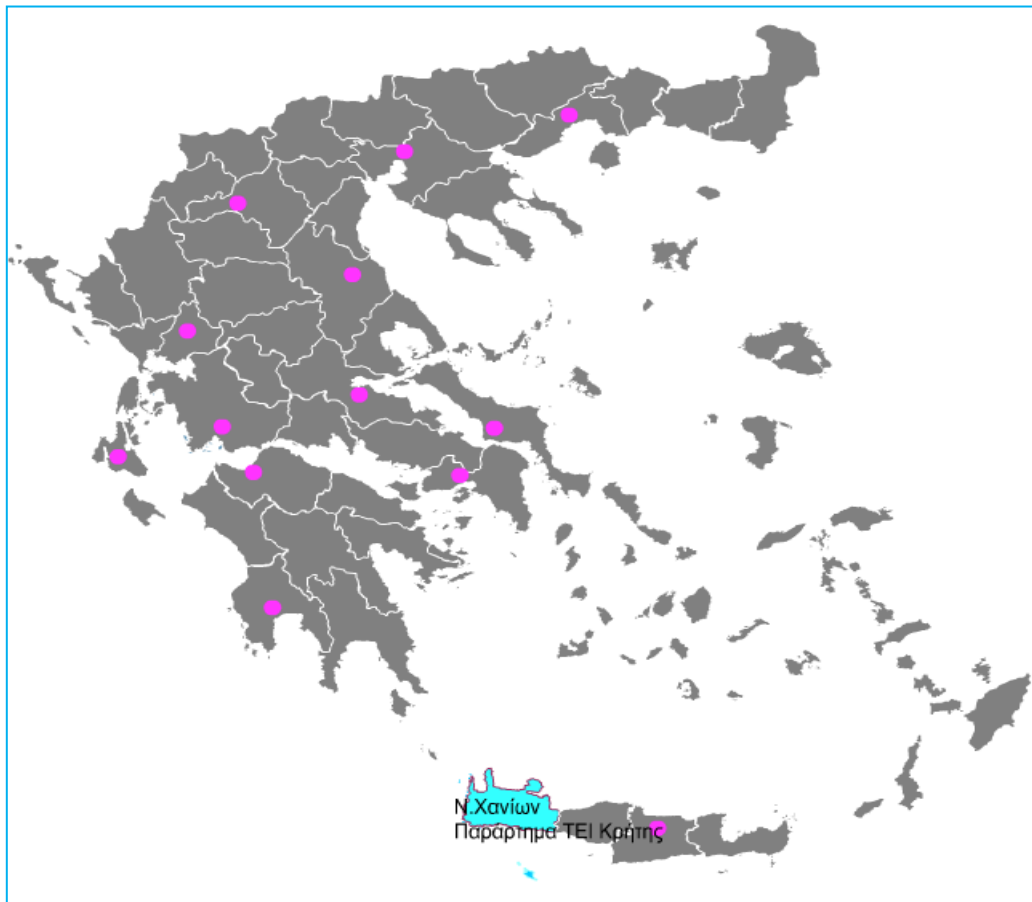
```
<fieldset style="position:absolute; top:30%;left:4%;right:76%; border-  
color:fuchsia;text-align:justify;font-family:'Times New Roman', Times, serif;font-  
size:17px">
```

Για να δεις την αρχική σελίδα από κάποιο Τεχνολογικό Εκπαιδευτικό Ίδρυμα κάνε click πάνω στον νομό που βρίσκεται.

```
</fieldset>
```

Με τον παραπάνω κώδικα δημιουργείται η αρχική σελίδα της εφαρμογής. Όπως φαίνεται και στην Εικόνα 6-2 η αρχική σελίδα περιέχει έναν ψηφιακό χάρτη ειδικοτήτων της ανώτατης τεχνολογικής εκπαίδευσης . Ο χάρτης δημιουργήθηκε με το πρόγραμμα Adobe Flash CS4 Professional και κάνοντας περιήγηση ο χρήστης πάνω του μπορεί να δει όλα τα TEI της χώρας. Όλοι οι νομοί αλλάζουν χρώμα όταν ο χρήστης περνάει από πάνω τους με το ποντίκι για πιο εύκολη περιήγηση και όμορφη παρουσίαση. Οι έδρες των TEI βρίσκονται στους νομούς όπου υπάρχει η ροζ βούλα και τα παραρτήματα τους παρουσιάζονται όταν ο χρήστης περάσει με το ποντίκι πάνω από τον νομό. Επίσης, κάνοντας κλικ στο νομό ανοίγει η αρχική σελίδα του

αντίστοιχου Τεχνολογικού Ιδρύματος κάθε φορά. Εάν σε κάποιο νομό δεν υπάρχει κάποιο ΤΕΙ ή το παράρτημα του τότε εμφανίζεται μόνο το όνομα του.



