



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



Πτυχιακή εργασία

Τίτλος:

**Πληροφοριακό σύστημα για την καθοδήγηση
χρηστών με χρήση GIS**

**Μανδουράρη Δήμητρα (ΑΜ: 2178)
Εισηγητής: Παπαδάκης Νίκος**

Abstract

This project aims to enhance the design and implementation of an information system to guide users to use GIS (Geographic Information System). The GIS system is a tool "smart map", which allows users to capture a snapshot of the real world, to create interactive spatial questions or descriptive data (searches created by the user) to analyze spatial data (spatial data), to adapt and perform in the analog channels (print maps and charts) or digital media (files of spatial data, interactive maps on the internet). Users of the site can have a picture of where are the points that interest them in search of a map through the special search field. More specifically, the search for points located in the prefecture of Lasithi. In developing the site using the web programming languages html, php javascript and MySQL and the database management system EasyPHP.

Σύνοψη

Η παρούσα πτυχιακή εργασία έχει ως σκοπό την ανάδειξη του σχεδιασμού και της υλοποίησης ενός πληροφοριακού συστήματος για την καθοδήγηση των χρηστών με τη χρήση GIS (Geographic Information System). Το σύστημα GIS ή αλλιώς Σύστημα Γεωγραφικών Πληροφοριών (ΣΓΠ) είναι ένα εργαλείο "έξυπνου χάρτη", το οποίο επιτρέπει στους χρήστες του να αποτυπώσουν μια περίληψη του πραγματικού κόσμου, να δημιουργήσουν διαδραστικά ερωτήσεις χωρικού ή περιγραφικού χαρακτήρα (αναζητήσεις δημιουργούμενες από τον χρήστη), να αναλύσουν τα χωρικά δεδομένα (spatial data), να τα προσαρμόσουν και να τα αποδώσουν σε αναλογικά μέσα (εκτυπώσεις χαρτών και διαγραμμάτων) ή σε ψηφιακά μέσα (αρχεία χωρικών δεδομένων, διαδραστικοί χάρτες στο Διαδίκτυο). Οι χρήστες του ιστοτόπου μπορούν να έχουν μια εικόνα για το που βρίσκονται τα σημεία που τους ενδιαφέρουν αναζητώντας τα στο χάρτη μέσω του ειδικού πεδίου αναζήτησης. Πιο συγκεκριμένα η αναζήτηση αφορά σημεία που βρίσκονται στο νομό Λασιθίου. Για την ανάπτυξη του δικτυακού τόπου χρησιμοποιήθηκαν οι γλώσσες προγραμματισμού διαδικτύου html, php, javascript και MySQL καθώς και το σύστημα διαχείρισης βάσεων δεδομένων EasyPHP.

Πίνακας Περιεχομένων

Abstract	iii
Σύνοψη	iv
1 Εισαγωγή	8
1.1 Περίληψη.....	8
1.2 Στόχοι Εργασίας	8
1.3 Δομή Εργασίας.....	8
2 Γλώσσες και εφαρμογές που χρησιμοποιήθηκαν	10
2.1 <i>PHP</i>	10
2.1.1 Τι είναι η <i>PHP</i>	10
2.1.2 Τι μπορεί να κάνει η <i>PHP</i>	11
2.1.3 Τα πλεονεκτήματα της <i>PHP</i>	11
2.1.4 <i>PHP Sessions</i>	12
Σύνταξη των <i>PHP Sessions</i>	12
Δήλωση ενός <i>Session</i>	13
Χρήση της λειτουργίας <i>isset</i> της <i>PHP</i>	13
Καθαρισμός και Καταστροφή της <i>Session</i>	13
2.2 <i>Mysql</i>	14
2.2.1 Τι είναι η <i>MySQL</i>	14
2.2.2 Τα πλεονεκτήματα της <i>MySQL</i>	15
2.3 <i>Javascript</i>	15
2.3.1 Τι είναι <i>Javascript</i>	15
2.3.2 Δυνατότητες της <i>Javascript</i>	16
2.3.3 Τρόπος Λειτουργίας της <i>JavaScript</i>	17
2.4 <i>JSP(JavaServer Pages)</i>	17
2.4.1 Τι είναι <i>JSP</i>	17
2.4.2 Αρχιτεκτονική web εφαρμογής με <i>JSP</i>	17
2.5 <i>HTML</i>	18
2.5.1 Τι είναι <i>html</i> ;	18
2.5.2 Εκδόσεις <i>HTML</i>	18
2.5.3 Γλώσσες Προγραμματισμού στο <i>Internet</i>	19
2.5.4 Δημιουργία αρχείων <i>HTML</i>	19
2.6 <i>CSS</i>	19
2.6.1 Τι είναι η <i>CSS</i>	20
2.6.2 Πλεονεκτήματα <i>CSS</i>	21
2.7 Εγκατάσταση <i>EasyPHP</i>	21
2.8 <i>Macromedia Dreamweaver</i>	27
3 Γεωγραφικά Συστήματα Πληροφοριών (GIS)	29
3.1 Εισαγωγή.....	29
3.2 Ιστορία των <i>GIS</i>	29
3.3 Χαρακτηριστικά των <i>GIS</i>	30
3.4 Πλεονεκτήματα	31
4 Google Maps API	32
4.1 Διεπαφή Προγραμματισμού Εφαρμογών <i>Google Maps</i>	32
4.1.1 <i>Events</i>	32

4.1.2 Controls	32
4.1.3 Overlays.....	33
5 Υλοποίηση Βάσεων Δεδομένων.....	35
5.1 <i>Τι είναι βάση δεδομένων.....</i>	35
5.1.1 Το σύστημα διαχείρισης βάσεων δεδομένων	35
5.1.2 Πλεονεκτήματα της χρήσης βάσεων δεδομένων	36
5.2 <i>Το Μοντέλο Οντοτήτων-Συσχετίσεων.....</i>	37
5.2.1 Βασικές έννοιες Μοντέλου Οντοτήτων-Συσχετίσεων	38
Οντότητα	38
Χαρακτηριστικό	38
Συσχέτιση	39
Βαθμός ή πολυπλοκότητα ενός τύπου συσχετίσεων	40
Πληθικότητα.....	40
Ασθενής Τύπος Οντότητας.....	41
5.3 <i>Η βάση δεδομένων maps</i>	41
5.3.1 Πίνακας city_info	41
5.3.2 Πίνακας points.....	42
5.3.3 Πίνακας points_categories.....	43
5.4 <i>Η βάση δεδομένων syst_kathodigisis.....</i>	43
5.4.1 Πίνακας users	44
5.4.2 Πίνακας cinema	45
5.4.3 Πίνακας cinema_info.....	46
5.4.4 Πίνακας super_market.....	47
5.4.5 Πίνακας supermarket_info	47
5.4.6 Πίνακας venzinadiko	48
5.4.7 Πίνακας venzinadiko_info.....	49
6 Αρχιτεκτονική Εφαρμογής.....	50
7 Αποτελέσματα.....	54
7.1 Συμπεράσματα.....	54
7.2 Μελλοντική Εργασία και Επεκτάσεις.....	54
Βιβλιογραφία	55
ΠΑΡΑΡΤΗΜΑΤΑ	56
ΠΑΡΑΡΤΗΜΑ Α: Κώδικας.....	57
ΠΑΡΑΡΤΗΜΑ Β: Παρουσίαση	74
ΠΑΡΑΡΤΗΜΑ Γ: Περίληψη	81

Πίνακας Εικόνων

Εικόνα 1 Οι φάσεις επεξεργασίας μιας JSP σελίδας	18
Εικόνα 2 Εγκατάσταση EasyPHP.....	21
Εικόνα 3 Επιλογή αποδοχής των όρων χρήσης.....	22
Εικόνα 4 Επιλογή next στις πληροφορίες	23
Εικόνα 5 Επιλογή διαδρομής για την εγκατάσταση.....	23
Εικόνα 6 Επιλογή ονόματος φακέλου	24
Εικόνα 7 Επιλογή Install	24
Εικόνα 8 Επιλογή Finish.....	25
Εικόνα 9 EasyPHP	25
Εικόνα 10 Αρχική σελίδα της EasyP	26
Εικόνα 11 Dreamweaver.....	28
Εικόνα 12 Οντότητα.....	38
Εικόνα 13 Χαρακτηριστικό	38
Εικόνα 14 Συσχέτιση.....	39
Εικόνα 15 Βαθμός ή πολυπλοκότητα.....	40
Εικόνα 16 Η βάση δεδομένων maps	41
Εικόνα 17 Πίνακας city_info	42
Εικόνα 18 Πίνακας points.....	43
Εικόνα 19 Πίνακας points_categories.....	43
Εικόνα 20 Η βάση δεδομένων syst_kathodigisis.....	44
Εικόνα 21 Πίνακας users	45
Εικόνα 22 Πίνακας cinema	46
Εικόνα 23 Πίνακας cinema_info	46
Εικόνα 24 Πίνακας super_market	47
Εικόνα 25 Πίνακας supermarket_info.....	48
Εικόνα 26 Πίνακας venzinadiko	48
Εικόνα 27 Πίνακας venzinadiko_info.....	49
Εικόνα 28 Αρχική σελίδα.....	50
Εικόνα 29 Παράδειγμα αναζήτησης	51
Εικόνα 30 Η αναζήτηση δεν πέτυχε.....	51
Εικόνα 31 Η είσοδος απέτυχε.....	52
Εικόνα 32 Register	52
Εικόνα 33 Η υποβολή απέτυχε	53

1 Εισαγωγή

1.1 Περίληψη

Η ταχύτητα αναπτυσσόμενη τεχνολογία στις μέρες μας, μας παρέχει τη δυνατότητα μετάδοσης της πληροφορίας χωρίς ιδιαίτερο κόστος. Τα Γεωγραφικά Συστήματα Πληροφοριών γνωστά ευρέως και ως GIS (Geographic Information System), είναι συστήματα διαχείρισης χωρικών δεδομένων (spatial data) και συσχετισμένων ιδιοτήτων. Ένα GIS στην πιο αυστηρή μορφή του είναι ένα ψηφιακό σύστημα, ικανό να ενσωματώσει, αποθηκεύσει, προσαρμόσει, αναλύσει και παρουσιάσει γεωγραφικά συσχετισμένες (geographically referenced) πληροφορίες. Σε πιο γενική μορφή, ένα ΓΣΠ είναι ένα εργαλείο "έξυπνου χάρτη", το οποίο επιτρέπει στους χρήστες του :

- να αποτυπώσουν μια περίληψη του πραγματικού κόσμου
- να δημιουργήσουν διαδραστικά ερωτήσεις χωρικού ή περιγραφικού χαρακτήρα (αναζητήσεις δημιουργούμενες από τον χρήστη)
- να αναλύσουν τα χωρικά δεδομένα (spatial data)
- να τα προσαρμόσουν και να τα αποδώσουν σε αναλογικά μέσα (εκτυπώσεις χαρτών και διαγραμμάτων) ή σε ψηφιακά μέσα (αρχεία χωρικών δεδομένων, διαδραστικοί χάρτες στο Διαδίκτυο).

Σκοπός της πτυχιακής αυτής ήταν ο σχεδιασμός και η υλοποίηση ενός πληροφοριακού συστήματος με το οποίο θα γινόταν η καθοδήγηση των χρηστών του ιστοτόπου. Πιο συγκεκριμένα, η καθοδήγηση αφορά περιοχές που ανήκουν στο νομό Λασιθίου.

Για την υλοποίηση του συστήματος χρησιμοποιήθηκαν γλώσσες προγραμματισμού όπως η Javascript, η HTML, η CSS, η PHP και η MySQL, καθώς και το πακέτο λογισμικού EasyPHP, που υποστηρίζει τις δυο τελευταίες. Επίσης χρησιμοποιήθηκε η υπηρεσία Google Maps API, μία διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface - API) της Google μέσω της οποίας δόθηκε η δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη στην ιστοσελίδα.

1.2 Στόχοι Εργασίας

Ο στόχος της πτυχιακής εργασίας ήταν η κατασκευή μιας δυναμικής ιστοσελίδας, ενός πληροφοριακού συστήματος, που σκοπό θα είχε την καθοδήγηση των χρηστών της σε μια ορισμένη γεωγραφική περιοχή. Για να δημιουργηθεί μια δυναμική ιστοσελίδα δεν αρκούν μόνο γλώσσες προγραμματισμού όπως η HTML και η Javascript, αλλά χρειάζονται πιο εξειδικευμένες γλώσσες όπως είναι η PHP.

Με το πακέτο λογισμικού EasyPHP και τις γλώσσες PHP και MySQL μπορεί να γίνει η διαχείριση της βάσης δεδομένων. Με τη βοήθεια της βάσης δεδομένων τοποθετούνται markers στο χάρτη και οι χρήστες έχουν τη δυνατότητα να εμφανίζουν μόνο τα σημεία για τα οποία ενδιαφέρονται (π.χ. μόνο τα πρατήρια βενζίνης ή/και τους κινηματογράφους). Επίσης υπάρχει πεδίο αναζήτησης στο οποίο ο χρήστης μπορεί να εισάγει τη διεύθυνση την οποία ψάχνει. Το σύστημα, στην περίπτωση αυτή, θα αναζητήσει το σημείο εκείνο και αν υπάρχει στη βάση θα κάνει ζουμ, δημιουργώντας έτσι μια εικόνα του πραγματικού κόσμου.

1.3 Δομή Εργασίας

Στο κεφάλαιο 2 της πτυχιακής παρουσιάζονται οι γλώσσες, οι εφαρμογές και τα προγράμματα που χρησιμοποιήθηκαν για την εκπόνηση της.

Στο κεφάλαιο 3, παρουσιάζονται συνοπτικά τα Γεωγραφικά Πληροφοριακά Συστήματα (GIS).

Στο κεφάλαιο 4, παρουσιάζονται το Google Maps API και οι μηχανισμοί του.

Στο κεφάλαιο 5, παρουσιάζεται η θεωρία γύρω από τις βάσεις δεδομένων, καθώς και η υλοποίηση των δύο βάσεων που χρησιμοποιήθηκαν για τη δημιουργία του ιστοτόπου.

Στο κεφάλαιο 6, αναλύεται η αρχιτεκτονική της εφαρμογής που δημιουργήθηκε.

Τέλος στο κεφάλαιο 7, παρουσιάζονται τα συμπεράσματα καθώς και οι μελλοντικές επεκτάσεις που μπορούν να γίνουν στο σύστημα.

2 Γλώσσες και εφαρμογές που χρησιμοποιήθηκαν

2.1 PHP



Σκοπός αυτής της πτυχιακής είναι να επιτρέψει να δημιουργήσουμε πραγματικές δυναμικές web τοποθεσίες. Αν δημιουργήσουμε μια web τοποθεσία χρησιμοποιώντας απλή HTML, θα καταλάβουμε τα όρια αυτής της προσέγγισης. Τα στατικά περιεχόμενα μιας Web τοποθεσίας, είναι απλώς στατικά και παραμένουν τα ίδια, εκτός και αν ενημερωθούν άμεσα. Οι χρήστες δεν μπορούν να συνδιαλέγονται με την τοποθεσία με κάποιο ουσιαστικό τρόπο.

Χρησιμοποιώντας μια γλώσσα, όπως την PHP και μια βάση δεδομένων, όπως τη MySQL, υπάρχει η δυνατότητα να γίνουν οι τοποθεσίες δυναμικές, ηλαδή να είναι προσαρμόσιμες και να περιέχουν ενημερωμένες πληροφορίες.

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Σήμερα περισσότερα από 16.000.000 web sites, ποσοστό μεγαλύτερο από το 35% των ιστοσελίδων του Διαδικτύου, χρησιμοποιούν scripts γραμμένα με τη γλώσσα PHP, ενώ το υπόλοιπο 65% το μοιράζονται στατικές σελίδες HTML και όλες οι άλλες γλώσσες προγραμματισμού.

2.1.1 Τι είναι η PHP

Η PHP έχει εξαιρετική ομοιότητα με τις γλώσσες Perl και C/C++ στις οποίες έχει τις ρίζες της, ως προς τον τρόπο σύνταξης, αλλά και πολλές εντολές της.

Είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από συμβατό web serve(π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο server Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με λειτουργικά συστήματα Linux και Microsoft Windows, υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.

2.1.2 Τι μπορεί να κάνει η PHP

Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορούμε να κάνουμε οτιδήποτε μπορεί ένα άλλο CGI πρόγραμμα να κάνει, όπως η συλλογή δεδομένων, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στέλνουμε και να λαμβάνουμε cookies. Αλλά η PHP μπορεί να κάνει πολύ περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- **Server-side scripting.** Αυτό είναι το πιο παραδοσιακό και κύριο πεδίο για την PHP. Χρειαζόμαστε τρία πράγματα για να δουλέψει. Ένα PHP μεταγλωττιστή (CGI ή server module), ένα web server και ένα web browser. Ο web server θα πρέπει να τρέξει με μια συνδεδεμένη εγκατάσταση της PHP. Μπορούμε να έχουμε πρόσβαση στην έξοδο του προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.
- **Command line scripting.** Μπορούμε να φτιάξουμε ένα PHP script και να το τρέξουμε χωρίς server ή browser. Χρειαζόμαστε μόνο τον PHP μεταγλωττιστή. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται με χρήση cron (σε * nix ή Linux) ή Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.
- **Σύνταξη desktop εφαρμογών.** Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να δημιουργήσει κανείς μια εφαρμογή με γραφικό περιβάλλον χρήστη, αλλά αν γνωρίζει PHP πολύ καλά και θέλει να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές, μπορεί επίσης να χρησιμοποιήσει το PHP-GTK για τέτοιου είδους προγράμματα. Υπάρχει επίσης η δυνατότητα εγγραφής cross-platform εφαρμογών με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή.

2.1.3 Τα πλεονεκτήματα της PHP

Η PHP μπορεί να χρησιμοποιηθεί για όλα τα σημαντικά λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS, και πιθανώς άλλα. Η PHP υποστηρίζει επίσης τους περισσότερους διακομιστές web σήμερα, όπως τους Apache, IIS και πολλούς άλλους. Και αυτό περιλαμβάνει οποιοδήποτε web server ο οποίος μπορεί να χρησιμοποιήσει το FastCGI PHP binary, όπως lighttpd και Nginx. Η PHP δουλεύει είτε ως module, είτε ως επεξεργαστής CGI.

Έτσι με την PHP υπάρχει ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, υπάρχει η δυνατότητα χρήσης συναρτησιακού (procedural programming) ή αντικειμενοστραφή προγραμματισμού (OOP), ή ένα μείγμα από τους δύο.

Με την PHP δεν υπάρχει περιορισμός εξαγωγής HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας libswf και Ming), οι οποίες παράγονται αμέσως. Μπορεί επίσης να γίνει εύκολα οποιοδήποτε εξαγωγή κειμένου όπως XHTML και οποιοδήποτε άλλο αρχείο XML. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό μας περιεχόμενο.

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

Η PHP υποστηρίζει επίσης την επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα. Υπάρχει επίσης η δυνατότητα να ανοιχτούν raw network sockets και να αλληλεπιδράσουν με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Μιλώντας για τη διασύνδεση, η PHP υποστηρίζει instantiation αντικειμένων Java και τα χρησιμοποιεί διάφανα σαν αντικείμενα PHP. Μπορεί επίσης να χρησιμοποιηθεί η CORBA επέκταση για να προσπελαστούν remote (απομακρυσμένα) αντικείμενα.

Η PHP έχει χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, τα οποία περιλαμβάνουν την Perl compatible regular expressions (PCRE), και πολλές άλλες επεκτάσεις και εργαλεία για ανάλυση και πρόσβαση σε έγγραφα XML. Η PHP τυποποιεί όλες τις επεκτάσεις XML στη σταθερή βάση των libxml2, και επεκτείνει το σύνολο των δυνατοτήτων της προσθέτοντας την υποστήριξη των SimpleXML, XMLReader και XMLWriter.

Καθώς χρησιμοποιείται η PHP στον τομέα του ecommerce, βρίσκονται οι Cybercash payment, CyberMUT, VeriSign Payflow Pro και CCVS συναρτήσεις χρήσιμες για τα online προγράμματα πληρωμής.

Τελευταίο αλλά σημαντικό, έχουμε πολλές άλλες ενδιαφέρουσες επεκτάσεις, τις mmoGoSearch search engine συναρτήσεις, πολλά εργαλεία συμπίεσης (gzip, bz2), μετατροπές ημερολογίου, μεταφράσεις...

2.1.4 PHP Sessions

Όταν μια ιστοσελίδα γίνεται όλο και πιο πολύπλοκη, γίνεται πολύπλοκος και ο κώδικας που το υποστηρίζει. Όταν φτάσουμε σε ένα στάδιο όπου η ιστοσελίδα πρέπει να περάσει τα δεδομένα του χρήστη από τη μία σελίδα στην άλλη, ίσως ήρθε η ώρα να αρχίσουμε να σκεφτόμαστε σχετικά με τη χρήση των PHP Sessions.

Μια κανονική ιστοσελίδα HTML δεν θα περάσει τα δεδομένα από τη μία σελίδα στην άλλη. Με άλλα λόγια, όλες οι πληροφορίες θα έχουν ξεχαστεί όταν μια νέα σελίδα φορτωθεί. Αυτό δημιουργεί πρόβλημα για εργασίες όπως ένα καλάθι αγορών, το οποίο απαιτεί τα δεδομένα (επιλεγμένο προϊόν του χρήστη) που πρέπει να θυμόμαστε από τη μία σελίδα στην άλλη. Στην περίπτωση της παρούσας εργασίας όταν το Latitude και το Longitude πρέπει να περάσει στην επόμενη σελίδα, ώστε να φορτωθεί ο χάρτης με αυτά τα δεδομένα ή το username και το password όταν ο χρήστης κάνει login. Ένα PHP Session λύνει αυτό το πρόβλημα επιτρέποντάς μας να αποθηκεύσουμε πληροφορίες του χρήστη στον διακομιστή για να χρησιμοποιηθούν αργότερα (δηλαδή το Latitude και το Longitude κ.λπ.).

Σύνταξη των PHP Sessions

Πριν ξεκινήσουμε την αποθήκευση των στοιχείων του χρήστη, πρέπει να ξεκινήσουμε για πρώτη φορά τα Sessions. Αυτό γίνεται στην αρχή του κώδικα PHP, πριν να σταλεί κάποιο Html ή άλλο κείμενο.

Παρακάτω φαίνεται μια απλή δέσμη ενεργειών που θα πρέπει να τοποθετηθεί στην αρχή του κώδικα PHP μας για να ξεκινήσει ένα PHP Session.

```
<?php
session_start(); // start up your PHP session!
?>
```

Αυτό το μικροσκοπικό κομμάτι του κώδικα καταχωρεί το Session του χρήστη με τον server και μας επιτρέπει να ξεκινήσουμε την αποθήκευση πληροφοριών του χρήστη και να εκχωρεί μια UID (μοναδικό αναγνωριστικό αριθμό) για τη συνεδρία του συγκεκριμένου χρήστη.

Δήλωση ενός Session

Όταν θέλουμε να αποθηκεύσουμε τα δεδομένα ενός χρήστη σε μια σύνοδο χρησιμοποιούμε το `$_SESSION`. Με αυτό γίνεται η αποθήκευση και η ανάκτηση των δεδομένων του Session.

```
<?php
session_start();
$_SESSION['views'] = 1; // store session data
echo "Pageviews = ". $_SESSION['views']; //retrieve data
?>
```

Χρήση της λειτουργίας `isset` της PHP

Τώρα που είμαστε σε θέση να αποθηκεύσουμε και να ανακτήσουμε δεδομένα από το `$_SESSION`, μπορούμε να διερευνήσουμε ορισμένες από τις πραγματικές λειτουργίες των Sessions. Όταν δημιουργείται μια μεταβλητή και αποθηκεύεται σε μια περίοδο, πιθανόν να θέλουμε να τη χρησιμοποιήσουμε στο μέλλον. Ωστόσο, προτού τη χρησιμοποιήσουμε είναι απαραίτητο να ελέγξουμε αν υπάρχει.

Σε αυτό το σημείο είναι όπου η λειτουργία `isset` της PHP θα μας βοηθήσει. Η `isset` είναι μια λειτουργία που θα λαμβάνει κάθε μεταβλητή που θέλουμε να χρησιμοποιήσουμε και ελέγχει για να δει αν έχει τεθεί. Δηλαδή, αν της έχει ήδη ανατεθεί μια τιμή.

Με το προηγούμενο παράδειγμα, μπορούμε να δημιουργήσουμε μια πολύ απλή προβολή σελίδας με τη χρήση της `isset` για να ελέγξουμε αν η μεταβλητή έχει ήδη δημιουργηθεί. Αν έχει μπορούμε να προσάξουμε τον counter μας. Αν δεν υπάρχει μπορούμε να δημιουργήσουμε έναν μετρητή προβολής σελίδας και να του δώσουμε την τιμή ένα. Εδώ είναι ο κώδικας για να γίνει αυτό:

```
<?php
session_start();
if(isset($_SESSION['views']))
    $_SESSION['views'] = $_SESSION['views']+ 1;
else
    $_SESSION['views'] = 1;

echo "views = ". $_SESSION['views'];
?>
```

Η πρώτη φορά που εκτελείτε αυτήν τη δέσμη ενεργειών σε ένα πρόσφατα ανοιγμένο browser, η δήλωση `if` θα αποτύχει διότι καμία μεταβλητή Session δεν θα έχει αποθηκευτεί ακόμα. Ωστόσο, εάν ανανεώσουμε τη σελίδα, η δήλωση `if` θα ήταν `true` και ο μετρητής θα αυξάνονται κατά ένα. Κάθε φορά που ξανατρέχουμε αυτό το σενάριο θα βλέπουμε μια αύξηση κατά ένα του counter.

Καθαρισμός και Καταστροφή της Session

Αν και τα δεδομένα Session είναι προσωρινά και δεν χρειάζεται να γίνει διαγραφή, ίσως θελήσουμε να διαγράψουμε μερικά δεδομένα για τις διάφορες εργασίες μας. Όπως για παράδειγμα μετά από μια συναλλαγή, όπου ο χρήστης έχει δώσει τα στοιχεία του, για παράδειγμα τον αριθμό της πιστωτικής του κάρτας.

```
<?php
session_start();
```

```
if(isset($_SESSION['cart']))  
    unset($_SESSION['cart']);  
?>
```

Μπορούμε επίσης να καταστρέψουμε εντελώς το Session εξ ολοκλήρου με την κλήση της λειτουργίας `session_destroy`.

```
<?php  
session_start();  
session_destroy();  
?>
```

Η καταστροφή του Session θα επαναφέρει τη συνεδρία μας, και όλα τα αποθηκευμένα δεδομένα μας θα χαθούν.

2.2 Mysql



2.2.1 Τι είναι η MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά του από την κόρη του Μόντυ Βιντένιους, την Μάι. Η φράση SQL σημαίνει Structured Query Language, δηλαδή Δομημένη Γλώσσα Ερωτημάτων ή Ερωταποκρίσεων.

Η MySQL είναι ένας SQL Database Server. Ένας server είναι ένα κομμάτι λογισμικού (software) που τρέχει (εκτελείται) στο παρασκήνιο σ' έναν υπολογιστή. Μια βάση δεδομένων (database) είναι μια δομημένη συλλογή από δεδομένα. Μια βάση δεδομένων είναι έτσι δομημένη ώστε να μπορούμε να βρίσκουμε πληροφορίες εύκολα και γρήγορα ανάλογα με το είδος των δεδομένων που περιέχει.

Προγράμματα ελεύθερου λογισμικού ανοικτού κώδικα τα οποία απαιτούν ένα πλήρως εξοπλισμένο σύστημα διαχείρισης βάσεων δεδομένων συχνά χρησιμοποιούν MySQL. Για εμπορική χρήση, διατίθενται πολλές έμμισθες εκδόσεις, οι οποίες προσφέρουν επιπλέον λειτουργικότητα. Μερικά παραδείγματα έργων λογισμικού ανοικτού κώδικα: Joomla, WordPress, MyBB, phpBB, Drupal και άλλα λογισμικά είναι χτισμένα στην LAMP στοίβα λογισμικού. Η MySQL χρησιμοποιείται επίσης σε πολλές υψηλού προφίλ, μεγάλης κλίμακας World Wide Web προϊόντα, συμπεριλαμβανομένου του Wikipedia, Google (αν και δεν είναι για αναζητήσεις) και το Facebook.

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

2.2.2 Τα πλεονεκτήματα της MySQL

Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL , Microsoft SQL και Oracle.

Η MySQL έχει πολλά πλεονεκτήματα , όπως χαμηλό κόστος , εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος.

- Απόδοση

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Πολλές από τις δοκιμές που έχουν γίνει δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

- Χαμηλό κόστος

Η MySQL είναι διαθέσιμη δωρεάν , με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος , αν παρθεί εμπορική άδεια, εφόσον απαιτείται από την εφαρμογή.

- Ευκολία Χρήσης

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν χρησιμοποιείται ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων δεν θα υπάρξει πρόβλημα προσαρμογής σε αυτό.

- Μεταφερσιμότητα

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα Unix όπως επίσης και στα Microsoft Windows .

- Κώδικας Προέλευσης

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

2.3 Javascript

Η JavaScript αναπτύχθηκε αρχικά από τον Brendan Eich του Netscape με το όνομα Μόκα, η οποία αργότερα μετονομάστηκε σε LiveScript, και τελικά σε JavaScript. LiveScript ήταν το επίσημο όνομα για τη γλώσσα, όταν εισήχθη για πρώτη φορά σε beta εκδόσεις του Netscape Navigator 2,0 τον Σεπτέμβριο του 1995, αλλά μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την Sun Microsystems τον Δεκέμβριο του 1995, όταν προοριζόταν να αναπτυχθεί στη νέα έκδοση του Netscape Navigator 2.0B3.

2.3.1 Τι είναι Javascript

Η JavaScript, επίσης γνωστή ως ECMAScript , είναι μία πρότυπη object-oriented scripting γλώσσα η οποία είναι δυναμική και έχει πρώτης τάξεως λειτουργίες. Θεωρείται επίσης μια λειτουργική γλώσσα προγραμματισμού , όπως τις Scheme και OCaml.

Η JavaScript είναι μια εφαρμογή του προτύπου γλώσσας ECMAScript και χρησιμοποιείται κυρίως με τη μορφή client-side Javascript, όπου υλοποιείται ως μέρος ενός web browser, ώστε να παρέχονται ενισχυμένη διεπαφή χρηστών και δυναμικές ιστοσελίδες. Αυτό επιτρέπει την πρόσβαση μέσω προγραμματισμού στα υπολογιστικά αντικείμενα μέσα σε ένα περιβάλλον υποδοχής.

Η χρήση JavaScript σε εφαρμογές έξω από ιστοσελίδες, όπως για παράδειγμα σε έγγραφα PDF, site-specific προγράμματα περιήγησης και desktop widgets, είναι επίσης σημαντική. Νεότερα και ταχύτερα VMs Javascript και πλαίσια βασισμένα πάνω σε αυτές (κυρίως Node.js), παρουσίασαν επίσης αύξηση της δημοτικότητας του Javascript για server-side web εφαρμογές.

Η JavaScript χρησιμοποιεί σύνταξη επηρεασμένη από αυτή της C. Πολλά ονόματα και συμβάσεις ονοματολογίας έχουν αντιγραφεί από τη Java, αλλά οι δύο γλώσσες είναι διαφορετικές και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού στο πλαίσιο της JavaScript, λαμβάνονται από Self και Scheme γλώσσες προγραμματισμού.

Είναι μια γλώσσα συγγραφής σεναρίων (scripting language) για την προσθήκη διαδραστικότητας (interactivity) σε ιστοσελίδες. Δημιουργήθηκε από την εταιρεία Netscape με αρχικό όνομα LiveScript και είναι πολύ δημοφιλής στους δημιουργούς ιστοσελίδων καθώς είναι απλή στη σύνταξή της και υποστηρίζεται απ' όλους τους δημοφιλείς φυλλομετρητές. Ανταγωνιστής της είναι η γλώσσα συγγραφής σεναρίων VBScript της εταιρείας Microsoft. Μια άλλη προσέγγιση στον προγραμματισμό από την πλευρά του πελάτη είναι η γλώσσα προγραμματισμού JavaScript. Για τον προγραμματισμό σε JavaScript πρέπει να εισάγετε δηλώσεις προγραμματισμού *απευθείας* στον κώδικα HTML. Αυτός ο κώδικας αλληλεπιδρά με τον browser, για παράδειγμα παράγει οπτικά εφέ, όπως rollover. Αυτό βρίσκεται σε αντίθεση με τα applet όπου ο κώδικας βρίσκεται στον διακομιστή, υπάρχει μόνο αναφορά για αυτόν στον κώδικα HTML, φορτώνεται στον πελάτη χρησιμοποιώντας την αναφορά και εκτελείται. [8]

2.3.2 Δυνατότητες της Javascript

Με τη JavaScript μπορούμε να φτιάξουμε σεναρία που να εκτελούν αυτόματες εργασίες, για παράδειγμα όταν μια σελίδα του Web ανοίγει ή κλείνει. Επίσης μπορούμε να κάνουμε την JavaScript να εκτελεί ενέργειες ανταποκρινόμενη σε ένα συγκεκριμένο γεγονός. Για παράδειγμα όταν ο χρήστης επιλέγει ένα κουμπί ή ένα σύνδεσμο, όταν εστιάζει από ένα στοιχείο μιας φόρμας σε ένα άλλο στοιχείο της . Τα σεναρία μπορεί να ανοίγουν νέα παράθυρα στον browser και να εμφανίζουν συγκεκριμένα HTML έγγραφα ή να παρουσιάζουν μια σελίδα επιλεγμένη από τον κατάλογο ιστορικού του browser. Μπορεί επίσης να είναι και περίπλοκες δηλαδή ένα σενάριο μπορεί να ελέγχει τα περιεχόμενα μιας φόρμας που θέλει να υποβάλει ένας χρήστης και στη συνέχεια να προειδοποιεί τον χρήστη αν τα δεδομένα είναι λάθος. Το σενάριο μπορεί να ψάξει για πληροφορίες σε μια μικρή βάση δεδομένων ή να κάνει πολύπλοκους υπολογισμούς οικονομικών στοιχείων.

Με την βοήθεια της Java Script μπορούμε να πετύχουμε:

- Πολυμερή έγγραφα με πλαίσια
- Επαναφόρτωση μέρους του παραθύρου
- Δημιουργία εγγραφών με αλληλεπίδραση
- Περισσότερος έλεγχος στην αλληλεπίδραση με το χρήστη
- Έγγραφα με μνήμη
- Ζωντανά έγγραφα
- Μηνύματα που ολισθαίνουν
- Ρολόγια
- Χρονικός μηχανισμός αντίστροφης μέτρησης
- Έγγραφα με αυτόματη ενημέρωση κ.α.

2.3.3 Τρόπος Λειτουργίας της JavaScript

Η JavaScript είναι ενσωματωμένη σ' ένα αρχείο HTML ανάμεσα σε tags SCRIPT. Παρακάτω δίνεται ένα παράδειγμα ενός πολύ απλού προγράμματος:

```
<SCRIPT LANGUAGE = "JavaScript"> document.writeln( "<H3> Hello there"); </SCRIPT>
```

Ο κώδικας βρίσκεται μέσα στα <SCRIPT> </SCRIPT> και αποτελείται από μία απλή δήλωση που εμφανίζει σε μια ιστοσελίδα το μήνυμα 'Hello there' στο σημείο όπου υπάρχει δήλωση μέσα στο HTML αρχείο. Το μήνυμα εμφανίζεται σε επικεφαλίδα μεγέθους 3, χρησιμοποιώντας το κατάλληλο tag. Το αντικείμενο document αντιπροσωπεύει το αρχείο HTML στο οποίο περικλείεται ο κώδικας. Η μέθοδος writeln του αντικειμένου document γράφει ένα string στον HTML κώδικα που θα εμφανιστεί.

Η JavaScript μοιάζει πολύ με μια κανονική γλώσσα προγραμματισμού καθώς περιλαμβάνει:

- Αριθμητικούς τύπους δεδομένων, όπως ακέραιους, καθώς και strings.
- Ένα μεγάλο αριθμό τελεστών που μπορούν να εφαρμοστούν στους διάφορους τύπους δεδομένων.
- Δομές ελέγχου, όπως οι if και while.
- Συναρτήσεις: κομμάτια κώδικα που μπορούν να κληθούν επανειλημμένα και ανεξάρτητα όποτε χρειάζεται να εκτελεστεί η συγκεκριμένη λειτουργία που μέθοδος υλοποιεί.
- Πίνακες που περιέχουν συλλογές δεδομένων.

Αυτό που την ξεχωρίζει πάντως είναι η ικανότητα, χρησιμοποιώντας την τεχνολογία της δυναμικής HTML (DHTML), να αλληλεπιδρά με τα στοιχεία μίας ιστοσελίδας, για παράδειγμα μπορεί να έχει πρόσβαση στα στοιχεία μιας φόρμας HTML ή ν' αλλάξει την εμφάνιση ενός στοιχείου κειμένου.

2.4 JSP(JavaServer Pages)

2.4.1 Τι είναι JSP

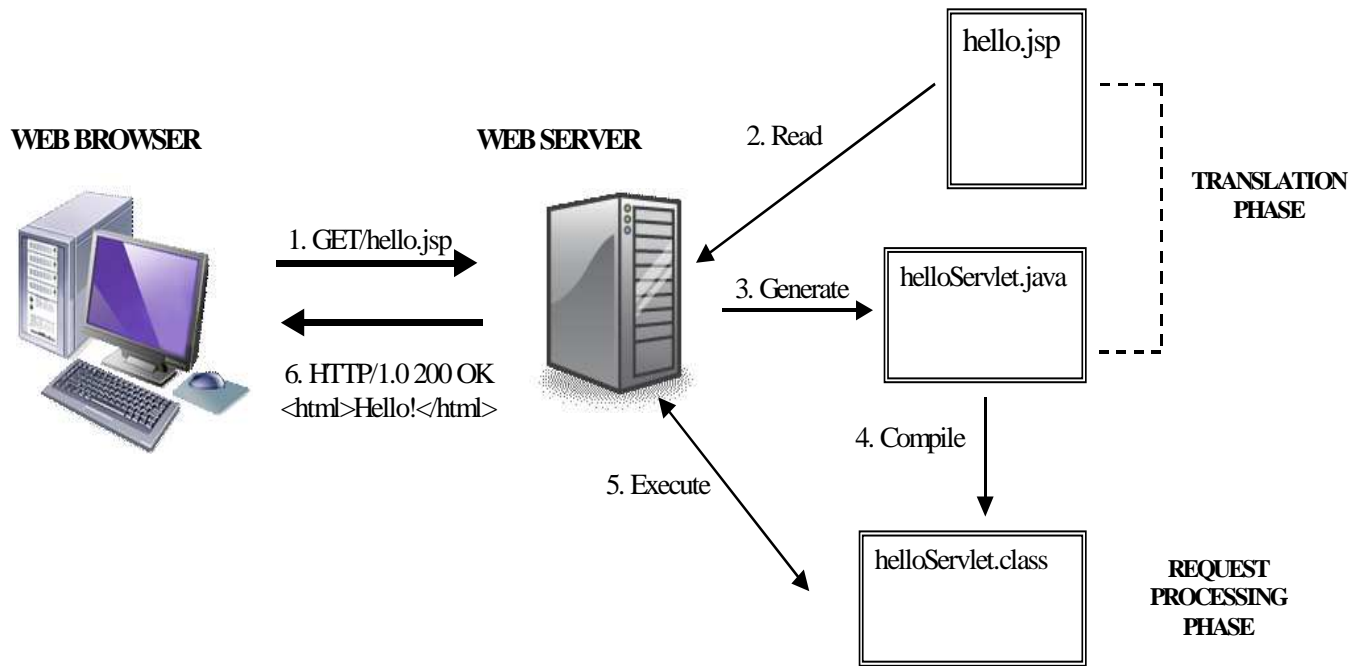
Οι JSPs (JavaServer Pages) είναι μια τεχνολογία που έχει δημιουργηθεί από την εταιρεία Sun Microsystems για να μπορεί να δημιουργεί δυναμικό περιεχόμενο (dynamic content) στο Web. Πρόκειται για HTML έγγραφα (ιστοσελίδες) τα οποία αναμειγνύονται με τη γλώσσα προγραμματισμού Java, η οποία και έχει τη δυνατότητα να παρέχει (δημιουργεί) αυτό το δυναμικό περιεχόμενο. Οι JSPs είναι μια εφαρμογή στην πλευρά του server (server-side application), που σημαίνει ότι δέχονται μια αίτηση (request) και παράγουν μια απόκριση ή απάντηση (response).

Επειδή οι JSPs χρησιμοποιούν τη γλώσσα προγραμματισμού Java, ακολουθούν την πολιτική write-once, run-anywhere. Αυτό σημαίνει ότι μια JSP μπορεί να εκτελεσθεί σ' έναν οποιονδήποτε application server ο οποίος υποστηρίζει τις JSPs χωρίς να χρειασθεί κάποια τροποποίηση στον κώδικα. Οι JSPs μπορούν να γραφούν σ' έναν text editor με την επέκταση (extension) .jsp

2.4.2 Αρχιτεκτονική web εφαρμογής με JSP

Τρία είναι τα κύρια στοιχεία αρχιτεκτονικής μιας εφαρμογής web:

- Servlet – το οποίο είναι υπεύθυνο για την επεξεργασία των requests από τον browser (κέντρο ελέγχου)
- JSP – η οποία είναι υπεύθυνη για την παρουσίαση του αποτελέσματος (presentation) και τέλος
- JavaBean – το οποίο είναι υπεύθυνο για την υλοποίηση της business λογικής της εφαρμογής.



Εικόνα 1 Οι φάσεις επεξεργασίας μιας JSP σελίδας

2.5 HTML

2.5.1 Τι είναι html;

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language (γλώσσα μορφοποίηση υπερκειμένου) και είναι η βασική γλώσσα δόμηση σελίδων του World Wide Web (ή απλά ιστού: Web). Είναι μία γλώσσα προγραμματισμού. Χρησιμοποιείται για να σημαίνει ένα τμήμα κειμένου και να το κάνει να εμφανίζεται καλύτερα. Επιτρέπει την ενσωμάτωση ήχου και εικόνων στις web σελίδες. Αρχικά είχε κατασκευασθεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές κ.α.

Η γλώσσα χρησιμοποιεί ένα αριθμό από tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα των σελίδα, για την εισαγωγή εικόνων, ήχου κ.α. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας.

2.5.2 Εκδόσεις HTML

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 είδος 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 γλώσσα HTML.

Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Force). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρίες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

2.5.3 Γλώσσες Προγραμματισμού στο Internet

Οι ενδογενείς αδυναμίες της HTML, μιας κατεξοχήν γλώσσας μορφοποίησης υπερκειμένου, σύντομα οδήγησαν στην ανάπτυξη γλωσσών προγραμματισμού στο Internet. Η χρήση των γλωσσών προγραμματισμού κρίθηκε απαραίτητη αφού η εξάπλωση του Internet και η χρήση των σελίδων HTML για αλληλεπίδραση με τους χρήστες (δυναμική αποστολή και λήψη δεδομένων) δεν μπορούσε να πραγματοποιηθεί μέσω της HTML. Αρχικά, αναπτύχθηκε από την εταιρεία Netscape η γλώσσα JavaScript η οποία, όπως και η HTML μεταφράζεται από τον Web Browser κατά την εμφάνιση της σελίδας. Η Microsoft ανέπτυξε απ' την πλευρά της μια δική της έκδοση της γλώσσας JavaScript την οποία ονόμασε JScript και μια έκδοση της γλώσσας Basic ειδικά για το Internet που ονόμασε VBScript.

Έτσι πλέον με την χρήση των παραπάνω γλωσσών είναι δυνατό να ελεγχθούν και να προγραμματιστούν όλα σχεδόν τα αντικείμενα που μπορεί να περιέχει μία ιστοσελίδα και με τη χρήση είτε εντολών διαδικαστικού χαρακτήρα, είτε εντολών διακλάδωσης, όπως για παράδειγμα η δομή [if ... then... else...]. Η χρήση τέτοιου είδους "προγραμμάτων" ή όπως ονομάζονται "scripts" είναι διαδικασία που έχει μεγάλες απαιτήσεις σε προγραμματισμό και απευθύνεται σε προγραμματιστές και μόνο.

2.5.4 Δημιουργία αρχείων HTML

Η δημιουργία αρχείων HTML είναι πολύ απλή. Αρκεί να "τρέξουμε" έναν οποιοδήποτε διορθωτή κειμένου text, όπως το Notepad των Windows. Να γράψουμε τον κώδικα HTML που επιθυμούμε και να το αποθηκεύσουμε (σώσουμε) σε ένα αρχείο με κατάληξη .htm ή .html. Συνήθως αποθηκεύουμε με επέκταση .html όταν θα δημοσιεύσουμε τα αρχεία μας σε διακομιστή Unix. Αντίθετα ο εξυπηρετητής Windows NT καταλαβαίνει τα αρχεία και σαν .htm και σαν .html.

2.6 CSS

Τα style sheets δημιουργήθηκαν από τα πρώτα βήματα της SGML το 1970. Αναπτύχθηκαν προκειμένου να παρέχουν πληροφορίες για την απεικόνιση των εγγράφων στο διαδίκτυο. Μετά τη διάδοση της HTML, προστέθηκαν νέες δυνατότητες στις γλώσσες συγγραφής style sheets έτσι ώστε να ικανοποιηθούν οι ανάγκες των προγραμματιστών σε εφαρμογές διαδικτύου. Αυτή η επανάσταση, έδωσε στους σχεδιαστές ιστοσελίδων τον απόλυτο έλεγχο γύρω από θέματα απεικόνισης του ιστότοπου. Παράλληλα όμως η πολυπλοκότητα της HTML γινόταν όλο και μεγαλύτερη.

Ο Robert Cailliau ήταν ο άνθρωπος ο οποίος σκέφτηκε πως αν υπήρχε διαχωρισμός μεταξύ της εμφάνισης και της δομής ενός εγγράφου τότε τα πράγματα θα ήταν πιο απλά. Ο ιδανικός τρόπος για να γίνει αυτό ήταν να δώσει στον χρήστη διαφορετικές επιλογές εμφάνισης ενός εγγράφου και έτσι δημιουργήθηκαν τρία είδη από Style Sheets: ένα για εκτύπωση εγγράφων, ένα για απεικόνιση στην οθόνη και τέλος, ένα για την επεξεργασία του εγγράφου.

Για να βελτιωθεί η απεικόνιση των ιστότοπων στο διαδίκτυο, εννιά διαφορετικές γλώσσες προτάθηκαν από τον οργανισμό W3C (World Wide Web Consortium στις λίστες ηλεκτρονικού ταχυδρομείου (mailing list) που συζητούσαν αυτό το θέμα. Από αυτές τις εννιά γλώσσες, δύο ήταν

αυτές που επιλέχθηκαν από τον οργανισμό: η CHSS (Cascading HTML Style Sheets) και η SSP (Stream-based Style Sheet Proposal) και από αυτές προέρχεται η CSS.

Η CHSS, ήταν μια γλώσσα η οποία έχει μεγάλες ομοιότητες με τη σημερινή μορφή της CSS, και προτάθηκε από τον Hakom Wium Lie τον Οκτώβριο του 1994. Από την άλλη πλευρά η SSP προήλθε από τον Bert Bos ο οποίος είχε κατασκευάσει έναν περιηγητή (browser) ο οποίος για την απεικόνιση των εγγράφων χρησιμοποιούσε την SSP style sheet γλώσσα. Ο Lie και ο Yves Lafon συνεννοήθηκαν με τον Dave Raggett για την επέκταση του περιηγητή Arena ώστε να υποστηρίζει τη CSS. Ο Lie και ο Bos εργαστήκαν μαζί προκειμένου να αναπτύξουν το πρότυπο της CSS.

Σε αντίθεση με πολλές style sheet γλώσσες όπως η DSSSI και η FOSI, η CSS επέτρεψε τη μορφοποίηση ενός εγγράφου από διαφορετικά style sheets. Ένα style sheet μπορεί να επηρεαστεί από άλλο, επιτρέποντας έτσι τη μίξη διαφόρων μορφοποιήσεων τα οποία μπορούσαν να διαχειριστούν ισότιμα από τον σχεδιαστή αλλά και από τον χρήστη.

Η πρόταση του Lie παρουσιάστηκε στο συνέδριο με τίτλο “Mosaic and the Web” στο Σικάγο το 1994. Εκείνη τη χρονική περίοδο μόλις είχε δημιουργηθεί ο οργανισμός W3C και έδειξε ιδιαίτερο ενδιαφέρον για την ανάπτυξη της CSS. Ο ίδιος οργανισμός διοργάνωσε ένα workshop στο οποίο υπεύθυνος ήταν ο Steven Pemberton. Ως αποτέλεσμα, υπήρχε όλο και μεγαλύτερο ενδιαφέρον για την ανάπτυξη της CSS. ο Lie και ο Bos ήταν τα πρώτα μέλη της τεχνικής επιτροπής αυτού του project μαζί με άλλους σημαντικούς επιστήμονες όπως ο Thomas Reardon από τη Microsoft. Τον Αύγουστο του 1996 η εταιρία Netscape Communication Corporation παρουσίασε μια εναλλακτική γλώσσα της CSS, που ονομαζόταν JSSS (Javascript Style Sheets). Οι προδιαγραφές της ποτέ δεν τελείωσαν με αποτέλεσμα να παρωχηθεί. Στα τέλη του 1996 η CSS έγινε επίσημη γλώσσα για μορφοποίηση ιστοσελίδων με την έκδοση CSS level 1 αυτής.

Η ανάπτυξη της HTML, CSS και DOM έγινε από μια ομάδα εργασίας που ονομάζεται ERB (HTML Editorial Review Board). Στη συνέχεια η ομάδα αυτή χωρίστηκε σε τρεις άλλες ομάδες εργασίας:

- (1) το HTML Working group προεδρευόμενο από τον Dan Connolly του οργανισμού W3C,
- (2) το DOM Working group, προεδρευόμενο από τον Lauren Wood της SoftQuad και
- (3) το CSS Working group, προεδρευόμενο από τον Chris Lilley του οργανισμού W3C.

Η ομάδα εργασίας που ήταν υπεύθυνη για την ανάπτυξη της CSS ξεκίνησε να επεξεργάζεται θέματα τα οποία δεν μπορούσαν να γίνουν με τη CSS level 1, με αποτέλεσμα να δημιουργηθεί ένα πρόχειρο της δεύτερης έκδοσης της γλώσσας το Νοέμβριο του 1997. Τελικά η CSS level 2 δημοσιεύθηκε τον Μάιο του 1998 επίσημα. Σήμερα βρίσκεται υπό ανάπτυξη, από το 1998, η τρίτη έκδοση της γλώσσας.

2.6.1 Τι είναι η CSS

Η CSS είναι μια markup γλώσσα η οποία περιγράφει τον τρόπο απεικόνισης εγγράφων που έχουν γραφεί σε markup γλώσσες όπως π.χ. η HTML και η XHTML. Επίσης η CSS μπορεί να χρησιμοποιηθεί σε οποιοδήποτε XML έγγραφο όπως τα SVG και XUL. Με τη γλώσσα αυτή περιγράφεται ο τρόπος εμφάνισης του εγγράφου σε οθόνες, εκτυπωτές κτλπ. Σήμερα η CSS χρησιμοποιείται σε πολύ μεγάλο βαθμό στον σχεδιασμό διαδικτυακών εφαρμογών. Συγκεκριμένα, ο οργανισμός W3C προωθεί τη χρήση style sheets στο διαδίκτυο από το 1994 για στοιχεία όπως είναι η σχεδίαση του layout της ιστοσελίδας, τα χρώματα και το τι γραμματοσειρές θα χρησιμοποιηθούν.

Με την επισύναψη αρχείων CSS σε δομημένα έγγραφα (HTML, XML κτλπ) ο χρήστης ή ο αναγνώστης έχει τη δυνατότητα να τροποποιήσει την παρουσίαση του εγγράφου χωρίς να κάνει αλλαγές στη δομή του, δηλαδή να προσθέσει κάποια ετικέτα κλπ. Άλλος ένας σημαντικός λόγος για τη χρήση της γλώσσας αυτής στο διαδίκτυο είναι ο διαχωρισμός του περιεχομένου από την εμφάνιση του εγγράφου, δίνοντας έτσι ένα πλεονέκτημα σε θέματα που αφορούν την πρόσβαση στο περιεχόμενο, την καλύτερη δόμηση του κώδικα, θέματα βελτιστοποίησης κλπ.

Χρησιμοποιώντας τη γλώσσα CSS στον προγραμματισμό ιστοσελίδων μπορούμε να εφαρμόσουμε την ίδια μορφοποίηση σε πολλές σελίδες, μπορούμε να μειώσουμε την πολυπλοκότητα του HTML κώδικα ορίζοντας τη μορφοποίηση των πινάκων μέσα στο CSS έγγραφο. Επίσης, υπάρχει

η δυνατότητα εμφάνισης του ίδιου εγγράφου με πολλές μορφοποιήσεις. Αυτό είναι ένα πολύ χρήσιμο χαρακτηριστικό καθώς μπορούμε να επιλέξουμε τον τρόπο με τον οποίο το ίδιο έγγραφο μπορεί να απεικονιστεί στην οθόνη ή στον εκτυπωτή ή και με φωνή σε Braille συστήματα.

2.6.2 Πλεονεκτήματα CSS

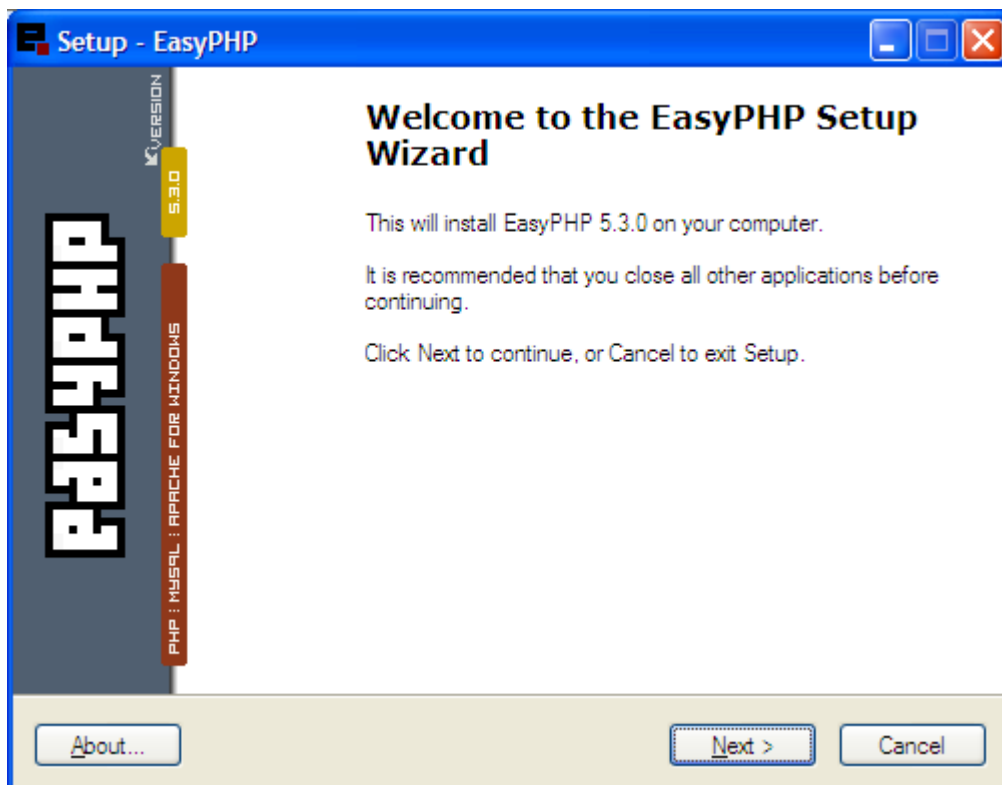
Το κύριο πλεονέκτημα της CSS όπως αναφέρθηκε παραπάνω είναι το ότι μια μορφοποίηση μπορεί να εφαρμοστεί σε ένα σύνολο ιστοσελίδων. Αν απαιτούνται αλλαγές στη μορφοποίηση, τότε με μια επεξεργασία του CSS εγγράφου όλες οι σελίδες του ιστότοπου θα λάβουν την ίδια αλλαγή. Η εναλλακτική αυτού θα ήταν η επεξεργασία κάθε σελίδας χωριστά προκειμένου να γίνει μια αλλαγή. Ως αποτέλεσμα, χρησιμοποιώντας τη CSS και τα υποστηριζόμενα, από τη γλώσσα, style sheets όλη η ευθύνη για τη μορφοποίηση των ιστοσελίδων αναλαμβάνεται από ένα γραφείο (μιας εταιρίας) ή από μόνο ένα άτομο, ακόμη και για πολυεθνικές εταιρίες με δεκάδες ιστότοπους.

2.7 Εγκατάσταση EasyPHP

Η EasyPHP είναι ένα πλήρες πακέτο λογισμικού για τα Windows που επιτρέπει την ανάπτυξη PHP και MySQL στον προσωπικό υπολογιστή του χρήστη. Ξεκίνησε το 1999 και το πακέτο περιλαμβάνει ένα διακομιστή Apache, μια βάση δεδομένων MySQL, την επέκταση της PHP καθώς και εργαλεία εύκολης ανάπτυξης για το web site ή τις εφαρμογές μας.

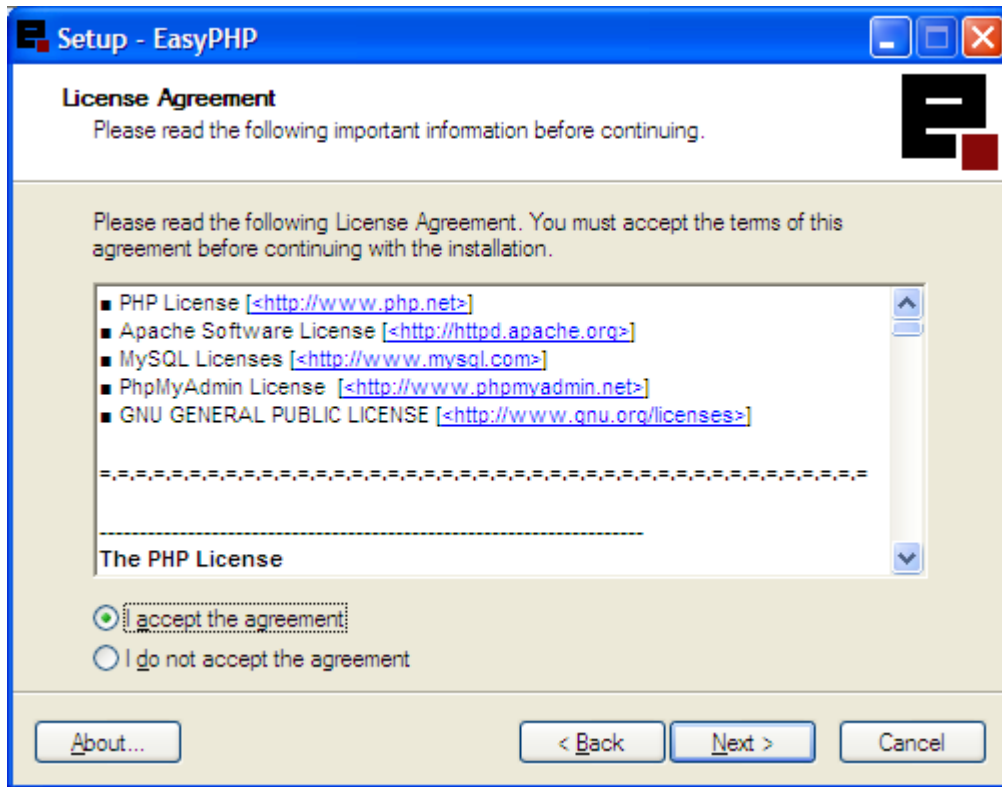
Τρέχουμε το αρχείο easyphp 5.3.0 το οποίο έχουμε κατεβάσει από το site <http://www.easyphp.org/download.php>.

Στο παράθυρο που εμφανίζεται πατάμε το κουμπί next.



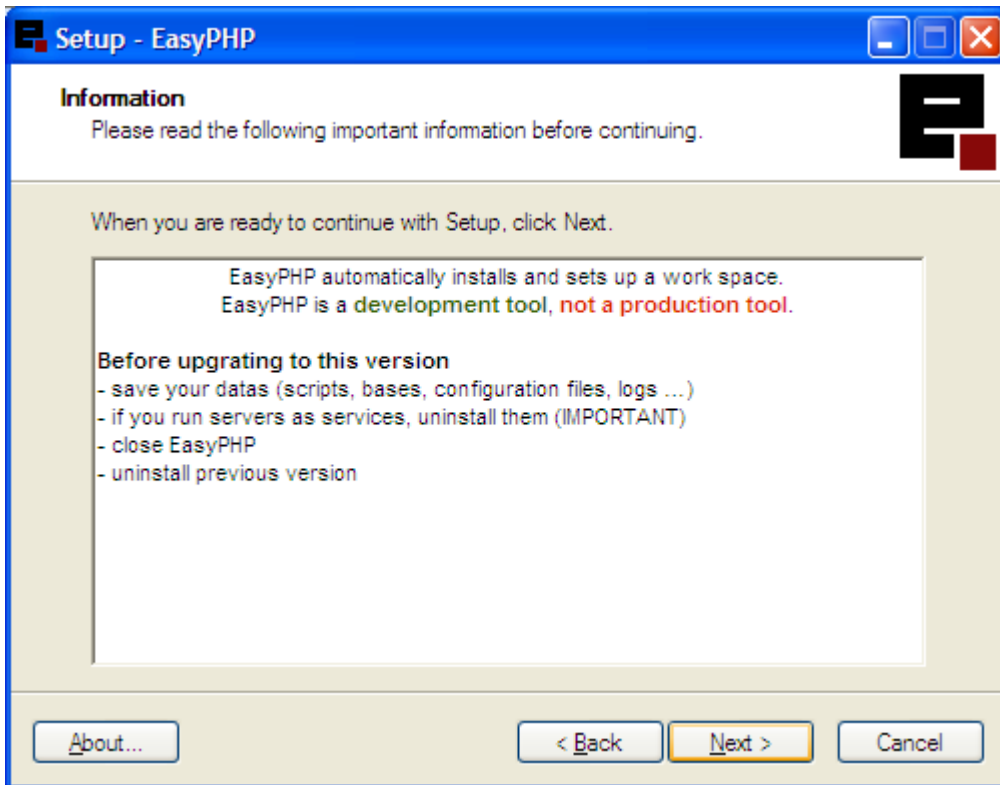
Εικόνα 2 Εγκατάσταση EasyPHP

Έπειτα επιλέγουμε την επιλογή “I accept the agreement”, που σημαίνει ότι αποδεχόμαστε τους όρους χρήσης και πατάμε next.



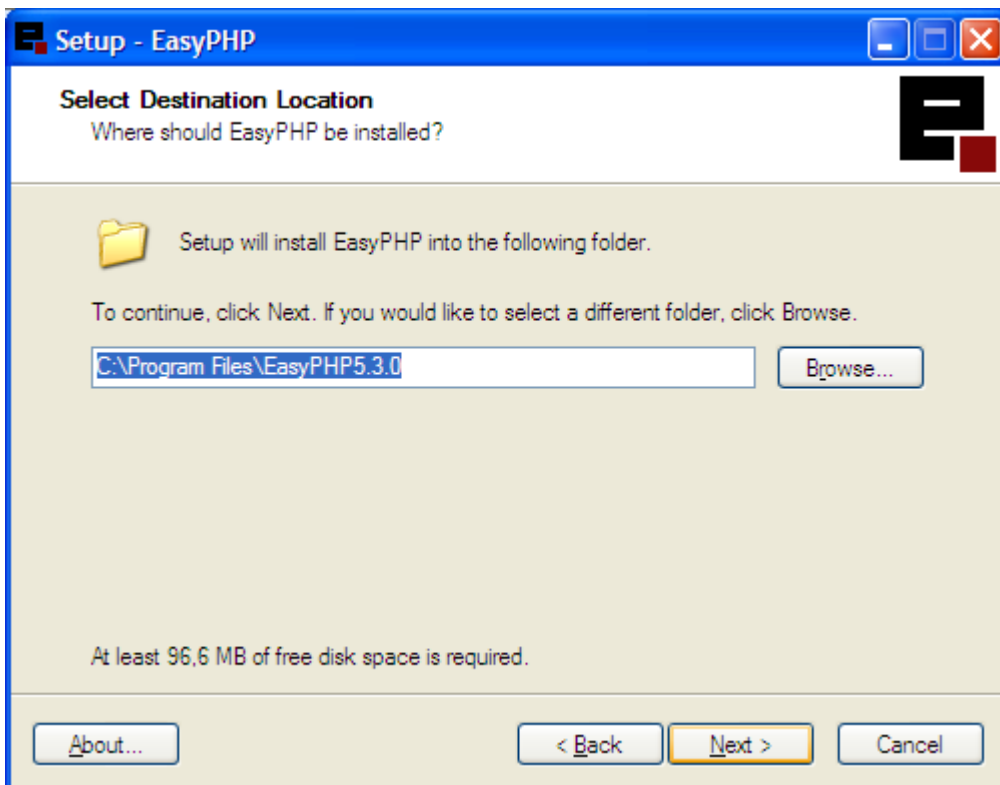
Εικόνα 3 Επιλογή αποδοχής των όρων χρήσης

Πατάμε next και στο επόμενο παράθυρο, το οποίο μας ενημερώνει ότι πρέπει να απεγκαταστήσουμε τυχόν άλλους server πριν την εγκατάσταση.