Εικόνα 6-3 : Ψηφιακός χάρτης

Ο κώδικας του νομού Θεσσαλονίκης παρουσιάζεται παρακάτω:

```
on (release){  
    getURL("http://www.teithe.gr/");  
}  
  
on (release, rollOver){  
    gotoAndStop(28);  
}  
  
on (release, rollOut){  
    gotoAndStop(18);
```

}

6.9.2. *index2.php*

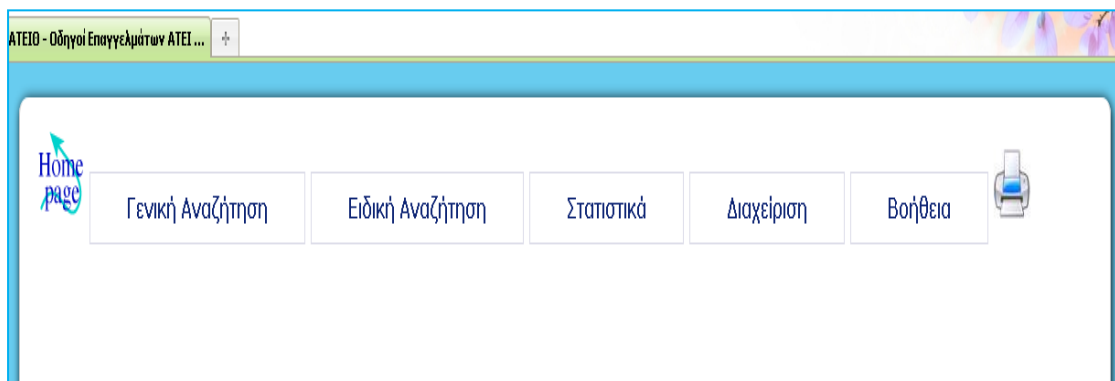
Με τον παρακάτω κώδικα γίνεται σύνδεση στην βάση δεδομένων, εισάγεται το css και καλούνται οι συναρτήσεις `do_header()`, `get_mod()`.

```
<?php

    $connection = db_connection();
    $sc = html_title("ATEIΘ - Οδηγοί Επαγγελματιών ΑΤΕΙ Ελλάδας",
    "css/index.css", "");
    $sc .= '<body>
    <div class="shadow">
        <div id="wrapper">';
        $sc .= do_header();
        $sc .= get_mod($connection);
        $sc .= '</body></html>';
        echo $sc;

?>
```

Η συνάρτηση `do_header()` εμφανίζει το μενού με το οποίο ο χρήστης μπορεί να περιηγηθεί στην εφαρμογή και η `get_mod()` αντιλαμβάνεται τι επέλεξε ο χρήστης από το μενού και εμφανίζει την αντίστοιχη καρτέλα.



Εικόνα 6-4 : Μενού πλοήγησης της εφαρμογής

Η συνάρτηση `get_mod()` αντιλαμβάνεται τι επέλεξε ο χρήστης από το μενού και εμφανίζει την αντίστοιχη καρτέλα. παρακάτω ακολουθεί απόσπασμα κώδικα όπου γίνεται έλεγχος αν ο χρήστης επέλεξε την γενική ή ειδική αναζήτηση.

```
switch ($mod) {  
    ...  
    case "search" :  
        $sc .= do_search($connection);  
        if (isset($_POST['searchButton'])) {  
            $sc .= search($connection);  
        }  
        break;  
    case "advanced" :  
        $sc .= do_advanced($connection);  
        if (isset($_POST['advancedButton'])) {  
            $sc .= advanced($connection);  
        }  
    ...  
}
```

6.9.3. *index.css*

Προκειμένου σε όλη την εφαρμογή να υπάρχει ένα προκαθορισμένο στυλ χρησιμοποιείται CSS. Ο κώδικας με τον οποίο εισάγεται το CSS στην εφαρμογή είναι ο ακόλουθος:

```
<link href=".$stylesheet." rel="stylesheet" type="text/css" />;
```

ΚΕΦΑΛΑΙΟ 7

Περιγραφή και χρήση της εφαρμογής

7.1. Εισαγωγή

Παρακάτω δίνονται οδηγίες σχετικά με την περιήγηση του χρήστη στην εφαρμογή και παρουσιάζονται κάποια χαρακτηριστικά στιγμιότυπα.