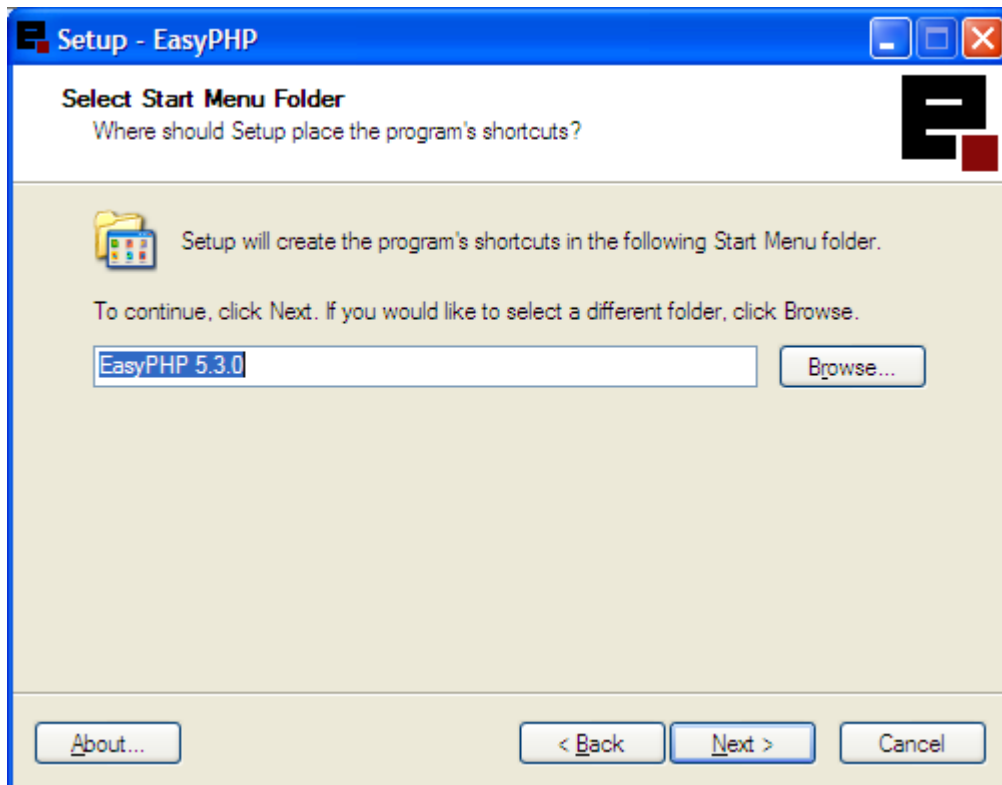
Εικόνα 4 Επιλογή next στις πληροφορίες

Για να συνεχίσουμε πατάμε next, αφήνοντας τη διαδρομή που έχει επιλέξει η εφαρμογή.



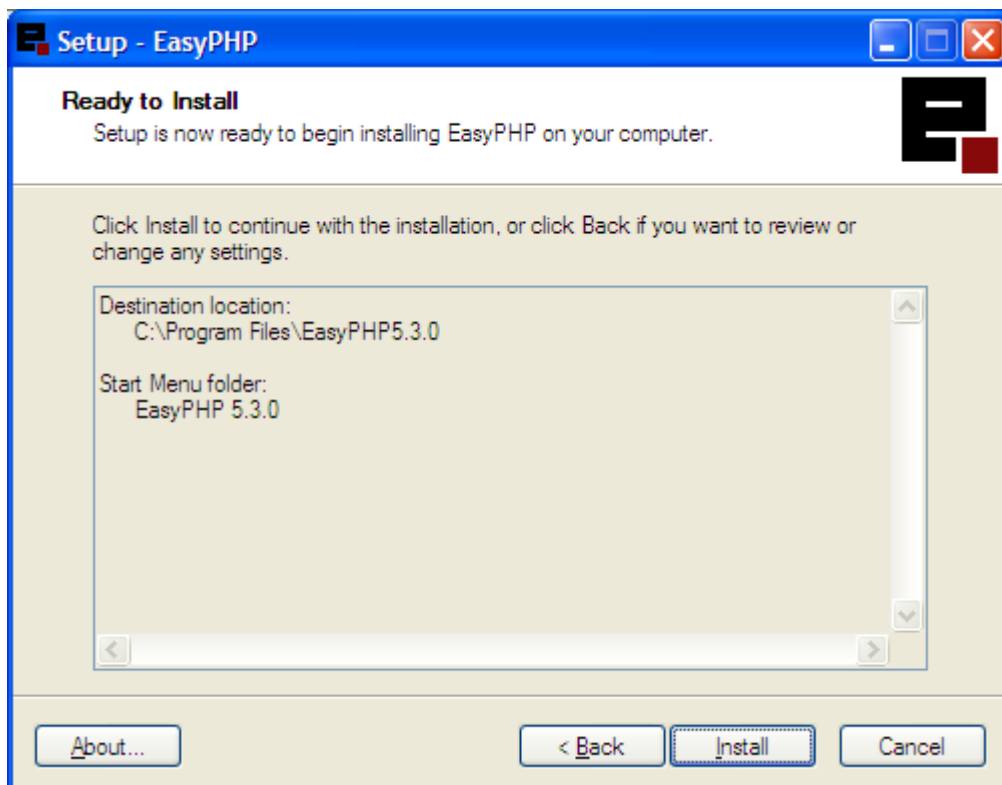
Εικόνα 5 Επιλογή διαδρομής για την εγκατάσταση

Πατάμε ξανά next και αφήνουμε το όνομα του φακέλου που θα δημιουργηθεί ως έχει.



Εικόνα 6 Επιλογή ονόματος φακέλου

Στη συνέχεια πατάμε install.



Εικόνα 7 Επιλογή Install

Τέλος πατάμε το κουμπί finish και αφήνουμε τσεκαρισμένες τις επιλογές "Open Help" και "Launch EasyPHP", τα οποία θα φορτώσουν την εφαρμογή μας και θα ανοίξουν ένα παράθυρο με πληροφορίες που θα μας βοηθήσουν.



Εικόνα 8 Επιλογή Finish

Εφόσον έχουμε ολοκληρώσει την εγκατάσταση ανοίγουμε την EasyPHP.



Εικόνα 9 EasyPHP

Πατώντας δεξί κλικ στο εικονίδιο της EasyPHP που θα εμφανιστεί στην κάτω δεξιά μπάρα και μετά administrator ανοίγουμε την αρχική σελίδα της EasyPHP. Το ίδιο μπορεί να γίνει πληκτρολογώντας τη διεύθυνση <http://127.0.0.1/home/>.



Εικόνα 10 Αρχική σελίδα της EasyP

2.8 Macromedia Dreamweaver

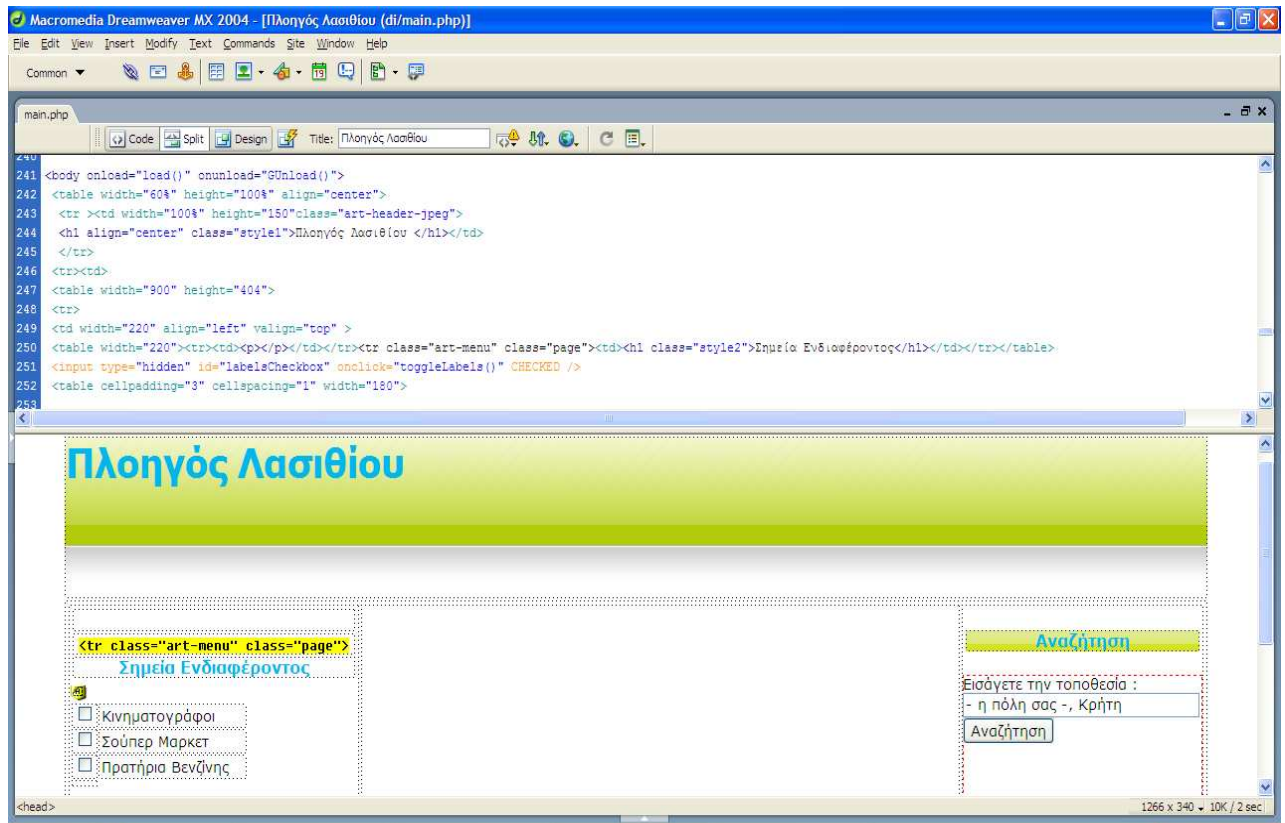


Το πρόγραμμα DreamWeaver της εταιρείας Macromedia είναι ένα κορυφαίο πρόγραμμα δημιουργίας και επεξεργασίας ιστοσελίδων, δηλαδή κώδικα HTML, που είναι ιδιαίτερα εύκολο και φιλικό στη χρήση του. Το DreamWeaver είναι εξαιρετικό για να μπορούμε να δημιουργήσουμε στα γρήγορα φόρμες (forms), πλαίσια (frames), πίνακες (tables) και άλλα αντικείμενα της HTML. Είναι, όμως, ιδιαίτερα καλό όταν θέλουμε να δώσουμε σε μια ιστοσελίδα τη δυνατότητα να κάνει κάτι. Πρέπει να έχουμε υπόψη μας ότι το DreamWeaver μπορεί να χρησιμοποιηθεί και για τη δημιουργία εφαρμογών πολυμέσων.

Το DreamWeaver έχει δυνατότητες για δημιουργία δυναμικής HTML (DHMTL) σελίδας και επιτρέπει την κίνηση στη γραμμή του χρόνου, απόλυτη τοποθέτηση περιεχομένων, δημιουργία επιπέδων (layers) και συγγραφή σεναρίων (scripts). Το DreamWeaver περιέχει δικές του συμπεριφορές (behaviors), που είναι έτοιμα scripts τα οποία μπορούμε να προσθέσουμε πολύ εύκολα σ' ένα αντικείμενο.

Το DreamWeaver μάς παρέχει την ελευθερία να σχεδιάσουμε οπτικά την εμφάνιση μιας ιστοσελίδας και τη δύναμη να την κάνουμε να λειτουργεί όπως ακριβώς θέλουμε. Υποστηρίζει πολλές διαφορετικές γλώσσες προγραμματισμού (PHP, ASP, JavaScript, CSS) και μπορούμε να δημιουργήσουμε τη δική μας προσωπική ιστοσελίδα (personal web page) ή μια ολόκληρη περιοχή (web site) σ' ένα εταιρικό δίκτυο (intranet).

Στην παρούσα εργασία χρησιμοποιήθηκε η έκδοση MX 2004 με την οποία δημιουργήθηκε η ιστοσελίδα μας.



Εικόνα 11 Dreamweaver

3 Γεωγραφικά Συστήματα Πληροφοριών (GIS)

3.1 Εισαγωγή

Τα Γεωγραφικά Συστήματα Πληροφοριών (Geographical Information Systems – G.I.S.) είναι λογισμικά που μαζί με τον κατάλληλο εξοπλισμό και το εξειδικευμένο προσωπικό αξιοποιούν τις δυνατότητες των υπολογιστών για αποθήκευση, ανάλυση και απόδοση των δεδομένων που συνδέονται άμεσα ή έμμεσα με τη γεωγραφική κατανομή.

Τα Γεωγραφικά Συστήματα Πληροφοριών (ΓΠΣ) θα πρέπει να θεωρηθούν κάτι παραπάνω από μέσα κωδικοποίησης, αποθήκευσης και ανάκτησης σχετικών δεδομένων με τις ιδιότητες της γήινης επιφάνειας. Τα δεδομένα ενός G.I.S, είτε είναι κωδικοποιημένα στην επιφάνεια ενός χαρτιού, είτε είναι άορα σημεία στην επιφάνεια μιας μαγνητικής ταινίας, θεωρείται ότι στοιχειοθετούν ένα μοντέλο του αληθινού κόσμου. Επειδή σε αυτά τα δεδομένα ο χρήστης έχει πρόσβαση, τα μετατρέπει και τα διαχειρίζεται ταυτόχρονα, τα G.I.S. μπορούν να χρησιμεύσουν σαν ένα δοκιμαστικό μοντέλο για τη μελέτη περιβαλλοντικών διαδικασιών, την ανάλυση των αποτελεσμάτων των τάσεων, ή ακόμα και για τη μελέτη των πιθανών συνεπειών ενός σχεδιασμού. Για παράδειγμα, στην περιβαλλοντική μελέτη, χρησιμοποιώντας τα G.I.S, είναι δυνατή η εξερεύνηση μίας σειράς πιθανών σεναρίων και η κατανόηση των επιπτώσεων πιθανών ενεργειών, πριν αυτές πραγματοποιηθούν και δημιουργήσουν ανεπανόρθωτες βλάβες στο περιβάλλον.

Γενικά τα Γεωγραφικά Συστήματα Πληροφοριών αποτελούν δυναμικά συστήματα πληροφοριών, τα οποία με την τεχνολογία της Πληροφορικής και τις ειδικές μαθηματικές μεθόδους διαχειρίζονται και αξιοποιούν δεδομένα από τις Γεωεπιστήμες για την παραγωγή μεταπληροφορίας, δηλαδή πληροφορίας υψηλότερου επιπέδου.

3.2 Ιστορία των GIS

Η ανάγκη του ανθρώπου για συστηματική ταξινόμηση των ιδιαίτερων στοιχείων της γήινης επιφάνειας σχετικά με τη χωρική κατανομή, καθώς και η ζήτηση χαρτών εξειδικευμένων θεμάτων που αφορούσαν στη γήινη επιφάνεια, ήταν η αιτία που οδήγησε στην κατασκευή των πρώτων χαρτών. Οι χάρτες αυτοί αποτέλεσαν την πρώτη μορφή G.I.S.

Η ανάπτυξη μεθόδων λήψης και ανάλυσης αεροφωτογραφιών και εικόνων τηλεανίχνευσης είχαν ως αποτέλεσμα τη χαρτογράφηση με μεγαλύτερη ακρίβεια από ό,τι τα προηγούμενα χρόνια. Οι ίδιες μέθοδοι ήταν αυτές που έδωσαν στους επιστήμονες τεράστιες δυνατότητες όχι απλώς για έρευνα, αλλά και για σημαντική αύξηση της ακρίβειας των αποτελεσμάτων που προέκυπταν από αυτή.

Η πρώτη προσπάθεια για συστηματική χρησιμοποίηση των χαρτογραφικών δεδομένων έγινε κατά τη διάρκεια των δεκαετιών του '60 και του '70. Ιδιαίτερα, οι σχεδιαστές και οι αρχιτέκτονες στις Η.Π.Α. συνειδητοποίησαν ότι τα δεδομένα, που προέρχονται από διαφορετικές πρωτογενείς έρευνες, μπορούν να συνδυαστούν και να ενοποιηθούν επικαλύπτοντας διαφανή αντίγραφα χαρτών σε μία φωτεινή τράπεζα. Ο πιο γνωστός υποστηρικτής της απλής αυτής τεχνικής ήταν ο Αμερικανός αρχιτέκτονας Ian McHarg. Η πρώτη οργανωμένη προσπάθεια χρησιμοποίησης των χαρτογραφικών δεδομένων από τον ηλεκτρονικό υπολογιστή έγινε το 1963 από τον Howard T. Fisher. Το πρόγραμμα του Fisher ονομάστηκε SYMAP (Synagraphic MAPping system) και δημιουργούσε απλούς χάρτες τυπώνοντας στατιστικές τιμές πάνω σε έναν κানাβο, ενώ τα αποτελέσματα προβάλλονταν με πολλούς τρόπους, χρησιμοποιώντας διαδοχικές γραμμικές εκτυπώσεις για την παραγωγή κατάλληλων αποχρώσεων του γκρι. Το πρόγραμμα SYMAP ακολούθησε μία σειρά άλλων προγραμμάτων χαρτογράφησης, όπως το GRID και το IMGRID που είχαν τη δυνατότητα να χρωματίζουν και να σκιαγραφούν επιφάνειες με αποτέλεσμα να επιτυγχάνεται με τη χρήση των ηλεκτρονικών υπολογιστών ό,τι ο McHarg με τις διαφανείς επικαλύψεις. Από τότε μία σειρά εξελίξεων όχι μόνο στα προγράμματα αυτά αλλά και στην τεχνολογία των υπολογιστών ως μηχανήματα, είχαν ως αποτέλεσμα

τη δημιουργία των νέων συστημάτων που χειρίζονται, αναλύουν και παρουσιάζουν πληροφορίες από το γεωγραφικό χώρο. Για το λόγο αυτό ονομάστηκαν Γεωγραφικά Συστήματα Πληροφοριών (G.I.S.), ή Συστήματα Πληροφοριών Γης (LIS) και χρησιμοποιήθηκαν από ένα ευρύ κοινό επιστημόνων ποικίλων ειδικοτήτων που συνεχώς αυξάνεται.

Τα Γεωγραφικά Συστήματα Πληροφοριών πέρασαν από διάφορα εξελικτικά στάδια, περίπου αντίστοιχα με τις γενιές των Η/Υ. Στην αρχή, χρησιμοποιούνταν οι υπολογιστές και οι γραφικές τους ικανότητες, για να γίνουν εργασίες που μέχρι τότε γίνονταν χειρονακτικά. Στη συνέχεια άρχισε η ενσωμάτωση των μαθηματικών και στατιστικών μεθοδολογιών στις χαρτογραφικές τεχνικές, η εξέλιξη των οποίων σε συνδυασμό με την εξέλιξη των Η/Υ και των συστημάτων τηλεανίχνευσης έδωσε τα σημερινά ισχυρά Γ.Σ.Π.

3.3 Χαρακτηριστικά των GIS

Τα G.I.S. οφείλουν να είναι ικανά να συσχετίζουν αντικείμενα με ένα πλαίσιο, να ανταπεξέρχονται σε μη γραφικές ιδιότητες (διαχείριση αριθμητικών μοντέλων) και να μπορούν να περιγράφουν τοπολογικές σχέσεις.

Η μετατροπή και η τροποποίηση των δεδομένων είναι ένα βασικό χαρακτηριστικό του Software που απαιτείται για την άνετη εργασία σε ένα Γεωγραφικό Σύστημα Πληροφοριών. Το G.I.S. θα πρέπει να έχει κάποια βασικά χαρακτηριστικά, έτσι ώστε να είναι εφικτή η αφαίρεση τυχόν σφαλμάτων, η ενημέρωση του αρχείου για την προσθήκη νέων δεδομένων, ο συσχετισμός δεδομένων με άλλες ομάδες για την εξαγωγή συμπερασμάτων νέας μορφής και η χρησιμοποίηση των υπαρχόντων δεδομένων σε άλλες εργασίες διαφορετικές από την αρχική.