7.2. Αρχική σελίδα

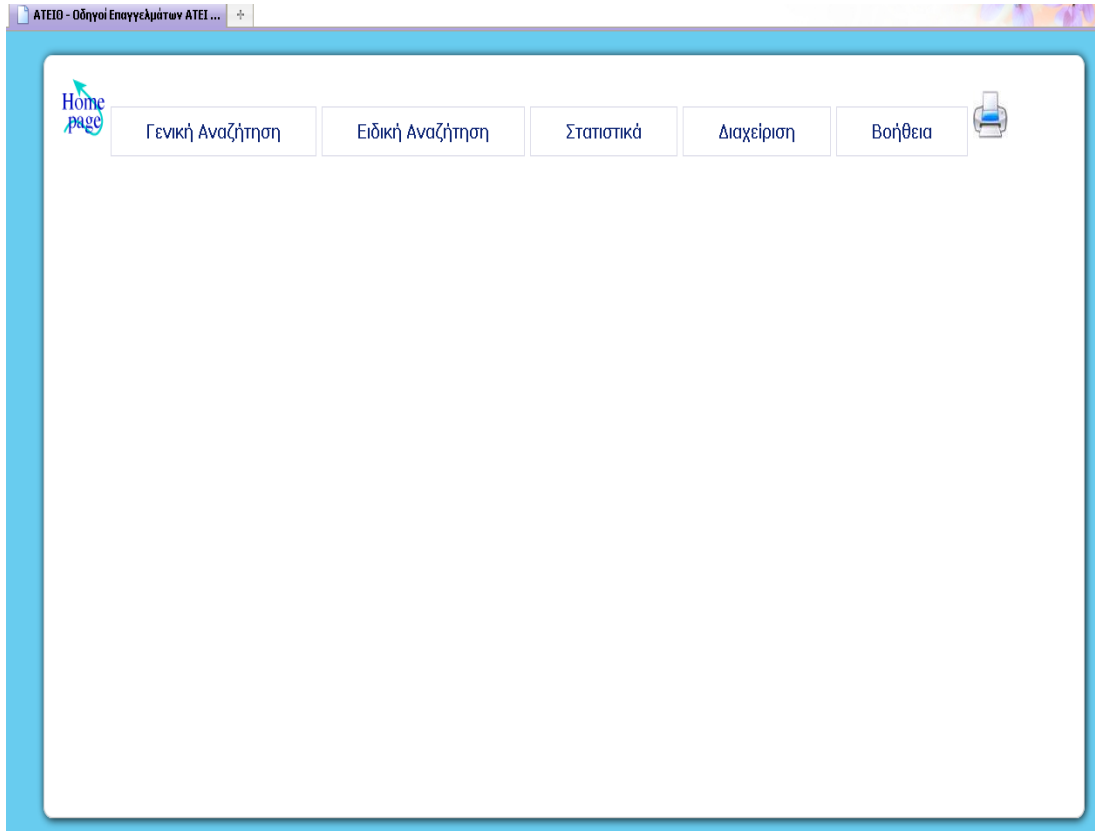


Εικόνα 7-1 : Η πρώτη σελίδα της εφαρμογής

Αυτή είναι η πρώτη σελίδα της εφαρμογής. Ο χρήστης μπορεί να περιηγηθεί στον χάρτη και να ενημερωθεί για τα ΤΕΙ όλης της χώρας. Πατώντας στο βέλος κάτω δεξιά φορτώνεται η δεύτερη σελίδα που φαίνεται παρακάτω.

7.2. Μενού περιήγησης

Ο χρήστης μπορεί να περιηγηθεί στην εφαρμογή μέσω του παρακάτω μενού.



Εικόνα 7-2 : Μενού της εφαρμογής

7.3. Γενική Αναζήτηση

Επιλέγοντας την Γενική Αναζήτηση εμφανίζεται η σελίδα που παρουσιάζεται στην Εικόνα 7-3. Η Γενική Αναζήτηση δίνει την δυνατότητα στον χρήστη να αναζητήσει κάποιο τμήμα με βάση κάποια συγκεκριμένα κριτήρια. Αφού επιλέξει τα κριτήρια τα οποία θέλει να πληρούν ή όχι όταν πατήσει το κουμπί Αναζήτηση εμφανίζονται τα αντίστοιχα αποτελέσματα. Στην Εικόνα 7-4 παρουσιάζονται τα αποτελέσματα της Γενικής Αναζήτησης με κριτήριο να βρίσκεται στην Λαμία, να έχουν Πρόγραμμα Σπουδών και δυνατότητα Μεταπτυχιακών Σπουδών στην Ελλάδα.

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

ATEIO - Οδηγοί Επαγγελεμάτων ΑΤΕΙ ...

Home Page

Γενική Αναζήτηση | Ειδική Αναζήτηση | Στατιστικά | Διαχείριση | Βοήθεια

Κριτήρια Αναζήτησης στη Βάση Δεδομένων

TEI:

Περιφέρεια:

Πρόγραμμα Σπουδών:

Επαγγελματικά Δικαιώματα: (Dropdown menu open with options: Επιλογή, ΝΑΙ, ΟΧΙ)

Μεταπτυχιακές Σπουδές στην Ελλάδα:

Μεταπτυχιακές Σπουδές στο Εξωτερικό:

Επαγγελματικοί - Επιστημονικοί Φορείς:

Εργασία στο Δημόσιο:

Εργασία στον Ιδιωτικό Τομέα:

Αυτοαπασχόληση:

Υποτροφίες:

Εικόνα 7-3 : Φόρμα Γενικής αναζήτησης

ATEIO - Οδηγοί Επαγγελεμάτων ΑΤΕΙ ...

Home Page

Γενική Αναζήτηση | Ειδική Αναζήτηση | Στατιστικά | Διαχείριση | Βοήθεια

Κριτήρια Αναζήτησης στη Βάση Δεδομένων

TEI:

Περιφέρεια:

Πρόγραμμα Σπουδών:

Επαγγελματικά Δικαιώματα:

Μεταπτυχιακές Σπουδές στην Ελλάδα:

Μεταπτυχιακές Σπουδές στο Εξωτερικό:

Επαγγελματικοί - Επιστημονικοί Φορείς:

Εργασία στο Δημόσιο:

Εργασία στον Ιδιωτικό Τομέα:

Αυτοαπασχόληση:

Υποτροφίες:

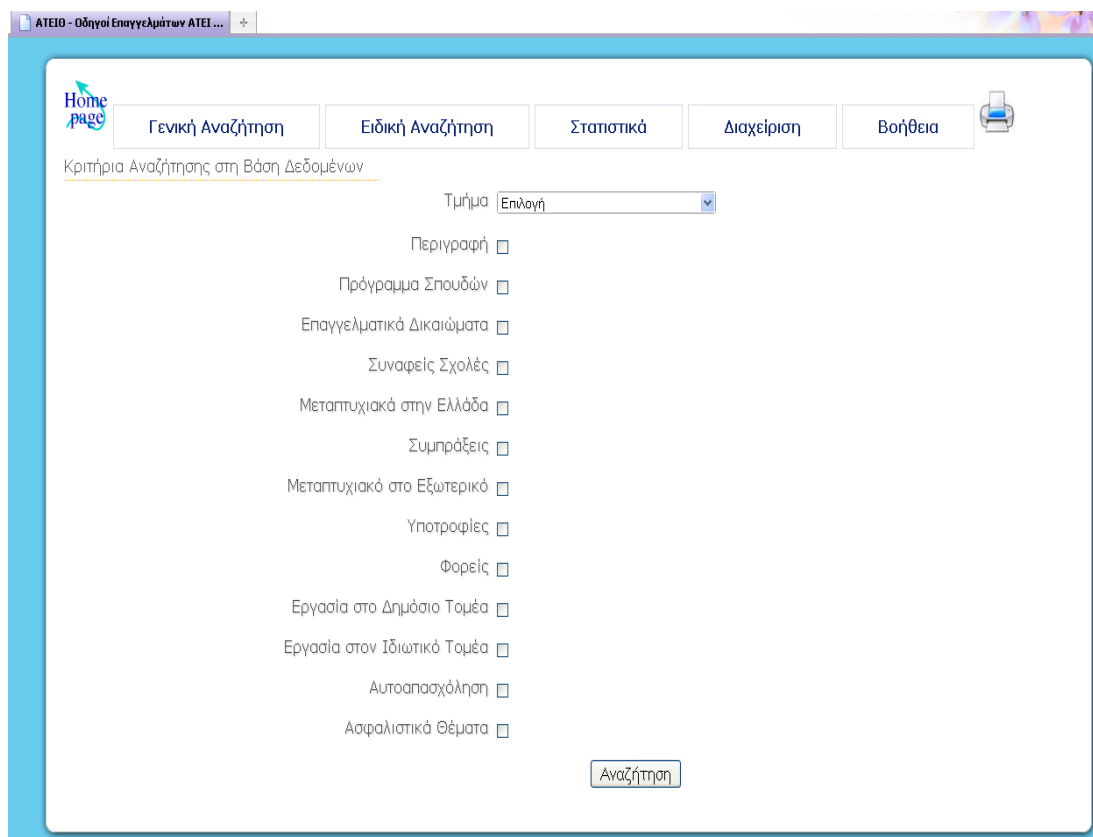
Αποτελέσματα 4

Όνομα	ΤΕΙ	Περιοχή	Πρ. Σπουδών	Επ. Δικαιώματα	Αμοιβή	Μετ. Ελλάδα	Σύμπτωση	Μετ. Εξωτερικό	Υποτροφ.	Φορείς	Δημ. Τομέας	Ιδ. Τομέας	Αυτοαπασχ.	Ασφαλ.
ΗΛΕΚΤΡΟΝΙΚΗΣ	Λαμίας	Στερεά Ελλάδα	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
ΔΑΣΟΠΟΝΙΑΣ & ΔΙΑΧΕΙΡΙΣΗΣ ΦΥΣΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ	Λαμίας	Στερεά Ελλάδα	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ	Λαμίας	Στερεά Ελλάδα	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Εικόνα 7-4 : Αποτελέσματα της αναζήτησης

7.4. Ειδική Αναζήτηση

Επιλέγοντας την Ειδική Αναζήτηση εμφανίζεται η παρακάτω σελίδα.