Σημαντικό χαρακτηριστικό των Γεωγραφικών Συστημάτων Πληροφοριών είναι η ευρεία χρησιμοποίηση των επιπέδων πληροφορίας (layers). Η εισαγωγή δεδομένων γίνεται με τη χρήση των layers τα οποία μπορεί να παρομοιαστούν με διαφανείς σελίδες, οι οποίες περιέχουν μία καθορισμένη από το χρήστη ομάδα πληροφοριών. Σε ένα layer είναι δυνατόν να απεικονιστούν όλα τα είδη των στοιχείων, σημεία, γραμμές, επιφάνειες και εικόνες. Για την τοποθέτηση ενός στοιχείου στο layer, δηλώνεται το γεωγραφικό μήκος και πλάτος, το είδος του συμβολισμού που επιθυμεί ο χρήστης και οι πληροφορίες (ονόματα, γράμματα, αριθμοί, κ.λπ.) που συνοδεύουν την καταχώριση. Με τον τρόπο αυτόν, είναι δυνατή η παρουσίαση όποιων ομάδων πληροφοριών απαιτούνται για τη μελέτη που πραγματοποιείται, χωρίς να γίνεται σύγχυση των πληροφοριών. Κατά κανόνα, τα layers τοποθετούνται με καθορισμένη σειρά. Ως κατώτερο επίπεδο ορίζουμε αυτό που αποτελείται από επιφάνειες (αν υπάρχει και δεύτερο όμοιο, τότε θέτουμε ως κατώτερο αυτό με τις μεγαλύτερες επιφάνειες), ακολουθούν τα επίπεδα με τα γραμμικά δεδομένα (συγκοινωνιακό δίκτυο, ποτάμια, ρήγματα, κ.λπ.) και ως ανώτερο επίπεδο τοποθετείται αυτό με τα σημειακά στοιχεία (πόλεις, επίκεντρα σεισμών, σημεία δειγματοληψιών, κ.λπ.). Πρέπει να σημειωθεί, επίσης, ότι ακόμα και τα σημειακά δεδομένα, όταν έχουν μεγάλη πυκνότητα, μπορεί να σχηματίσουν ένα επίπεδο (κυρίως, όταν γίνει σμίκρυνση του χάρτη, οπότε πλησιάζει το ένα σημείο στο άλλο) και να καλύψουν τις υποκείμενες πληροφορίες. Για την αποφυγή αυτού, χρησιμοποιείται η λειτουργία zoom range, με την οποία καθορίζονται τα όρια μεγέθυνσης και σμίκρυνσης στα οποία επιθυμεί ο χρήστης να είναι ορατό το κάθε επίπεδο πληροφορίας, έτσι ώστε να βελτιστοποιεί το οπτικό αποτέλεσμα.

Η δυνατότητα μεγέθυνσης και σμίκρυνσης (zoom in και zoom out αντίστοιχα) επιτρέπει τη μεταβολή της απεικόνισης κατά βούληση, βλέποντας αντίστοιχα μικρότερη ή μεγαλύτερη περιοχή στην οθόνη του υπολογιστή. Η δυνατότητα αυτή δε βοηθά μόνο στην καλύτερη παρουσίαση και επεξεργασία των δεδομένων, αλλά είναι καθοριστικής σημασίας για την ακρίβεια των ψηφιοποιημένων δεδομένων. Έτσι, κατά την παρατήρηση ενός ψηφιακού χάρτη, χρησιμοποιώντας τη λειτουργία αυτή, είναι δυνατή η εμφάνιση λεπτομερειών που δεν είναι δυνατό να φανούν σε μικρότερη κλίμακα. Φυσικά, η μέθοδος αυτή (zoom in/out) μειώνει σημαντικά τα πιθανά σφάλματα ακρίβειας. Εξάλλου, τροποποιώντας το zoom επηρεάζονται όλα τα layers, διατηρώντας έτσι τη σχέση των πληροφοριών μεταξύ των επιπέδων. Επίσης, υπάρχει η δυνατότητα τροποποίησης των χαρακτηριστικών ολόκληρου του layer, ή μεμονωμένων στοιχείων αυτού. Τέλος, ένα λογισμικό G.I.S. θα πρέπει να είναι κατάλληλο για την εφαρμογή ποικίλων μεθόδων επεξεργασίας στα καταχωρισμένα

δεδομένα, έτσι ώστε να δύναται να απαντά σε πολύπλοκες και συνδυαστικές ερωτήσεις που θα τίθενται από το χρήστη, η φύση των οποίων δεν είναι πάντα γνωστή εκ των προτέρων. Οι βασικότερες ερωτήσεις που μπορούν να τεθούν σε ένα G.I.S. είναι οι ακόλουθες :

α) Ερωτήσεις που αναφέρονται σε τοποθεσίες , αποστάσεις και ποσότητες. Εδώ, περιλαμβάνεται κάθε ερώτηση που αφορά στη θέση ενός αντικειμένου, προσδιοριζόμενη με ποικιλία τρόπων όπως με γεωγραφικές συντεταγμένες, με το όνομα της περιοχής στην οποία βρίσκεται κτλ, στην επιλογή σημείου το οποίο το πρόγραμμα θα προσδιορίζει, καθώς και στη μέτρηση αποστάσεων μεταξύ σημείων που ορίζονται από το χρήστη με τις μονάδες που έχουν επιλεγεί. Επίσης, σε αυτή την ομάδα ερωτήσεων κατατάσσονται και αυτές που αφορούν στο μέγεθος μιας περιοχής, ή ακόμη και μιας γραμμής, στην εμφάνιση μιας συγκεκριμένης μορφής από το σημείο Α στο Β, καθώς και τα αντικείμενα που βρίσκονται πλησίον σημείων με συγκεκριμένους συνδυασμούς χαρακτηριστικών γνωρισμάτων.

β) Ερωτήσεις που προσδιορίζουν καταστάσεις ελεγχόμενες από ορισμένες μεταβλητές. Έτσι για παράδειγμα, θα πρέπει ένα G.I.S. έχοντας βέβαια τα κατάλληλα δεδομένα, να μπορεί να δώσει πληροφορίες για τις αλλαγές που έχουν προκύψει με την πάροδο του χρόνου σε μία κατάσταση, περιοχή κ.λπ. Τέλος, σε αυτή την κατηγορία ερωτήσεων υπάγονται και αυτές που έχουν τη μορφή "τί θα συμβεί, αν...", όπως και "τί θα συμβεί στις γύρω περιοχές σε ακτίνα 50 Km, αν ανέλθει η στάθμη του ποταμού κατά x μέτρα".

3.4 Πλεονεκτήματα

Σε σχέση με τις παραδοσιακές μεθόδους συλλογής, καταχώρισης, οργάνωσης και επεξεργασίας των δεδομένων, τα G.I.S. παρουσιάζουν τα ακόλουθα πλεονεκτήματα :

- Είναι ταχύτερη η δημιουργία των χαρτών.
- Είναι φθηνότερη η δημιουργία και αναπαραγωγή των χαρτών, ακόμα και σε περιορισμένο αριθμό αντιγράφων.
- Είναι επιτρεπτός ο πειραματισμός σε γραφικές παρουσιάσεις των ίδιων δεδομένων.
- Είναι δυνατός ο συνδυασμός διαφορετικών επιπέδων πληροφορίας.
- Περιορίζεται η χρήση του τυπωμένου χάρτη και των αναλογικών δεδομένων ως αρχεία, διατηρώντας μεγάλη ποσότητα δεδομένων σε ψηφιακή μορφή έτοιμα να χρησιμοποιηθούν κάθε στιγμή.
- Διευκολύνονται οι αναλύσεις των δεδομένων που απαιτούν αλληλεπίδραση μεταξύ των στατιστικών αναλύσεων και της χαρτογράφησης.
- Παρέχεται η δυνατότητα δημιουργίας χαρτών που είναι δύσκολο, αν όχι αδύνατο, να δημιουργηθούν με το χέρι, όπως είναι οι τρισδιάστατοι ή οι στερεοσκοπικοί χάρτες.
- Ανά πάσα στιγμή, είναι δυνατή η επιδιόρθωση ή η προσθήκη δεδομένων.
- Εξυπηρετούν τις ειδικές ανάγκες των διάφορων χρηστών που έχουν πολλές φορές εντελώς διαφορετικές απαιτήσεις για το είδος και την ακρίβεια των πληροφοριών που χρειάζονται.
- Μειώνουν σημαντικά την ποσότητα των πολλαπλά αποθηκευμένων στοιχείων. Έτσι, οι καταγραφές που περιέχουν ένα μεγάλο αριθμό όμοιων στοιχείων, μπορούν να ενωθούν και να αποτελέσουν ένα μόνο αρχείο, έτσι ώστε κάθε πληροφορία να αποθηκεύεται μία μόνο φορά.
- Αποκλείουν την ύπαρξη ασυμφωνιών που προκύπτουν όταν η ίδια πληροφορία καταχωρίζεται με διαφορετικές τιμές σε διαφορετικές υπηρεσίες.
- Κατατάσσουν τα δεδομένα με τις ίδιες προδιαγραφές συγκέντρωσης και αποθήκευσης.
- Παρέχουν στοιχεία μεγάλης ακρίβειας, λόγω του ψηφιακού τρόπου αποθήκευσής τους.
- Επιτρέπουν τη συνεχή πληροφόρηση και ερμηνεία των στοιχείων σε συνδυασμό με τη συνεχή ενημέρωση, ανταλλαγή πληροφοριών και ευκολότερη διανομή στους χρήστες.
- Παρέχουν ασφάλεια και προστασία των στοιχείων από διάφορες φυσικές καταστροφές, καθώς και εύκολη αναπαραγωγή τους.
- Παρέχουν ασφάλεια σε ό,τι αφορά στην τροποποίηση δεδομένων από μη εξουσιοδοτημένα άτομα.

4 Google Maps API

4.1 Διεπαφή Προγραμματισμού Εφαρμογών Google Maps

Το Google Maps API είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη σε ιστοσελίδες με τη χρήση Javascript. Το API παρέχει έναν αριθμό εργαλείων για το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών, όπως και στην επίσημη ιστοσελίδα του Google Maps (<http://maps.google.com/>), επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες.

Το Google Maps API είναι μια δωρεάν υπηρεσία, διαθέσιμη για οποιοδήποτε ιστότοπο που είναι ελεύθερος για το κοινό. Το μόνο που χρειάζεται είναι ένα κλειδί API (API key) που μπορεί να ζητηθεί εφόσον υπάρχει λογαριασμός στη Google και αυτομάτως συνδέεται με το λογαριασμό. Κάθε κλειδί είναι έγκυρο μόνο για ένα συγκεκριμένο όνομα τομέα.

Βασικό στοιχείο οποιασδήποτε Google Maps API εφαρμογής είναι ο ίδιος ο χάρτης, ο οποίος φορτώνεται και αρχικοποιείται με συγκεκριμένες διαστάσεις (ύψος και πλάτος), κέντρο (γεωγραφικό πλάτος, γεωγραφικό μήκος), επίπεδο εστίασης και τύπο (κανονικό, δορυφορικό, υβριδικό). Στη συνέχεια είναι δυνατή η αλληλεπίδραση με το χάρτη μέσω διαφόρων μηχανισμών όπως είναι τα events, τα controls, τα overlays αλλά και συναρτήσεις επεξεργασίας των χαρακτηριστικών του.

4.1.1 Events

Το Google Maps API εμπεριέχει δηλώσεις γεγονότων (events) τα οποία εμφανίζονται μέσω των φυλλομετρητών ιστοσελίδων (web browsers) πάνω σε αντικείμενα του Google Maps. Ας σημειωθεί ωστόσο ότι τα γεγονότα αυτά είναι διαφορετικά από τα standard γεγονότα πάνω στο Μοντέλο Αντικειμενικού Εγγράφου (Document Object Model - DOM), το οποίο είναι μια προγραμματιστική διεπαφή που επιτρέπει σε προγράμματα και εκτελέσιμα σενάρια δράσης (scripts) την ενημέρωση του περιεχομένου και του στυλ όλου του εγγράφου με δυναμικό τρόπο. Έτσι το Google Maps API δημιουργεί και διαχειρίζεται δικά του γεγονότα ενώ το Μοντέλο Αντικειμένου Εγγράφου δημιουργεί άλλα γεγονότα και τα επεξεργάζεται σύμφωνα με το μοντέλο αντικειμένου που χρησιμοποιεί ο συγκεκριμένος φυλλομετρητής. Επιπλέον, το Google Maps API έχει το πλεονέκτημα ότι προσφέρει μηχανισμούς που ακούνε (listen) και απαντάνε (respond) στα DOM γεγονότα ανεξάρτητα από τις ιδιαιτερότητες του κάθε φυλλομετρητή. Παράδειγμα γεγονότων του Google Maps API είναι το κλικ του ποντικιού, το διπλό κλικ, η κίνηση του ποντικιού τα οποία σηματοδοτούν την έναρξη κάποιων διαδικασιών που περιγράφονται στον κώδικα Javascript. Εξάλλου η Javascript είναι μια scripting γλώσσα που “καθοδηγεί γεγονότα” (“event driven”) εξελισσόμενα στους φυλλομετρητές.

4.1.2 Controls

Υπάρχουν διάφορα στοιχεία στη διεπαφή με το χρήστη (user interface) που επιτρέπουν την αλληλεπίδραση του χρήστη με το χάρτη. Τα στοιχεία αυτά είναι γνωστά ως ρυθμιστές (controls) και μπορούν να περιληφθούν στις Google Maps εφαρμογές σε οποιοδήποτε σημείο πάνω στο χάρτη. Τα κυριότερα είναι:

- `GLatLngControl3D`: Τοποθετεί ένα μεγάλο τρισδιάστατο κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.

- **GLargeMapControl:** Τοποθετεί ένα μεγάλο απλό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallMapControl:** Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl3D:** Τοποθετεί ένα μικρό τρισδιάστατο κουμπί ελέγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl:** Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη που χρησιμοποιείται στα μικρά παράθυρα που εμφανίζονται στο χάρτη Google για επεξήγηση των οδικών οδηγιών.
- **GScaleControl:** Τοποθετεί μια κλίμακα με στάδια μεγέθυνσης του χάρτη.
- **GMapTypeControl:** Τοποθετεί κουμπιά που επιτρέπουν στο χρήστη να διαλέγει μεταξύ τύπων του χάρτη (όπως δορυφορικός, εδαφικός κ.ά.).
- **GHierarchicalMapTypeControl:** Τοποθετεί κουμπιά επιλογών (checkboxes) για ιεραρχική προτίμηση των τύπων του χάρτη.
- **GOverviewMapControl:** Τοποθετεί μια πτυσσόμενη επισκόπηση του χάρτη στη γωνία της οθόνης.
- **GNavLabelControl:** Τοποθετεί μια δυναμική ετικέτα που επισημαίνει τη «διεύθυνση» της συγκεκριμένης περιοχής χάρτη ανάλογα με το επίπεδο εστίασης.
Επιπλέον υπάρχει η δυνατότητα δημιουργίας ειδικών ρυθμιστικών στοιχείων από τον προγραμματιστή.

4.1.3 Overlays

Τα overlays είναι αντικείμενα πάνω στο χάρτη που αντιστοιχούν σε συγκεκριμένες συντεταγμένες και γι αυτό μετακινούνται ανάλογα όταν μετακινείται ή αλλάζει εστίαση ο χάρτης. Χρησιμοποιούνται για το σχεδιασμό και την επίδειξη σημείων, γραμμών ή ολόκληρων περιοχών. Το Google Maps API διαθέτει πολλά είδη overlays, τα οποία αναλύονται παρακάτω:

- **Σημεία**

Αναπαριστώνται με τη χρήση δεικτών (markers) πάνω στο χάρτη και συχνά προβάλλουν μια σχετική εικόνα στο σημείο που επιδεικνύουν. Τα markers είναι σχεδιασμένα να είναι διαδραστικά, δηλαδή επάνω τους πραγματοποιούνται γεγονότα, όπως ένα κλικ ή δεξί κλικ, και συχνά προβάλλεται παράλληλα με αυτά ένα παράθυρο πληροφοριών για το σημείο. Μια επιπλέον ιδιότητα τους είναι ότι μπορεί να είναι συρόμενοι (draggable) από ένα σημείο σε άλλο στο χάρτη. Επίσης δυνατή είναι η επεξεργασία και αλλαγή της εικόνας (icon) που αντιστοιχεί σε κάθε δείκτη. Η εικόνα αποτελείται από μια εικόνα στο παρασκήνιο και μια άλλη ως σκιά τοποθετημένες σε συγκεκριμένες θέσεις μεταξύ τους. Παρότι οι δείκτες έχουν μεγάλη χρησιμότητα, η προσθήκη πολλών από αυτούς σε έναν χάρτη Google μπορεί να έχει αρνητική επίδραση στην ταχύτητα απεικόνισης του χάρτη (map rendering) ειδικά σε συγκεκριμένα επίπεδα εστίασης. Λύση σε αυτό δίνει ο Διαχειριστής Δεικτών (Marker Manager), ο οποίος έχει την ικανότητα να παρακολουθεί ποιοι δείκτες είναι ορατοί σε ποια επίπεδα εστίασης και να στέλνει μόνο τους απαραίτητους στο χάρτη για σχεδιαστικούς σκοπούς. Δηλαδή προσθέτει ή αφαιρεί δυναμικά δείκτες από το χάρτη με αποτέλεσμα την αύξηση της ταχύτητας απεικόνισης του χάρτη και της μείωσης των άχρηστων οπτικών στοιχείων.

- Γραμμές

Αναπαριστώνται με τη χρήση των polylines (που ουσιαστικά είναι μια ακολουθία συνδεδεμένων σημείων) πάνω στο χάρτη, για τις οποίες μπορεί να καθοριστεί χρώμα, πάχος και επίπεδο διαφάνειας. Χωρίζονται σε τρεις κατηγορίες: τις γραμμές (drawing), τις γεωδαιτικές (geodesic) και τις κωδικοποιημένες (encoded).

Οι γραμμικές polylines εμφανίζονται ως ευθύγραμμα τμήματα στο χάρτη και για την απεικόνισή τους το Google Maps API εκμεταλλεύεται πιθανές σχεδιαστικές ικανότητες του φυλλομετρητή, αλλιώς γίνεται αίτηση για μια εικόνα polyline από τους εξυπηρετητές (servers) της Google. Οι γεωδαιτικές polylines εμφανίζονται ως καμπυλωτές γραμμές λαμβάνοντας υπόψη τους την καμπυλότητα της γης. Οι κωδικοποιημένες polylines είναι συνήθως μεγαλύτερες και πιο περίπλοκες από τις υπόλοιπες και αποτελούνται από πολλά ευθύγραμμα τμήματα που συνδέονται με τυχαίο τρόπο μεταξύ τους. Γι αυτό το λόγο απαιτούν περισσότερη μνήμη και χρόνο για να απεικονισθούν. Έχουν όμως το πλεονέκτημα ότι μπορούν να καθορίζουν πόσο λεπτομερής θα είναι μια polyline σε ένα δοσμένο επίπεδο εστίασης καθώς δίνουν τη δυνατότητα να αγνοούνται τμήματα της γραμμής σε συγκεκριμένες ομάδες επιπέδων εστίασης.

- Περιοχές

Αναπαριστώνται με τη χρήση είτε πολυγώνων (polygons) πάνω στο χάρτη, αν η περιοχή έχει τη μορφή αυθαίρετου σχήματος, είτε με τη χρήση ground overlays, αν η περιοχή έχει τη μορφή ορθογώνιου σχήματος. Τα πολύγωνα είναι παρόμοια με τις polylines αφού και αυτά είναι μια ακολουθία σημείων, προσθέτοντας επιπλέον ότι σχηματίζουν κύκλο και μπορούν να πάρουν οποιαδήποτε μορφή. Τα ground overlays είναι συχνά χρήσιμα για περιοχές που τα άκρα τους στοχεύουν κάθετα σε σημεία στο χάρτη.

- Ο χάρτης

Αναπαριστώνται με τη χρήση του tile overlay, κάτι το οποίο μπορεί να τροποποιηθεί μέσω του Google Maps API. Τα διαθέσιμα tiles δεν καλύπτουν όλες τις περιοχές σε όλα τα επίπεδα εστίασης. Για παράδειγμα πολλές περιοχές στον Ειρηνικό Ωκεανό δεν εμφανίζονται σε υψηλά επίπεδα εστίασης, ενώ το Μανχάταν προσφέρει πολύ λεπτομερείς εικόνες στον δορυφορικό τύπο χάρτη.

Αξίζει να σημειωθεί ότι στο μικρότερο επίπεδο εστίασης (επίπεδο 0), ένα tile αναπαριστά ολόκληρη τη γη. Αν N είναι το επίπεδο εστίασης, τότε το Google Maps διαιρεί το tile ολόκληρης της γης σε 4^N tiles και προβάλλει το κατάλληλο.

- Το παράθυρο πληροφοριών (info window)

Αποτελεί ένα ειδικό overlay. Προστίθεται στο χάρτη αυτόματα και σε κάθε χάρτη αντιστοιχεί ένα μόνο αντιστοιχεί ένα μόνο αντικείμενο παραθύρου πληροφοριών.

5 Υλοποίηση Βάσεων Δεδομένων

5.1 Τι είναι βάση δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά οργανωμένα (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων συντομευμένα (DBMS).

Ένα σύστημα βάσης δεδομένων (database system) ουσιαστικά δεν είναι παρά ένα σύστημα τήρησης εγγράφων με υπολογιστή. Με απλά λόγια, είναι ένα σύστημα για υπολογιστές, που ο γενικός σκοπός του είναι να τηρεί πληροφορίες και να δίνει αυτές τις πληροφορίες όταν ζητούνται. Οι πληροφορίες που τηρούνται σε ένα τέτοιο σύστημα μπορεί να είναι οτιδήποτε έχει σημασία για το άτομο ή τον οργανισμό που εξυπηρετεί το συγκεκριμένο σύστημα, δηλαδή οτιδήποτε χρειάζεται για την υποβοήθηση των εργασιών αυτού του ατόμου ή οργανισμού.