Εικόνα 7-5 : Φόρμα Ειδικής Αναζήτησης

Η Ειδική Αναζήτηση αφορά ένα συγκεκριμένο τμήμα και ο χρήστης έχει την δυνατότητα να επιλέξει ποια πεδία του οδηγού σπουδών του τμήματος θέλει να εμφανιστούν.

7.5. Στατιστικά

Η επιλογή Στατιστικά εμφανίζει στατιστικά στοιχεία για το σύνολο των τμημάτων.

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

The screenshot shows the ATEIO website interface with a navigation menu and four data tables. The tables are titled: Προγράμματα Σπουδών, Επαγγελματικά Δικαιώματα, Αμοιβές, and Μεταπτυχιακές Σπουδές στην Ελλάδα. Each table has three columns: Κριτήριο, Αριθμός Τμημάτων, and Ποσοστό %.

Προγράμματα Σπουδών		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχει πρόγραμμα σπουδών	84	98
Δεν υπάρχει πρόγραμμα σπουδών	2	2
Σύνολο	86	100

Επαγγελματικά Δικαιώματα		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν επαγγελματικά δικαιώματα	85	99
Δεν υπάρχουν επαγγελματικά δικαιώματα	1	1
Σύνολο	86	100

Αμοιβές		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν αναφορά σε αμοιβές	36	42
Δεν υπάρχει αναφορά σε αμοιβές	50	58
Σύνολο	86	100

Μεταπτυχιακές Σπουδές στην Ελλάδα		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν αναφορά σε αμοιβές	36	42
Δεν υπάρχει αναφορά σε αμοιβές	50	58
Σύνολο	86	100

Εικόνα 7-6 : Στατιστικά στοιχεία τμημάτων

The screenshot shows the ATEIO website interface with a navigation menu and four data tables. The tables are titled: Συμπράξεις ΤΕΙ με Παν/μια του Εξωτερικού, Μεταπτυχιακές Σπουδές στο Εξωτερικό, Υποτροφίες - Κληροδοτήματα, and Επαγγελματικοί - Επιστημονικοί Φορείς και Σύλλογοι. Each table has three columns: Κριτήριο, Αριθμός Τμημάτων, and Ποσοστό %.

Συμπράξεις ΤΕΙ με Παν/μια του Εξωτερικού		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν συμπράξεις ΤΕΙ με Παν/μια του Εξωτερικού	63	73
Δεν υπάρχουν συμπράξεις ΤΕΙ με Παν/μια του Εξωτερικού	23	27
Σύνολο	86	100

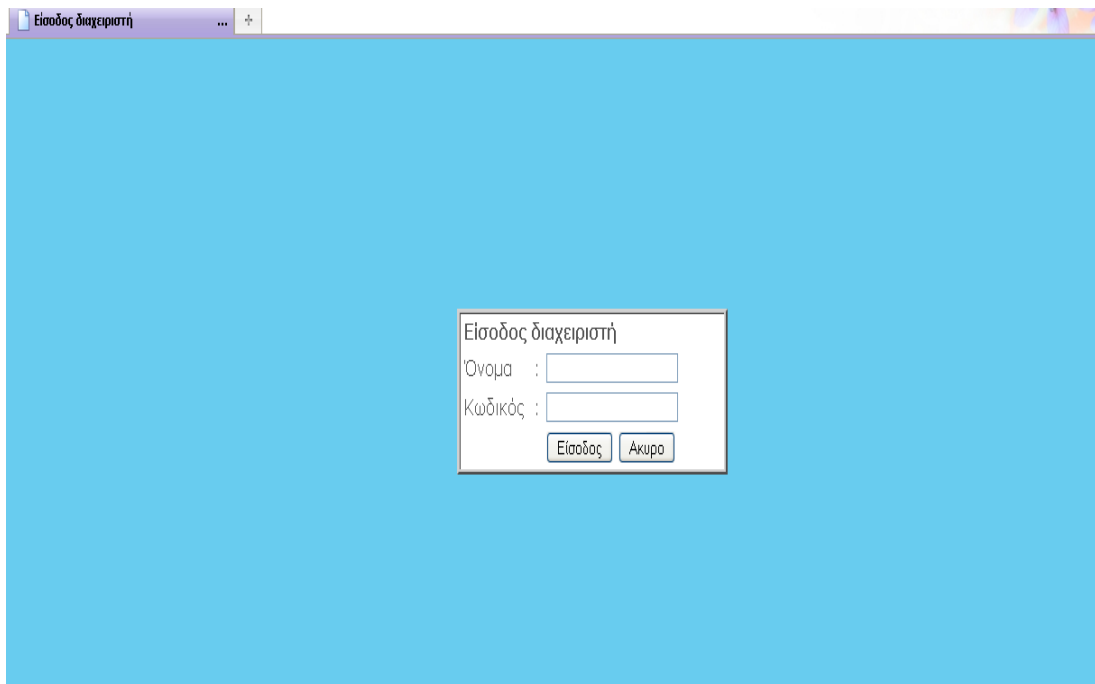
Μεταπτυχιακές Σπουδές στο Εξωτερικό		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν μεταπτυχιακές σπουδές στο εξωτερικό	70	81
Δεν υπάρχουν μεταπτυχιακές σπουδές στο εξωτερικό	16	19
Σύνολο	86	100