Μία βάση δεδομένων μπορεί να θεωρηθεί και ένα είδος ηλεκτρονικής αρχειοθήκης, με άλλα λόγια ένας χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων. Ο χρήστης του συστήματος έχει στη διάθεση του ορισμένα βοηθήματα, για να εκτελεί σε αυτά τα αρχεία διάφορες εργασίες, στις οποίες συγκαταλέγονται, ανάμεσα σε άλλες, και οι εξής:

- Η προσθήκη νέων κενών αρχείων στη βάση δεδομένων
- Η εισαγωγή νέων δεδομένων σε υπάρχοντα αρχεία
- Η ανάκληση δεδομένων από υπάρχοντα αρχεία
- Η ενημέρωση δεδομένων σε υπάρχοντα αρχεία
- Η διαγραφή δεδομένων από υπάρχοντα αρχεία
- Η αφαίρεση υπαρχόντων αρχείων, κενών ή όχι, από τη βάση δεδομένων

5.1.1 Το σύστημα διαχείρισης βάσεων δεδομένων

Το σύστημα διαχείρισης βάσεων δεδομένων (database management system, DBMS) είναι το λογισμικό που χειρίζεται όλες τις προσπελάσεις στη βάση δεδομένων. Εννοιολογικά, αυτά που συμβαίνουν είναι τα εξής:

1. Ένας χρήστης διατυπώνει μία αίτηση προσπέλασης, χρησιμοποιώντας μια συγκεκριμένη υπογλώσσα δεδομένων (συνήθως την SQL).

2. Το DBMS παίρνει αυτή την αίτηση και την αναλύει.

3. Το DBMS εξετάζει κατά σειρά το εξωτερικό σχήμα του συγκεκριμένου χρήστη, την αντίστοιχη εξωτερική/εννοιολογική απεικόνιση, το εννοιολογικό σχήμα, την εννοιολογική/εσωτερική απεικόνιση, και τον ορισμό της αποθηκευτικής δομής.

4. Το DBMS εκτελεί τις απαιτούμενες πράξεις στην αποθηκευμένη βάση δεδομένων.

Οι λειτουργίες του DBMS πρέπει να υποστηρίζουν τουλάχιστον όλα τα παρακάτω:

- Ορισμός δεδομένων

Το DBMS πρέπει να έχει τη δυνατότητα να δέχεται τους ορισμούς δεδομένων (τα εξωτερικά σχήματα, το εννοιολογικό σχήμα, το εσωτερικό σχήμα, και όλες τις σχετικές απεικονίσεις) σε μορφή πηγαίου (source) κώδικα και να τους μετατρέπει στην κατάλληλη μορφή αντικειμενικού (object) κώδικα. Με άλλα λόγια, το DBMS πρέπει να περιλαμβάνει επεξεργαστές γλώσσας (language processor) για την κάθε μια από τις διάφορες γλώσσες ορισμού δεδομένων.

- Χειρισμός δεδομένων
Το DBMS πρέπει να έχει τη δυνατότητα να χειρίζεται αιτήσεις του χρήστη για την ανάκληση, την ενημέρωση ή τη διαγραφή δεδομένων που υπάρχουν στη βάση δεδομένων ή για την προσθήκη νέων δεδομένων στη βάση δεδομένων. Αυτές οι αιτήσεις μπορούν να είναι προσχεδιασμένες (η ανάγκη τους δηλαδή είχε προβλεφθεί πριν από το χρόνο που θα εκτελεστεί πραγματικά η αίτηση) ή μη προσχεδιασμένες (η ανάγκη τους παρουσιάστηκε περιστασιακά).
- Ασφάλεια και ακεραιότητα δεδομένων
Το DBMS πρέπει να παρακολουθεί τις αιτήσεις των χρηστών και να απορρίπτει οποιεσδήποτε απόπειρες να παραβιαστούν οι κανόνες ασφάλειας και ακεραιότητας που έχουν οριστεί.
- Ανάκαμψη και ταυτοχρονισμός
Το DBMS πρέπει να επιβάλλει κάποιους ελέγχους για την ανάκαμψη και τον ταυτοχρονισμό.
- Λεξικό δεδομένων
Το λεξικό δεδομένων είναι μια βάση δεδομένων του συστήματος η οποία περιέχει μετα-δεδομένα, δηλαδή ορισμούς άλλων αντικειμένων του συστήματος. Όλα τα σχήματα και οι απεικονίσεις (εξωτερικά, εννοιολογικά, κ.λπ.) είναι αποθηκευμένα στο λεξικό, τόσο σε μορφή πηγαίου όσο και σε μορφή αντικειμενικού κώδικα.
- Απόδοση
Είναι αυτονόητο ότι το DBMS θα πρέπει να εκτελεί όλες τις λειτουργίες που είδαμε παραπάνω όσο πιο αποδοτικά γίνεται.

5.1.2 Πλεονεκτήματα της χρήσης βάσεων δεδομένων

Τα πλεονεκτήματα που προκύπτουν από τον κεντρικό έλεγχο των δεδομένων είναι τα εξής:

- Ο πλεονασμός (redundancy) μπορεί να μειωθεί στο ελάχιστο.
Στα συμβατικά συστήματα (εκείνα που δεν είναι συστήματα βάσεων δεδομένων), η κάθε εφαρμογή έχει τα δικά της αρχεία. Αυτό το γεγονός οδηγεί πολύ συχνά σε υψηλό βαθμό πλεονασμού (επανάληψης) για τα αποθηκευμένα δεδομένα, με αποτέλεσμα την σπατάλη αποθηκευτικού χώρου.
- Η ασυνέπεια μπορεί να αποφευχθεί (ως ένα βαθμό).
Στην ουσία, αυτό είναι συνέπεια της προηγούμενης παρατήρησης.
Αν υποθέσουμε ότι μια πληροφορία αναπαριστάται στη βάση μας από δύο εγγραφές και ότι το DBMS δεν είναι ενημερωμένο για αυτή τη «διπλοεγγραφή» (δηλαδή ότι ο πλεονασμός δεν είναι ελεγχόμενος), τότε είναι βέβαιο ότι θα υπάρξουν περιπτώσεις που οι δυο καταχωρήσεις δεν θα συμφωνούν μεταξύ τους. Όταν δηλαδή η μια καταχώρηση θα έχει ενημερωθεί ενώ η άλλη όχι. Στις περιπτώσεις αυτές λέμε ότι η βάση δεδομένων είναι ασυνεπής (inconsistent).

- Τα δεδομένα μπορούν να είναι κοινόχρηστα.
Η κοινή χρήση δε σημαίνει μόνο ότι οι υπάρχουσες εφαρμογές μπορούν να μοιράζονται τα δεδομένα της βάσης δεδομένων αλλά και ότι είναι δυνατή η ανάπτυξη νέων εφαρμογών που θα μπορούν να χρησιμοποιούν τα ίδια αποθηκευμένα δεδομένα. Με άλλα λόγια, είναι δυνατό να ικανοποιούνται οι απαιτήσεις νέων εφαρμογών σε ό,τι αφορά τα δεδομένα, χωρίς να είναι αναγκαία η δημιουργία νέων αποθηκευμένων δεδομένων.
- Μπορούν να επιβάλλονται πρότυπα.
Τα πρότυπα αυτά μπορεί να είναι πρότυπα της επιχείρησης, του εγκατεστημένου συστήματος, του τμήματος της επιχείρησης, του τεχνολογικού τομέα, ή εθνικά και διεθνή πρότυπα. Η τυποποίηση της αναπαράστασης των δεδομένων διευκολύνει ιδιαίτερα την ανταλλαγή δεδομένων (data interchange), δηλαδή τη μεταφορά δεδομένων μεταξύ διαφορετικών συστημάτων. Τα πρότυπα ονομασίας και τεκμηρίωσης των δεδομένων είναι επίσης πολύ επιθυμητά για να διευκολύνεται ο μερισμός και η καλύτερη κατανόηση των δεδομένων.
- Μπορούν να εφαρμόζονται περιορισμοί ασφάλειας.
Μπορεί να εξασφαλιστεί ότι η πρόσβαση στη βάση δεδομένων θα μπορεί να γίνεται μόνο μέσω κατάλληλων καναλιών και, κατά συνέπεια, μπορούν να οριστούν κανόνες ασφάλειας με βάση τους οποίους θα γίνεται έλεγχος κάθε φορά που θα υπάρχει απόπειρα προσπέλασης εμπιστευτικών δεδομένων. Είναι δυνατό να καθοριστούν διαφορετικοί κανόνες για το κάθε είδος προσπέλασης (ανάκληση, εισαγωγή, διαγραφή κ.λπ.) σε κάθε στοιχείο πληροφοριών της βάσης δεδομένων.
- Μπορεί να διατηρείται η ακεραιότητα.
Το πρόβλημα της ακεραιότητας (integrity) είναι να εξασφαλίζεται ότι τα δεδομένα της βάσης δεδομένων είναι ακριβή. Η ασυμφωνία μεταξύ δυο καταχωρίσεων που υποτίθεται ότι αντιπροσωπεύουν το ίδιο γεγονός είναι ένα παράδειγμα έλλειψης ακεραιότητας. Φυσικά, αυτό το συγκεκριμένο πρόβλημα μπορεί να παρουσιαστεί μόνο αν υπάρχει πλεονασμός στα αποθηκευμένα δεδομένα. Ακόμη και αν δεν υπάρχει πλεονασμός όμως, πάλι υπάρχει περίπτωση να περιέχει η βάση δεδομένων λανθασμένες πληροφορίες.
- Οι αντικρουόμενες απαιτήσεις μπορούν να εξισορροπούνται.
Το κάθε σύστημα μπορεί να δομηθεί με τέτοιο τρόπο, ώστε να παρέχει γενικές υπηρεσίες οι οποίες θα είναι βέλτιστες για την κάθε επιχείρηση.
- Ανεξαρτησία δεδομένων
Η ανεξαρτησία δεδομένων είναι πολλή σημαντική για τις βάσεις δεδομένων και μπορεί να οριστεί ως ανοσία των εφαρμογών στις αλλαγές της αποθηκευτικής δομής και της τεχνικής της προσπέλασης δεδομένων. Αυτό σημαίνει ότι οι συγκεκριμένες εφαρμογές δεν πρέπει να εξαρτώνται από καμία αποθηκευτική δομή ή τεχνική προσπέλασης.

5.2 Το Μοντέλο Οντοτήτων-Συσχετίσεων

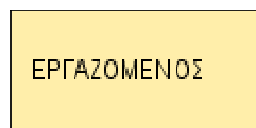
Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο Ο/Σ - ER model) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Στη μηχανική λογισμικού χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του με top-down προσέγγιση. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης καλείται διάγραμμα οντοτήτων-συσχετίσεων, ή διάγραμμα Ο/Σ ή ΟΣΔ εν συντομία. Προτάθηκε αρχικά το 1976 από τον Peter Chen, ωστόσο στη συνέχεια επινοήθηκαν πολλές παραλλαγές της διαδικασίας.

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεών του. Σκοπός του είναι να περιγράψει τις αναγκαίες πληροφορίες, οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους. Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεών τους σε έναν ορισμένο τομέα ενδιαφέροντος. Στην περίπτωση σχεδιασμού ενός συστήματος πληροφοριών, που στηρίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων χαρτογραφείται σε προχωρημένο στάδιο σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Το στάδιο αυτό ονομάζεται συνήθως στάδιο λογικού σχεδιασμού. Ύστερα, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο χαρτογραφείται σε κάποιο φυσικό μοντέλο. Ας σημειωθεί ότι ορισμένες φορές και οι δύο φάσεις αναφέρονται ως "φυσικός σχεδιασμός".

5.2.1 Βασικές έννοιες Μοντέλου Οντοτήτων-Συσχετίσεων

Βάση των μοντέλων οντοτήτων-συσχετίσεων είναι η κατηγοριοποίηση των αντικειμένων και των σχέσεών τους μεταξύ τους.

Οντότητα

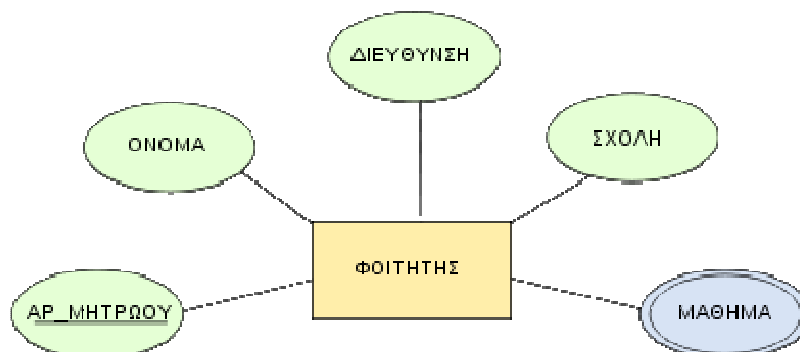


Εικόνα 12 Οντότητα

Οι διάφοροι τύποι οντοτήτων παριστάνονται στο διάγραμμα Ο/Σ με ένα ορθογώνιο.

Οντότητα (entity) είναι ένα αντικείμενο ενδιαφέροντος στον πραγματικό κόσμο το οποίο ξεχωρίζει από τα υπόλοιπα. Μια οντότητα λειτουργεί αφαιρετικά σε έναν πολύπλοκο τομέα. Οντότητες μπορεί να είναι άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες κλπ. **Στιγμιότυπο** (instance) μιας οντότητας είναι μια συγκεκριμένη περίπτωση ενός τύπου οντότητας.

Χαρακτηριστικό



Εικόνα 13 Χαρακτηριστικό

Ένας φοιτητής μπορεί να έχει το πεδίο ΜΑΘΗΜΑ, το οποίο όμως δέχεται ως τιμές ένα σύνολο μαθημάτων που παρακολουθεί.

Κάθε οντότητα έχει διάφορα στοιχεία που την προσδιορίζουν. Ένα τέτοιο στοιχείο ονομάζεται **ιδιότητα** (attribute), **χαρακτηριστικό** ή **πεδίο** της οντότητας. Τα χαρακτηριστικά χωρίζονται σε

- **μονότιμα** (single valued), τα οποία έχουν μόνο μια τιμή και
- **πλειότιμα** (multi-valued), τα οποία έχουν σύνολο από τιμές

Στο διάγραμμα Ο/Σ οι ιδιότητες που έχει μια οντότητα παριστάνονται μέσα σε έλλειψη, με υπογραμμισμένο το πρωτεύον κλειδί. Τα πλειότιμα χαρακτηριστικά μιας οντότητας παριστάνονται μέσα σε έλλειψη με διπλό περίγραμμα.

Συσχέτιση



Εικόνα 14 Συσχέτιση

Η συσχέτιση σε αυτό το παράδειγμα ονομάζεται "Παρακολουθεί". Τα βέλη δείχνουν μια συσχέτιση πολλά-προς-πολλά. Κάθε φοιτητής μπορεί να παρακολουθεί ένα ή περισσότερα μαθήματα. Κάθε μάθημα μπορεί να παρακολουθείται από έναν ή περισσότερους μαθητές. Από τη σχέση αυτή μπορούμε να βρούμε ποιοι φοιτητές παρακολουθούν ένα μάθημα ή ποια μαθήματα παρακολουθεί ένας φοιτητής.

Συσχέτιση (relationship) είναι ο σύνδεση δύο ή περισσότερων τύπων οντοτήτων που παρουσιάζει ενδιαφέρον για σχεδιασμό. Με συσχετίσεις μπορούν να συνδέονται και χαρακτηριστικά οντοτήτων.

Ένας τύπος συσχέτισης (σύνολο συσχετίσεων) παριστάνεται με ρόμβο. Στο εσωτερικό αναγράφεται το όνομα με μικρά γράμματα.

Υποδεικνύουμε τα όρια της συσχέτισης με ένα δείκτη.

Ως όρια μπόουμε να συναντήσουμε:

0 έως άπειρο

(κατώτατο όριο 0, ανώτατο όριο άπειρο)

τουλάχιστον 1

(κατώτατο όριο 1, ανώτατο όριο άπειρο)

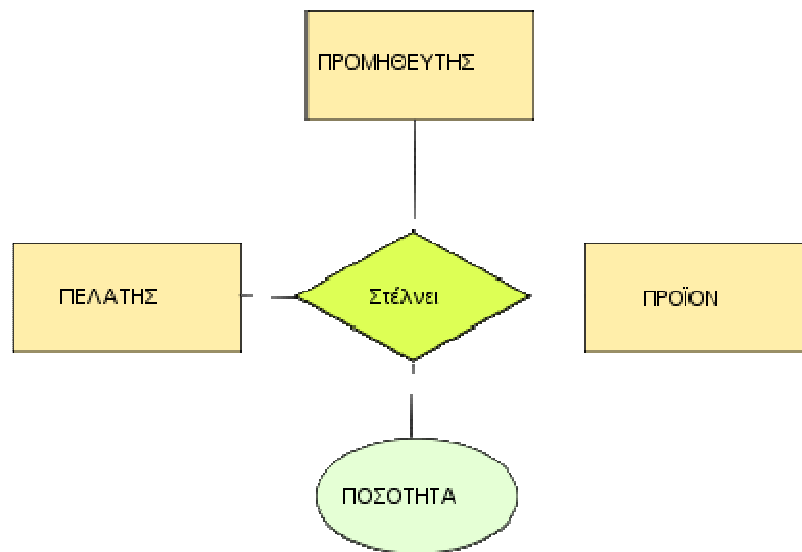
ακριβώς 1

(κατώτατο όριο 1, ανώτατο όριο 1)

το πολύ 1

(κατώτατο όριο 0, ανώτατο όριο 1)

Βαθμός ή πολυπλοκότητα ενός τύπου συσχετίσεων



Εικόνα 15 Βαθμός ή πολυπλοκότητα

Βαθμός συσχέτισης: 3 (Οι τύποι οντοτήτων PROMHΘEYTHS, PELATHS και PROION παίρνουν μέρος στη συσχέτιση)

Ο βαθμός μιας συσχέτισης είναι ο αριθμός των τύπων οντοτήτων που παίρνουν μέρος στη συσχέτιση. Οι πιο συνηθισμένες συσχετίσεις είναι

- μοναδικές, ο βαθμός τους τότε είναι 1
- δυαδικές, ο βαθμός τους τότε είναι 2
- τριαδικές, ο βαθμός τους τότε είναι 3

Πληθικότητα

Η **πληθικότητα** (cardinality), περιγράφει τον αριθμό στιγμιοτύπων ενός τύπου οντοτήτων που μπορούν να αντιστοιχίζονται με μία οντότητα ενός άλλου τύπου σε μια συσχέτιση.

Ο **λόγος πληθικότητας** ή **πληθικός λόγος** (cardinality ratio), είναι ο λόγος των πληθικοτήτων μιας συσχέτισης.

Μπορούμε να έχουμε συσχετίσεις με λόγο πληθικότητας:

1-1 (ένα-προς-ένα)

Αντιστοιχίζεται μια οντότητα ενός τύπου με το πολύ ή ακριβώς μια οντότητα ενός άλλου τύπου.

1-N (ένα-προς-πολλά)

Αντιστοιχίζεται μια οντότητα ενός τύπου με κανένα, ένα ή πολλά στιγμιότυπα ενός άλλου τύπου.

M-N (πολλά-προς-πολλά)

Αντιστοιχίζεται κάθε στιγμιότυπο του ενός τύπου με ένα, κανένα ή πολλά στιγμιότυπα του άλλου τύπου.

Ασθενής Τύπος Οντότητας

Αδύναμη ή ασθενής οντότητα λέγεται μια οντότητα που εξαρτάται από την ύπαρξη κάποιας άλλης. Οι αδύναμες οντότητες συμμετέχουν σε συσχετίσεις μέσω ταυτοποιητικών συσχετίσεων με ισχυρή οντότητα.

Ταυτοποιητική συσχέτιση ονομάζεται η συσχέτιση στην οποία το πρωτεύον κλειδί της ισχυρής οντότητας χρησιμοποιείται ως μέρος του πρωτεύοντος κλειδιού της αδύναμης οντότητας.













Διακριτικό ή μερικό κλειδί ονομάζεται το χαρακτηριστικό της αδύναμης οντότητας το οποίο μαζί με το πρωτεύον κλειδί της ισχυρής οντότητας είναι το πρωτεύον κλειδί της αδύναμης.

Κατά την αναπαράσταση αδύναμων οντοτήτων:

- η οντότητα παριστάνεται με διπλό ορθογώνιο
- ταυτοποιητική συσχέτιση με διπλό ρόμβο
- το μερικό κλειδί με διακεκομμένη γραμμή

5.3 Η βάση δεδομένων maps

Η βάση δεδομένων “maps” δημιουργήθηκε με σκοπό να αποθηκευτούν όλα τα δεδομένα που αφορούν το χάρτη του ιστοτόπου. Περιέχει τρεις πίνακες δεδομένων, όπως φαίνεται και στην παρακάτω εικόνα. Σε κάθε πίνακα δηλώνεται ο τύπος του πεδίου (π.χ. char, int, date), και το πρωτεύον του κλειδί (primary key), το οποίο πρέπει να είναι μοναδικό.

	Πίνακας ▾	Ενέργεια	Εγγραφές ¹	Τύπος	Collation	Μέγεθος	Επιβάρυνση
<input type="checkbox"/>	city_info	     	7	MyISAM	utf8_unicode_ci	2,5 KB	-
<input type="checkbox"/>	points	     	7	MyISAM	utf8_unicode_ci	3,0 KB	-
<input type="checkbox"/>	points_categories	     	7	MyISAM	utf8_unicode_ci	2,4 KB	-
	3 Πίνακας/Πίνακες	Σύνολο	21	MyISAM	latin1_swedish_ci	7,8 KB	0 Bytes

Εικόνα 16 Η βάση δεδομένων maps

5.3.1 Πίνακας city_info

Τα πεδία του πίνακα city_info είναι τα εξής:

- id : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, δέχεται μέχρι τέσσερα ψηφία και είναι auto_increment.