Υποτροφίες - Κληροδοτήματα		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν Υποτροφίες - Κληροδοτήματα	62	72
Δεν υπάρχουν Υποτροφίες - Κληροδοτήματα	24	19
Σύνολο	86	100

Επαγγελματικοί - Επιστημονικοί Φορείς και Σύλλογοι		
Κριτήριο	Αριθμός Τμημάτων	Ποσοστό %
Υπάρχουν Επαγγελματικοί - Επιστημονικοί φορείς και σύλλογοι	65	76
Δεν υπάρχουν Επαγγελματικοί - Επιστημονικοί φορείς και σύλλογοι	21	24

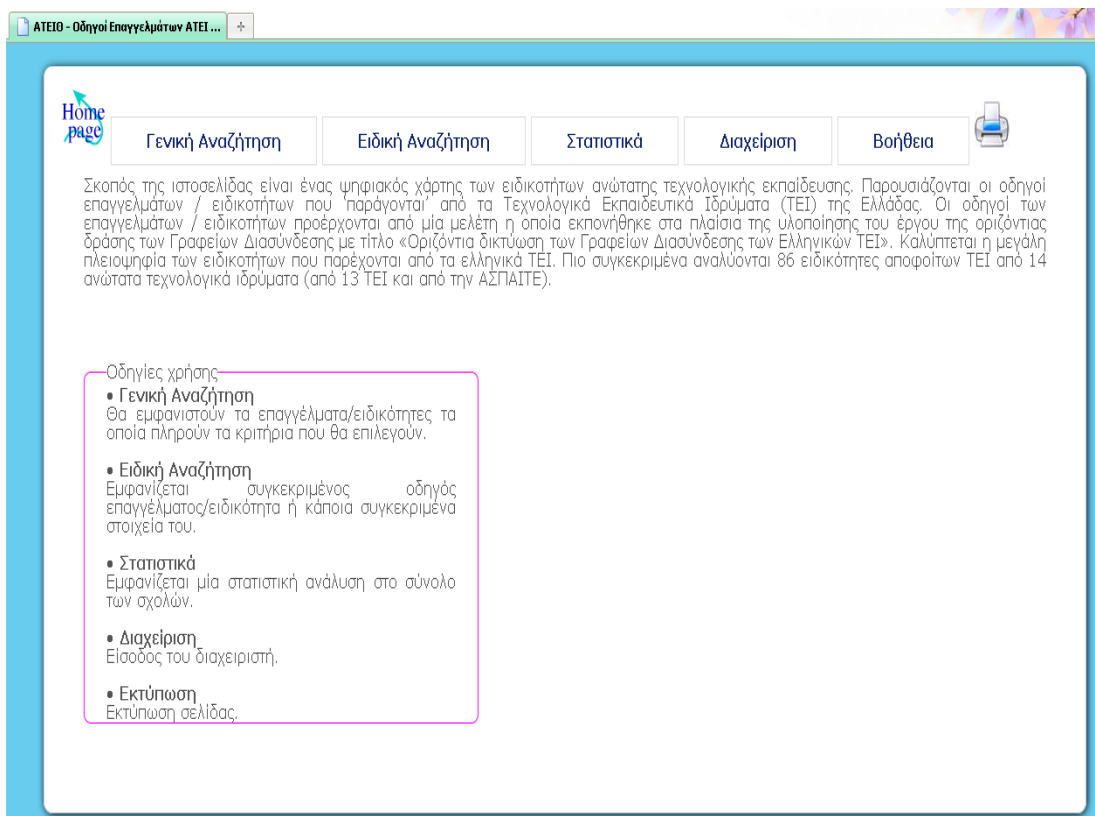
Εικόνα 7-7 : Στατιστικά στοιχεία τμημάτων 2

7.6. Διαχείριση



Εικόνα 7-8 : Είσοδος διαχειριστή

7.7. Βοήθεια



Εικόνα 7-9 : Σελίδα Βοήθειας

BIBΛΙΟΓΡΑΦΙΑ

Gehrke, Johannes, Ramakrishnan, και Raghu (2002), Συστήματα διαχείρισης βάσεων δεδομένων, Εκδόσεις Τζιόλα.