- title : Είναι τύπου varchar, δέχεται μέχρι εκατόν πενήντα χαρακτήρες και αφορά την επωνυμία της επιχείρησης που περιγράφει.
- lat : Είναι το latitude του σημείου στο χάρτη, είναι τύπου varchar και δέχεται μέχρι τριάντα χαρακτήρες.
- lng : Είναι το longitude του σημείου στο χάρτη, είναι τύπου varchar και δέχεται μέχρι τριάντα χαρακτήρες.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	id	int(4)		UNSIGNED	Όχι	None	auto_increment	
<input type="checkbox"/>	title	varchar(150)	utf8_unicode_ci		Όχι	None		
<input type="checkbox"/>	lat	varchar(30)	utf8_unicode_ci		Όχι	None		
<input type="checkbox"/>	lng	varchar(30)	utf8_unicode_ci		Όχι	None		

Εικόνα 17 Πίνακας city_info

5.3.2 Πίνακας points

Τα πεδία του πίνακα points είναι τα εξής:

- id : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, δέχεται μέχρι τέσσερα ψηφία και είναι auto_increment.
- category : Είναι τύπου varchar, δέχεται μέχρι τρεις χαρακτήρες και δείχνει την κατηγορία στην οποία ανήκει το σημείο (βενζινάδικο, σουπερ μάρκετ ή κινηματογράφος).
- city : Είναι η πόλη στην οποία βρίσκεται το σημείο, είναι τύπου varchar και δέχεται μέχρι δεκαπέντε χαρακτήρες.
- address : Είναι η διεύθυνση στην οποία βρίσκεται το σημείο, είναι τύπου varchar και δέχεται μέχρι διακόσιους πενήντα χαρακτήρες.
- title : Είναι τύπου varchar, δέχεται μέχρι εκατόν πενήντα χαρακτήρες και αφορά την επωνυμία της επιχείρησης που περιγράφει.
- lat : Είναι το latitude του σημείου στο χάρτη, είναι τύπου varchar και δέχεται μέχρι τριάντα χαρακτήρες.
- lng : Είναι το longitude του σημείου στο χάρτη, είναι τύπου varchar και δέχεται μέχρι τριάντα χαρακτήρες.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια						
<input type="checkbox"/>	id	int(5)		UNSIGNED	Όχι	None	auto_increment							
<input type="checkbox"/>	category	varchar(3)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	city	varchar(15)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	address	varchar(250)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	title	varchar(150)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	lat	varchar(30)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	lng	varchar(30)	utf8_unicode_ci		Όχι	None								

Εικόνα 18 Πίνακας points

5.3.3 Πίνακας points_categories

Τα πεδία του πίνακα points_categories είναι τα εξής:











































- **id** : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, δέχεται μέχρι τέσσερα ψηφία και είναι auto_increment.
- **title** : Είναι τύπου varchar, δέχεται μέχρι εκατόν πενήντα χαρακτήρες και αφορά την επωνυμία της επιχείρησης που περιγράφει.
- **en_title** : Είναι τύπου varchar, δέχεται μέχρι εκατό χαρακτήρες.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια						
<input type="checkbox"/>	id	int(3)		UNSIGNED	Όχι	None	auto_increment							
<input type="checkbox"/>	title	varchar(150)	utf8_unicode_ci		Όχι	None								
<input type="checkbox"/>	en_title	varchar(100)	utf8_unicode_ci		Όχι	None								

Εικόνα 19 Πίνακας points_categories

5.4 Η βάση δεδομένων syst_kathodigisis

Η βάση δεδομένων “syst_kathodigisis” δημιουργήθηκε με σκοπό να αποθηκεύονται όλα τα δεδομένα που αφορούν τους χρήστες, οι οποίοι είναι εγγεγραμμένοι στον ιστότοπο καθώς και πληροφορίες για τα σημεία ενδιαφέροντος. Περιέχει επτά πίνακες δεδομένων, όπως φαίνεται και στην παρακάτω εικόνα. Σε κάθε πίνακα δηλώνεται ο τύπος του πεδίου (π.χ. char, int, date), και το πρωτεύον του κλειδί (primary key), το οποίο πρέπει να είναι μοναδικό.

	Πίνακας ▾	Ενέργεια	Εγγραφές ¹	Τύπος	Collation	Μέγεθος	Επιβάρυνση
<input type="checkbox"/>	cinema	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/>	cinema_info	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/>	supermarket_info	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/>	super_market	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/>	users	     	1	MyISAM	latin1_swedish_ci	4,1 KB	-
<input type="checkbox"/>	venzinadiko	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
<input type="checkbox"/>	venzinadiko_info	     	0	MyISAM	latin1_swedish_ci	1,0 KB	-
	7 Πίνακας/Πίνακες	Σύνολο	1	MyISAM	latin1_swedish_ci	10,1 KB	0 Bytes

Εικόνα 20 Η βάση δεδομένων syst_kathodigisis














































































5.4.1 Πίνακας users

Στον πίνακα users αποθηκεύονται όλα τα στοιχεία των χρηστών του ιστότοπου κατά την εγγραφή τους.

Τα πεδία του είναι τα εξής:

- id_xristi : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, auto increment και δέχεται μέχρι έντεκα ψηφία.
- onoma : Είναι τύπου varchar, δέχεται μέχρι εξήντα χαρακτήρες και αποθηκεύει το όνομα του χρήστη.
- eritheto : Είναι τύπου varchar, δέχεται μέχρι εξήντα χαρακτήρες και αποθηκεύει το επίθετο του χρήστη.
- ar_pistotikis : Είναι τύπου int, δέχεται μέχρι έντεκα χαρακτήρες και αποθηκεύει τον αριθμό της πιστωτικής του χρήστη.
- telefono : Είναι τύπου int, δέχεται μέχρι έντεκα χαρακτήρες και αποθηκεύει τον αριθμό τηλεφώνου του χρήστη.
- perioxi : Είναι τύπου varchar, δέχεται μέχρι εξήντα χαρακτήρες και αποθηκεύει την πόλη στην οποία κατοικεί ο χρήστης.
- odos : Είναι τύπου varchar, δέχεται μέχρι σαράντα χαρακτήρες και αποθηκεύει την οδό στην οποία κατοικεί ο χρήστης.
- arithmos : Είναι τύπου int, δέχεται μέχρι έντεκα χαρακτήρες και αποθηκεύει τον αριθμό της οδού στην οποία κατοικεί ο χρήστης.
- username : Είναι τύπου varchar, δέχεται μέχρι εξήντα χαρακτήρες και αποθηκεύει το username του χρήστη.

- password : Είναι τύπου varchar, δέχεται μέχρι εξήντα χαρακτήρες και αποθηκεύει το password του χρήστη.
- e-mail : Είναι τύπου varchar, δέχεται μέχρι πενήντα χαρακτήρες και αποθηκεύει το e-mail του χρήστη.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	onoma	varchar(60)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	epitheto	varchar(60)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	ar_pistotikis	int(11)			Όχι	None		      
<input type="checkbox"/>	tilefono	int(11)			Όχι	None		      
<input type="checkbox"/>	perioxi	varchar(60)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	odos	varchar(40)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	arithmos	int(11)			Όχι	None		      
<input type="checkbox"/>	<u>id_xristi</u>	int(11)			Όχι	None	auto_increment	      
<input type="checkbox"/>	username	varchar(60)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	password	varchar(60)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	e-mail	varchar(50)	utf8_unicode_ci		Όχι	None		      




































Εικόνα 21 Πίνακας users

5.4.2 Πίνακας cinema

Στον πίνακα cinema αποθηκεύονται τα δεδομένα που αφορούν την τοποθεσία των κινηματογράφων που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

- id : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, auto increment και δέχεται μέχρι έντεκα ψηφία.
- odos : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την οδό όπου βρίσκεται ο κινηματογράφος.
- arithmos : Είναι τύπου int, δέχεται μέχρι δυο χαρακτήρες και αποθηκεύει τον αριθμό της οδού όπου βρίσκεται ο κινηματογράφος.
- perioxi : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την πόλη στην οποία βρίσκεται ο κινηματογράφος.
- id_info : Είναι τύπου int, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα cinema με τον πίνακα cinema_info.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>id</u>	int(11)			Όχι	None	auto_increment	      
<input type="checkbox"/>	odos	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	arithmos	int(2)			Όχι	None		      
<input type="checkbox"/>	perioxi	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	id_info	int(3)			Όχι	None		      





























Εικόνα 22 Πίνακας cinema

5.4.3 Πίνακας cinema_info

Στον πίνακα cinema_info αποθηκεύονται όλα τα δεδομένα που αφορούν στη θεματολογία και το ωράριο των κινηματογράφων που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

- id_info : Είναι τύπου int, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα cinema με τον πίνακα cinema_info.
- date : Είναι η ημερομηνία, την οποία προβάλλεται η ταινία.
- time : Είναι η ώρα έναρξης της προβολής της ταινίας.
- movie : Είναι τύπου varchar, δέχεται μέχρι εκατό χαρακτήρες και αποθηκεύει τον τίτλο της ταινίας που πρόκειται να προβληθεί.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>id_info</u>	int(3)			Όχι	None		      
<input type="checkbox"/>	date	date			Όχι	None		      
<input type="checkbox"/>	time	time			Όχι	None		      
<input type="checkbox"/>	movie	varchar(100)	utf8_unicode_ci		Όχι	None		      




































Εικόνα 23 Πίνακας cinema_info

5.4.4 Πίνακας super_market

Στον πίνακα super_market αποθηκεύονται όλα τα δεδομένα που αφορούν στην τοποθεσία των σούπερ μαρκετ που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

- id : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, auto increment και δέχεται μέχρι έντεκα ψηφία.
- odos : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την οδό όπου βρίσκεται το σούπερ μαρκετ.
- arithmos : Είναι τύπου int, δέχεται μέχρι δυο χαρακτήρες και αποθηκεύει τον αριθμό της οδού όπου βρίσκεται το σούπερ μαρκετ.
- perioxi : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την πόλη στην οποία βρίσκεται το σούπερ μαρκετ.
- id_info : Είναι τύπου int, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα super_market με τον πίνακα supermarket_info.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	id	int(11)			Όχι	None	auto_increment	      
<input type="checkbox"/>	perioxi	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	odos	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	arithmos	int(2)			Όχι	None		      
<input type="checkbox"/>	id_info	int(3)			Όχι	None		      





























Εικόνα 24 Πίνακας super_market

5.4.5 Πίνακας supermarket_info

Στον πίνακα supermarket_info αποθηκεύονται όλα τα δεδομένα που αφορούν στην επωνυμία και το ωράριο των σούπερ μάρκετ που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

- id_info : Είναι τύπου int, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα super-market με τον πίνακα supermarket_info.
- date : Είναι η ημερομηνία, την οποία είναι ανοικτό το σούπερ μάρκετ.
- time : Είναι η ώρα, την οποία ανοίγει το σούπερ μάρκετ.
- name : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την επωνυμία του σούπερ μάρκετ.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>id_info</u>	int(3)			Όχι	None		      
<input type="checkbox"/>	time	time			Όχι	None		      
<input type="checkbox"/>	date	date			Όχι	None		      
<input type="checkbox"/>	name	varchar(30)	utf8_unicode_ci		Όχι	None		      


















Εικόνα 25 Πίνακας supermarket_info

5.4.6 Πίνακας venzinadiko

Στον πίνακα venzinadiko αποθηκεύονται όλα τα δεδομένα που αφορούν στην τοποθεσία των πρατηρίων βενζίνης που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

- id : Είναι το πρωτεύον κλειδί του πίνακα, τύπου int, auto increment και δέχεται μέχρι έντεκα ψηφία.
- odos : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την οδό όπου βρίσκεται το σούπερ μαρκετ.
- arithmos : Είναι τύπου int, δέχεται μέχρι δυο χαρακτήρες και αποθηκεύει τον αριθμό της οδού όπου βρίσκεται το σούπερ μαρκετ.
- perioxi : Είναι τύπου varchar, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την πόλη στην οποία βρίσκεται το σούπερ μαρκετ.
- id_info : Είναι τύπου int, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα supermarket με τον πίνακα supermarket_info.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>id</u>	int(11)			Όχι	None	auto_increment	      
<input type="checkbox"/>	perioxi	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	odos	varchar(30)	utf8_unicode_ci		Όχι	None		      
<input type="checkbox"/>	arithmos	int(2)			Όχι	None		      
<input type="checkbox"/>	id_info	int(3)			Όχι	None		      



























Εικόνα 26 Πίνακας venzinadiko

5.4.7 Πίνακας `venzinadiko_info`

Στον πίνακα `venzinadiko_info` αποθηκεύονται όλα τα δεδομένα που αφορούν στην επωνυμία και το ωράριο των πρατηρίων βενζίνης που υπάρχουν στο σύστημα.

Τα πεδία του είναι τα εξής:

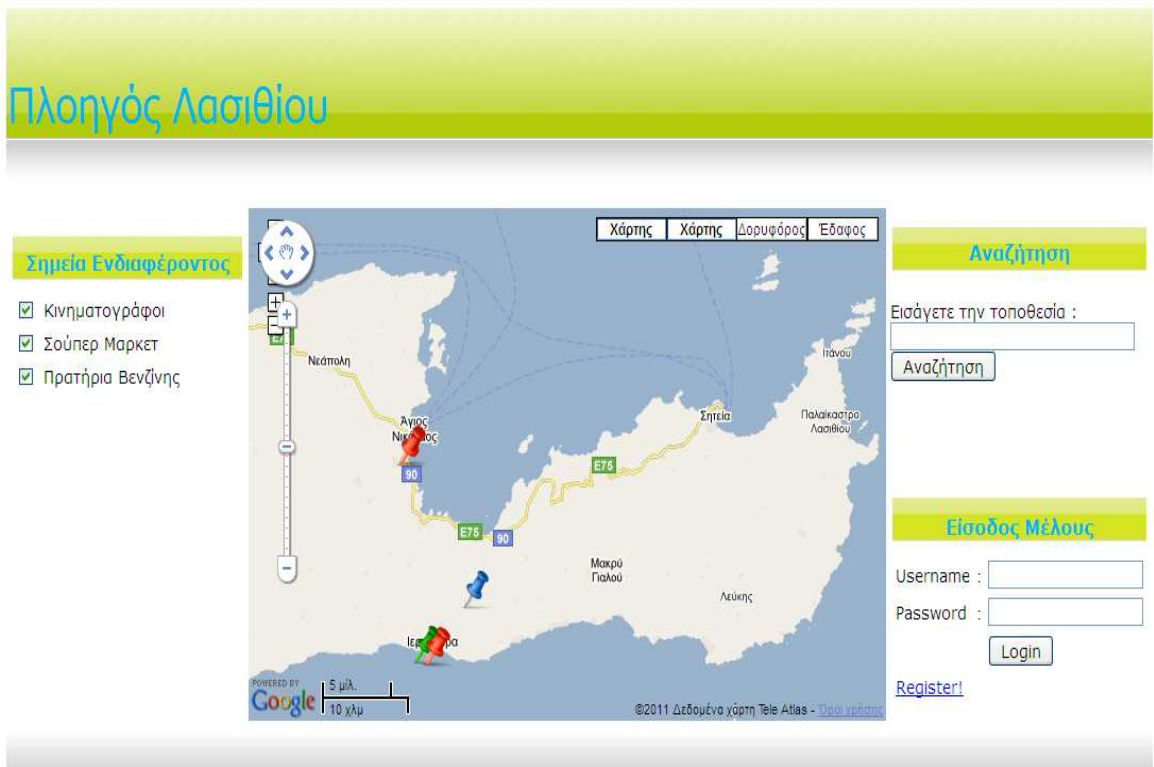
- `id_info` : Είναι τύπου `int`, δέχεται μέχρι τρεις χαρακτήρες και συνδέει τον πίνακα `venzinadiko` με τον πίνακα `venzinadiko_info`.
- `date` : Είναι η ημερομηνία, την οποία είναι ανοικτό το πρατήριο βενζίνης.
- `time` : Είναι η ώρα την οποία ανοίγει το πρατήριο βενζίνης.
- `name` : Είναι τύπου `varchar`, δέχεται μέχρι τριάντα χαρακτήρες και αποθηκεύει την επωνυμία του πρατηρίου βενζίνης.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<code>id_info</code>	<code>int(3)</code>			Όχι	None		      
<input type="checkbox"/>	<code>date</code>	<code>date</code>			Όχι	None		      
<input type="checkbox"/>	<code>time</code>	<code>time</code>			Όχι	None		      
<input type="checkbox"/>	<code>name</code>	<code>varchar(30)</code>	<code>utf8_unicode_ci</code>		Όχι	None		      

Εικόνα 27 Πίνακας `venzinadiko_info`

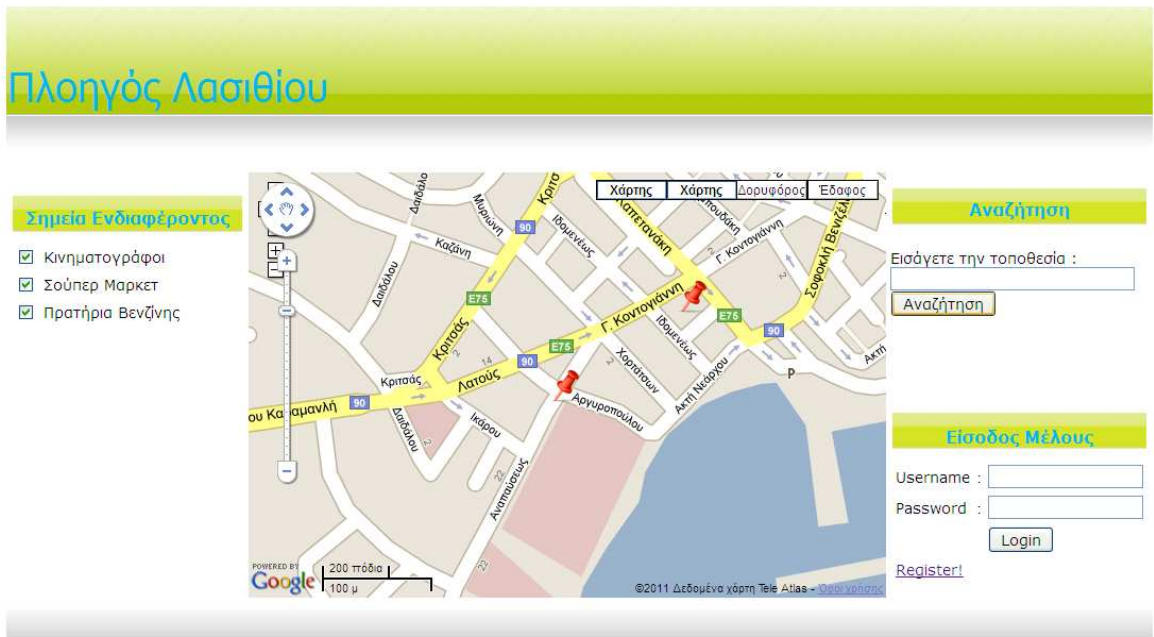
6 Αρχιτεκτονική Εφαρμογής

Στην αρχική σελίδα, όπως φαίνεται και στην παρακάτω εικόνα, εμφανίζεται ο χάρτης και τα markers. Η αρχική σελίδα φορτώνεται από το αρχείο main_a.php και τα markers εμφανίζονται με τη βοήθεια του genxml.php το οποίο δημιουργεί ένα xml αρχείο απ' όπου αντλούνται τα στοιχεία τους. Στα αριστερά εμφανίζονται τα σημεία ενδιαφέροντος με checkboxes, τα οποία ο χρήστης μπορεί να επιλέξει και να εμφανίσει στο χάρτη τα ανάλογα markers. Στα δεξιά υπάρχει πεδίο για αναζήτηση όπου μπορούμε να εισάγουμε την διεύθυνση που μας ενδιαφέρει.



Εικόνα 28 Αρχική σελίδα

Η εφαρμογή θα αναζητήσει τη διεύθυνση αυτή στη βάση δεδομένων και εφόσον υπάρχει θα μας εμφανίσει το σημείο στο χάρτη κάνοντας επίσης το ζουμ, το οποίο έχουμε ρυθμίσει. Πατώντας το κουμπί της αναζήτησης κατευθυνόμαστε στην find.php, όπου γίνεται η αναζήτηση και έπειτα μας κατευθύνει στην main.php. Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα της διεύθυνσης «Αναπαύσεως 7».



Εικόνα 29 Παράδειγμα αναζήτησης

Στην περίπτωση που η διεύθυνση που εισαχθεί δεν βρεθεί, εμφανίζεται το μήνυμα που φαίνεται στην εικόνα το οποίο ενημερώνει το χρήστη ότι η διεύθυνση δεν βρέθηκε και εμφανίζει ένα link με το οποίο μπορεί να επιστρέψει πίσω στην αρχική σελίδα (main_a.php).

Address not found![Back to main page](#)

Εικόνα 30 Η αναζήτηση δεν πέτυχε

Ένα άλλο πεδίο που υπάρχει στον ιστότοπο είναι η είσοδος μέλους. Οι χρήστες εισάγουν το username και το password τους και πατώντας το κουμπί login κατευθυνόμαστε στην checklogin.php όπου γίνεται η αναζήτηση των στοιχείων τους στη βάση. Στην περίπτωση που δεν βρεθούν εμφανίζεται το μήνυμα που φαίνεται στην εικόνα που ακολουθεί.