Meloni, J. (2006), Μάθετε PHP, Apache και MYSQL σε 24 ώρες, Εκδόσεις Μ. Γκιούρδας.

Σαλαμπάσης, Μ. (2008), Εισαγωγή στον προγραμματισμό διαδικτυακών εφαρμογών.

Willing, L. και Thomson, L. (2005), Ανάπτυξη Web Εφαρμογών με Php και Mysql.

Zandstra, M. (2000), Teach Yourself PHP4 in 24 Hours.

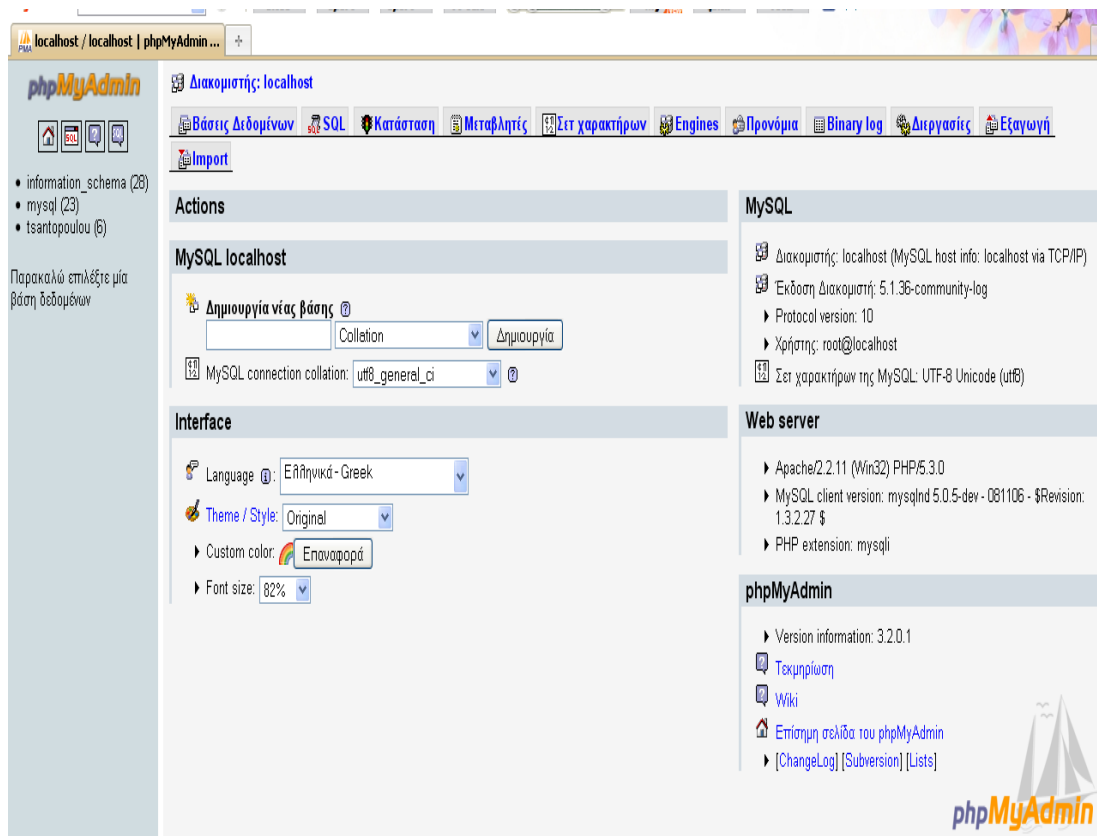
Πηγές στο Διαδίκτυο

- <http://www.php.net/>
- <http://myphp.gr/manual/el>
- <http://www.wikipedia.gr>
- <http://techteam.gr/>
- <http://www.freestuff.gr/forums/viewtopic.php?t=18943>

ΠΑΡΑΡΤΗΜΑ Α

Οδηγός Εγκατάστασης Της Εφαρμογής

Αφού έχουμε κάνει εγκατάσταση το Wamp ακολουθώντας τα βήματα που αναφέρθηκαν στο Κεφάλαιο 5 ανοίγουμε έναν φυλλομετρητή όπως τον Internet Explorer και γράφουμε τη διεύθυνση **http://localhost/phpmyadmin**. Αν η εγκατάσταση του Wamp ήταν επιτυχής θα εμφανιστεί η παρακάτω εικόνα:



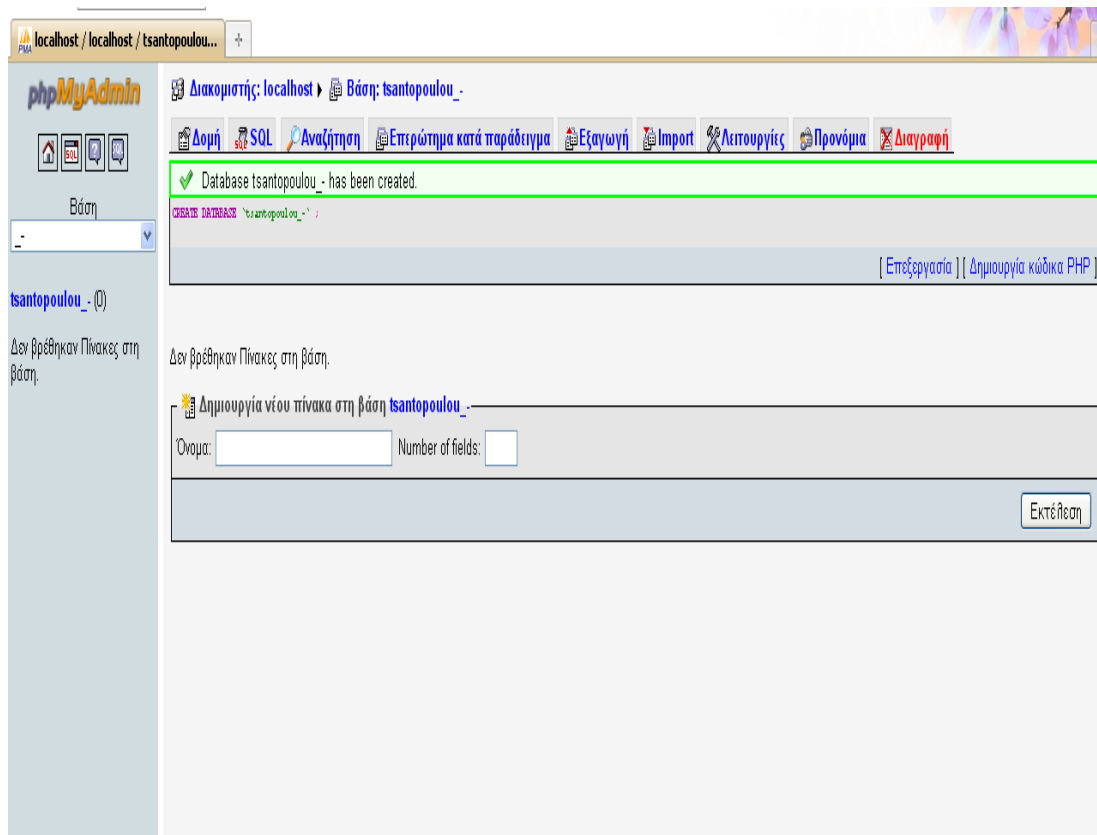
Εικόνα 1: Δημιουργία βάσης

Έχοντας ολοκληρώσει σωστά την διαδικασία που αναφέρθηκε παραπάνω έχουμε πια στον υπολογιστή μας ότι χρειαζόμαστε για να λειτουργήσει η εφαρμογή. Εκτός όμως από τα παραπάνω, θα πρέπει να γίνουν και μία σειρά βημάτων για μπορέσει να τρέξει σωστά.

Πρέπει να δημιουργήσουμε την βάση δεδομένων και στη συνέχεια να εισάγουμε τους πίνακες. Για τη δημιουργία της βάσης δεδομένων γράφουμε στο πεδίο «Δημιουργία νέας βάσης» tsantopoulou και πατάμε το κουμπί «Δημιουργία». μετά αυτή τη

Πτυχιακή εργασία της φοιτήτριας Τσαντοπούλου Βαΐα- Αικατερίνη

ενέργεια παρατηρούμε την βάση που μόλις δημιουργήσαμε στη δεξιά πλευρά μαζί με τις άλλες βάσεις που υπήρχαν. Την επιλέγουμε και εμφανίζεται η εικόνα:



Εικόνα 2 : Εισαγωγή δεδομένων

Επιλέγουμε «Import» προκειμένου να εισάγουμε τους πίνακες τις βάσης. Επιλέγουμε το κουμπί «Αναζήτηση» και εισάγουμε το αρχείο **Tables.txt** το οποίο βρίσκεται στο φάκελο CD με όλα τα αρχεία της εφαρμογής. Αφού γίνει η επιλογή, πατάμε «Εκτέλεση». Στη συνέχεια αντιγράφουμε από το CD τα αρχεία που βρίσκονται μέσα στον φάκελο αρχεία και κάνουμε επικόλληση στην διαδρομή: C/wamp/www.

Έχοντας ακολουθήσει όλα τα βήματα, το μόνο που έχουμε να ανοίγουμε τον Internet Explorer, να πληκτρολογήσουμε τη διεύθυνση : **http://localhost/** και θα εμφανιστεί η αρχική οθόνη της εφαρμογής.