Wrong Username or Password! [Go back to main page to register](#)

Εικόνα 31 Η είσοδος απέτυχε

Αν τα στοιχεία βρεθούν οδηγούμαστε στο αρχείο login_success.php το οποίο μας ενημερώνει για την επιτυχία της εισόδου στο σύστημα και με τη σειρά του μας κατευθύνει στην αρχική μας σελίδα (main_a.php).

Πατώντας το link “Register” ο χρήστης κατευθύνεται σε μία φόρμα (forma.php) στην οποία καλείται να συμπληρώσει τα στοιχεία του. Με το κουμπί «Υποβολή» τα στοιχεία του στέλνονται στη βάση όπου και αποθηκεύονται με τη βοήθεια της insert.php.

The image shows a registration form with a green header containing the text 'Πληγός Λασιθίου'. Below the header, there is a list of input fields for personal and contact information, followed by a 'Υποβολή' (Submit) button. The fields are:

Όνομα	:	<input type="text"/>
Επίθετο	:	<input type="text"/>
E-mail	:	<input type="text"/>
Τηλέφωνο	:	<input type="text"/>
Διεύθυνση	:	<input type="text"/>
Περιοχή	:	<input type="text"/>
Οδός	:	<input type="text"/>
Αριθμός	:	<input type="text"/>
Αριθμός πιστωτικής	:	<input type="text"/>
Username	:	<input type="text"/>
Password	:	<input type="text"/>

Υποβολή

Εικόνα 32 Register

Στην περίπτωση που κάποιο από τα στοιχεία δεν συμπληρωθεί εμφανίζεται το μήνυμα που φαίνεται στην παρακάτω εικόνα, το οποίο ενημερώνει το χρήστη ότι πρέπει να συμπληρωθούν.



Εικόνα 33 Η υποβολή απέτυχε

Για να γίνουν όμως όλα αυτά πρέπει να γίνει αρχικά σύνδεση με τις βάσεις δεδομένων που έχουμε δημιουργήσει. Αυτή η σύνδεση επιτυγχάνεται καλώντας στην αρχή κάθε αρχείου τα `config_map.php` και `config_syst_kath.php`. Το πρώτο μας συνδέει στη βάση `map` και το δεύτερο στη βάση `syst_kathodigisis`.

Η εμφάνιση της σελίδας οφείλεται στο αρχείο `style.css` απ' όπου αντλούνται όλες οι πληροφορίες που έχουν να κάνουν με τις γραμματοσειρές, τα χρώματα και τις εικόνες που χρησιμοποιούνται.

7 Αποτελέσματα

7.1 Συμπεράσματα

Η πτυχιακή εστίασε στο σχεδιασμό και την ανάπτυξη ενός πληροφοριακού συστήματος (GIS) για την καθοδήγηση των χρηστών. Έγινε χρήση γλωσσών προγραμματισμού όπως η PHP, η MySQL, η Javascript, η HTML και CSS . Εξετάστηκε ο τρόπος χρήσης της PHP με τη χρησιμοποίηση της βάσης δεδομένων MySQL που είναι απαραίτητες για τη δημιουργία δυναμικών ιστοσελίδων.

Επιπλέον, έγινε χρήση της Διεπαφής Προγραμματισμού Εφαρμογών Google Maps, με τη βοήθεια της οποίας ενσωματώσαμε το χάρτη στην ιστοσελίδα μας. Μελετήθηκε ο τρόπος λειτουργίας ενός δικτυακού GIS και παρουσιάστηκαν τα χαρακτηριστικά καθώς και τα πλεονεκτήματά τους.

7.2 Μελλοντική Εργασία και Επεκτάσεις

Η παρούσα πτυχιακή εργασία είχε ως στόχο το σχεδιασμό και την υλοποίηση ενός πληροφοριακού συστήματος (GIS) με σκοπό την καθοδήγηση των χρηστών στο νομό Λασιθίου. Το Σύστημα που δημιουργήθηκε είναι δυναμικό και παρέχει μεγάλες δυνατότητες ανάπτυξης και επέκτασης. Εκτός από τα δεδομένα, το σύστημα είναι ανοιχτό και στην προσθήκη νέων λειτουργιών. Ένα τέτοιο παράδειγμα θα μπορούσε να ήταν η προσθήκη της δυνατότητας εμφάνισης της απόστασης μεταξύ δυο σημείων ή η αναζήτησης βέλτιστης διαδρομής ή ακόμη και η διασύνδεση της εφαρμογής απευθείας σε κινητές συσκευές.

Βιβλιογραφία

- [1] Ανάπτυξη Web εφαρμογών με PHP και MySQL , Luke Welling, Laura Thomson
- [2] Οδηγός της Javascript , John Pollock , εκδόσεις Μ.Γκιούρδας
- [3] Εισαγωγή στα συστήματα βάσεων δεδομένων , C.J.Date εκδόσεις Κλειδάριθμος
- [4] <http://en.wikipedia.org/>
- [5] <http://php.about.com/>
- [6] <http://www.mysql.com/>
- [7] <http://www.w3schools.com/>
- [8] <http://code.google.com/intl/el-GR/apis/maps/>
- [9] <http://www.tizag.com/phpT/phpsessions.php>

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α: Κώδικας

ΠΑΡΑΡΤΗΜΑ Β: Παρουσίαση

ΠΑΡΑΡΤΗΜΑ Γ: Περίληψη

ΠΑΡΑΡΤΗΜΑ Α: Κώδικας

Ο κώδικας για την δημιουργία της εφαρμογής *main_a.php*:

```

<?
require_once 'config_map.php';
$id= (int)$_GET[id];
    $query = "SELECT * FROM points WHERE id='$id'";
    $result = mysql_db_query($dbname, $query, $conn);
    $data = mysql_fetch_array($result);
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Πλοηγός Λασιθίου</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="style.css" rel="stylesheet" type="text/css" />

<script charset="utf-8" src="http://maps.google.com/maps?file=api&v=2&hl=el&oe=utf-
8&key=ABQIAAAA2AU5nm0q6DaKdvPrdaeviBT23T-fexH0xgSYxTBtL6X-
XnZ2QxQJqGg7BXKPT_3QBZAiKGvaz_TTJg"
type="text/javascript"></script>
<script type="text/javascript">

    var iconGreen = new GIcon();
    iconGreen.image = 'eikones/green.png';
    iconGreen.iconSize = new GSize(32, 32);
    iconGreen.iconAnchor = new GPoint(16, 16);
    iconGreen.infoWindowAnchor = new GPoint(5, 1);

    var iconRed = new GIcon();
    iconRed.image = 'eikones/red.png';
    iconRed.iconSize = new GSize(32, 32);
    iconRed.iconAnchor = new GPoint(16, 16);
    iconRed.infoWindowAnchor = new GPoint(5, 1);

    var iconBlue = new GIcon();
    iconBlue.image = 'eikones/blue.png';
    iconBlue.iconSize = new GSize(32, 32);
    iconBlue.iconAnchor = new GPoint(16, 16);
    iconBlue.infoWindowAnchor = new GPoint(5, 1);

    var customIcons = [];
    customIcons["sup"] = iconRed;
    customIcons["ven"] = iconBlue;
    customIcons["cin"] = iconGreen;
    var markerGroups = { "sup": [], "ven": [], "cin": []};

function load() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map"));
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
    }
}

```

```

var center = new GLatLng(35.17156, 25.91537);
map.setCenter(center, 10);
geocoder = new GClientGeocoder();
    map.setUIToDefault();
//map.enableGoogleBar(); //search

    document.getElementById("cineCheckbox").checked = true;
document.getElementById("supCheckbox").checked = true;
    document.getElementById("venCheckbox").checked = true;
document.getElementById("labelsCheckbox").checked = true;

GDownloadUrl("genxml.php", function(data) {
    var xml = GXml.parse(data);
    var markers = xml.documentElement.getElementsByTagName("marker");
    for (var i = 0; i < markers.length; i++) {
        var name = markers[i].getAttribute("title");
        var address = markers[i].getAttribute("address");
        var city = markers[i].getAttribute("city");
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
            parseFloat(markers[i].getAttribute("lng")));
        var category = markers[i].getAttribute("category");
        var marker = createMarker(point, name, address, city, category);
        map.addOverlay(marker);
    }
});
}
}

function createMarker(point, name, address, city, category) {
    var marker = new GMarker(point, customIcons[category]);
    markerGroups[category].push(marker);
    var html = "<b>" + name + "</b> <br/>" + address + "</b> <br/>" + city;
    GEvent.addListener(marker, 'click', function() {
        marker.openInfoWindowHtml(html);
    });
    return marker;
}

function toggleGroup(category) {
    for (var i = 0; i < markerGroups[category].length; i++) {
        var marker = markerGroups[category][i];
        if (marker.isHidden()) {
            marker.show();
        } else {
            marker.hide();
        }
    }
}

function toggleLabels() {
    var showLabels = document.getElementById("labelsCheckbox").checked;
    for (groupName in markerGroups) {
        for (var i = 0; i < markerGroups[groupName].length; i++) {
            var marker = markerGroups[groupName][i];
            marker.setLabelVisibility(showLabels);
        }
    }
}

```

```

    }
  }
}
</script>

</head>

<body onload="load()" onunload="GUnload()">

<table width="60%" height="100%" align="center">
<tr ><td width="100%" height="150" class="art-header-jpeg">
<h1 align="center" class="style1">Πλοηγός Λασιθίου </h1></td>
</tr>
<tr><td>
<table width="900" height="404">
<tr>
<td width="220" align="left" valign="top" >
<table width="220"><tr><td><p></p></td></tr><tr class="art-menu"><td><h1
class="style2">Σημεία Ενδιαφέροντος</h1></td></tr></table>
<input type="hidden" id="labelsCheckbox" onclick="toggleLabels()" CHECKED />
<table cellpadding="3" cellspacing="1" width="180">

  <tbody>

    <form name="checkboxes">
    <tr></tr>
    <tr>
      <td valign="top" width="10" align="left"><input id="cineCheckbox" type="checkbox"
onclick="toggleGroup('cin')"></td>
      <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Κινηματογράφοι</font></span></td>
    </tr>
    <tr>
      <td valign="top" width="10" align="left"><input id="supCheckbox" type="checkbox"
onclick="toggleGroup('sup')"></td>
      <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Σούπερ Μαρκετ</font></span></td>
    </tr>
    <tr>
      <td valign="top" width="10" align="left"><input id="venCheckbox" type="checkbox"
onclick="toggleGroup('ven')"></td>
      <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Πρατήρια Βενζίνης</font></span></td>
    </tr></form>
  </tbody>
</table>
</td>

<td>
<div id="map" align="center" style="width: 600px; height: 400px"></div>
</td>

<td width="100%"><table width="100%"><tr class="art-menu"><td><h1
class="style2">Αναζήτηση</h1></td></tr></table>

```

```

<form method="post" action="find.php">

  <p class="style5">Εισάγετε την τοποθεσία :
    <input name="address" id="address" type="text" size="34" />
    <input type="submit" class="style5" value="Αναζήτηση" />
  </p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>

</form>

<p align="left">

<table width="100%" border="0" align="center" cellpadding="0" cellspacing="1"
bordercolor="#FF0000">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bordercolor="#FF0000">
<tr>
<td colspan="3" class="art-menu" height="30px"><h1 class="style2">Είσοδος Μέλους</h1></td>
</tr>
<tr>
<td width="78" class="style5">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername"></td>
</tr>
<tr>
<td class="style5">Password</td>
<td>:</td>
<td><input name="mypassword" type="password" id="mypassword"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td></td>
<td><input type="submit" name="Submit" class="style5" value="Login"></td>
</tr>
<tr>
<td colspan="3"><a href="forma.php" class="style5">Register!</a></td>
</tr>
</table>
</td>
</form>
</tr></table>
</td>
</tr></table>
</td>
</tr>
<tr class="art-footer"><td></td></tr>
</table>
</body>
</html>

```

Ο κώδικας για την δημιουργία της εφαρμογής *main.php*:

```

<?php
session_start();
$lat= $_SESSION['lat'];
$lng= $_SESSION['lng'];
session_destroy();
?>
<?
require_once 'config_map.php';
$id= (int)$_GET[id];
$query = "SELECT * FROM points WHERE id='$id'";
$result = mysql_db_query($dbname, $query, $conn);
$data = mysql_fetch_array($result);
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Πλοηγός Λασιθίου</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="style.css" rel="stylesheet" type="text/css" />

<script charset="utf-8" src="http://maps.google.com/maps?file=api&v=2&hl=el&oe=utf-
8&key=ABQIAAAA2AU5nm0q6DaKdvPrdaeviBT23T-fexH0xgSYxTBtL6X-
XnZ2QxQJqGg7BXKPT_3QBZAiKGvaz_TTJg"
type="text/javascript"></script>
<script type="text/javascript">

var iconGreen = new GIcon();
iconGreen.image = 'eikones/green.png';
iconGreen.iconSize = new GSize(32, 32);
iconGreen.iconAnchor = new GPoint(16, 16);
iconGreen.infoWindowAnchor = new GPoint(5, 1);

var iconRed = new GIcon();
iconRed.image = 'eikones/red.png';
iconRed.iconSize = new GSize(32, 32);
iconRed.iconAnchor = new GPoint(16, 16);
iconRed.infoWindowAnchor = new GPoint(5, 1);

var iconBlue = new GIcon();
iconBlue.image = 'eikones/blue.png';
iconBlue.iconSize = new GSize(32, 32);
iconBlue.iconAnchor = new GPoint(16, 16);
iconBlue.infoWindowAnchor = new GPoint(5, 1);

var customIcons = [];
customIcons["sup"] = iconRed;
customIcons["ven"] = iconBlue;
customIcons["cin"] = iconGreen;
var markerGroups = { "sup": [], "ven": [], "cin": []};

```

```

function load() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    //var lat=35.17156;
    //var lng=25.91537;
    //var lat= $_SESSION['lat'];
    //var lng= $_SESSION['lng'];
    var lat= '<?php echo $lat; ?>';
    var lng= '<?php echo $lng; ?>';
    var center = new GLatLng(lat, lng);
    map.setCenter(center, 17);
    geocoder = new GClientGeocoder();
    map.setUIToDefault();
    //map.enableGoogleBar(); //search

    document.getElementById("cineCheckbox").checked = true;
    document.getElementById("supCheckbox").checked = true;
    document.getElementById("venCheckbox").checked = true;
    document.getElementById("labelsCheckbox").checked = true;

GDownloadUrl("genxml.php", function(data) {
  var xml = GXml.parse(data);
  var markers = xml.documentElement.getElementsByTagName("marker");
  for (var i = 0; i < markers.length; i++) {
    var name = markers[i].getAttribute("title");
    var address = markers[i].getAttribute("address");
    var city = markers[i].getAttribute("city");
    var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
      parseFloat(markers[i].getAttribute("lng")));
    var category = markers[i].getAttribute("category");
    var marker = createMarker(point, name, address, city, category);
    map.addOverlay(marker);
  }
});
}
}

function createMarker(point, name, address, city, category) {
  var marker = new GMarker(point, customIcons[category]);
  markerGroups[category].push(marker);
  var html = "<b>" + name + "</b> <br/>" + address + "<b> <br/>" + city;
  GEvent.addListener(marker, 'click', function() {
    marker.openInfoWindowHtml(html);
  });
  return marker;
}

function toggleGroup(category) {
  for (var i = 0; i < markerGroups[category].length; i++) {
    var marker = markerGroups[category][i];
    if (marker.isHidden()) {
      marker.show();
    } else {

```

```

    marker.hide();
  }
}
}

```

```

function toggleLabels() {
  var showLabels = document.getElementById("labelsCheckbox").checked;
  for (groupName in markerGroups) {
    for (var i = 0; i < markerGroups[groupName].length; i++) {
      var marker = markerGroups[groupName][i];
      marker.setLabelVisibility(showLabels);
    }
  }
}
}

```

```
</script>
```

```
</head>
```

```
<body onload="load()" onunload="GUnload()">
```

```

<table width="60%" height="100%" align="center">
<tr ><td width="100%" height="150" class="art-header-jpeg">
<h1 align="center" class="style1">Πλοηγός Λασιθίου </h1></td>
</tr>
<tr><td>
<table width="900" height="404">
<tr>
<td width="220" align="left" valign="top" >
<table width="220"><tr><td><p></p></td></tr><tr class="art-menu"><td><h1
class="style2">Σημεία Ενδιαφέροντος</h1></td></tr></table>
<input type="hidden" id="labelsCheckbox" onclick="toggleLabels()" CHECKED />
<table cellpadding="3" cellspacing="1" width="180">

```

```
<tbody>
```

```

    <form name="checkboxes">
    <tr></tr>
    <tr>
    <td valign="top" width="10" align="left"><input id="cineCheckbox" type="checkbox"
onclick="toggleGroup('cin')"></td>
    <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Κινηματογράφοι</font></span></td>
    </tr>
    <tr>
    <td valign="top" width="10" align="left"><input id="supCheckbox" type="checkbox"
onclick="toggleGroup('sup')"></td>
    <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Σούπερ Μαρκετ</font></span></td>
    </tr>
    <tr>
    <td valign="top" width="10" align="left"><input id="venCheckbox" type="checkbox"
onclick="toggleGroup('ven')"></td>

```



```

        <td align="left" valign="middle" bordercolor="#ECE9D8"><span
class="style5"><font>Πρατήρια Βενζίνης</font></span></td>
                </tr></form>
        </tbody>
</table>
</td>

<td>
<div id="map" align="center" style="width: 600px; height: 400px"></div>
</td>

<td width="100%"><table width="100%"><tr class="art-menu"><td><h1
class="style2">Αναζήτηση</h1></td></tr></table>

<form method="post" action="find.php">

        <p class="style5">Εισάγετε την τοποθεσία :
                <input name="address" id="address" type="text" size="34" />
                <input type="submit" class="style5" value="Αναζήτηση" />
        </p>
        <p>&nbsp;</p>
        <p>&nbsp;</p>

</form>

<p align="left">

<table width="100%" border="0" align="center" cellpadding="0" cellspacing="1"
bordercolor="#FF0000">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bordercolor="#FF0000">
<tr>
<td colspan="3" class="art-menu" height="30px"><h1 class="style2">Είσοδος Μέλους</h1></td>
</tr>
<tr>
<td width="78" class="style5">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername"></td>
</tr>
<tr>
<td class="style5">Password</td>
<td>:</td>
<td><input name="mypassword" type="password" id="mypassword"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td></td>
<td><input type="submit" name="Submit" class="style5" value="Login"></td>
</tr>
<tr>
<td colspan="3"><a href="forma.php" class="style5">Register!</a> </td>
</tr>
</table>

```

```

</td>
</form>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
<tr class="art-footer"><td></td></tr>
</table>
</body>
</html>

```

Ο κώδικας για την δημιουργία της εφαρμογής *config_syst_kath.php*:

```

<?php
$dbname = 'syst_kathodigisis';// onoma vashs
$dbhost = 'localhost';// host
$dbuser = 'dimitra';// username xrhsth vashs
$dbpass = '1234';// password xrhsth vashs

$conn = mysql_connect($dbhost, $dbuser, $dbpass);
$db = mysql_select_db($dbname,$conn);
mysql_query(' set character set utf8 ');
mysql_query("SET NAMES 'utf8' ");
?>

```

Ο κώδικας για την δημιουργία της εφαρμογής *config_map.php*:

```

<?php
$dbname = 'maps';// onoma vashs
$dbhost = 'localhost';// host
$dbuser = 'dimitra';// username xrhsth vashs
$dbpass = '1234';// password xrhsth vashs

$google_api ='ABQIAAAA2AU5nm0q6DaKdvPrdaeviBT23T-fexH0xgSYxTBtL6X-
XnZ2QxQJqGg7BXXPT_3QBZAiKGVaz_TTJg';//kodikos API apo thn google

$conn = mysql_connect($dbhost, $dbuser, $dbpass);
$db = mysql_select_db($dbname,$conn);
mysql_query(' set character set utf8 ');
mysql_query("SET NAMES 'utf8' ");
?>

```

Ο κώδικας για την δημιουργία της εφαρμογής *forma.php*:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Εγγραφή χρήστη</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="JavaScript" type="text/JavaScript">

```

```

<!--
function MM_findObj(n, d) { //v4.01
  var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p); }
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_validateForm() { //v4.0
  var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
  for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
      if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
        if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an e-mail address.\n';
      } else if (test!='R') { num = parseFloat(val);
        if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
        if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
          min=test.substring(8,p); max=test.substring(p+1);
          if (num<min || max<num) errors+='- '+nm+' must contain a number between '+min+' and '+max+'.\n';
        } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is required.\n'; }
    } if (errors) alert("The following error(s) occurred:\n"+errors);
    document.MM_returnValue = (errors == "");
  }
}
//-->
</script>
</head>

<body>
<table width="60%" height="100%" align="center">
  <tr <td width="100%" height="150" class="art-header-jpeg">
    <h1 align="center" class="style1">Πλοηγός Λασιθίου </h1></td>
  </tr>
  <tr><td>
    <table align="center" width="50%" height="420"><td>
      <form method="post" action="insert.php">
      <table align="center">
        <tr height="30"><td>Όνομα</td> <td>: </td> <td><input type="text" name="onoma"
          align="right"></td></tr>
        <tr height="30"><td>Επίθετο</td><td>: </td> <td><input type="text" name="epitheto"
          align="right"></td></tr>
        <tr height="30"><td>E-mail</td><td>: </td><td><input type="text" name="e-mail"
          align="right"></td></tr>
        <tr height="30"><td>Τηλέφωνο</td> <td>: </td><td><input type="text" name="tilefono"
          align="right"></td></tr>
        <tr height="30"><td>Διεύθυνση</td><td>: </td></tr>
        <tr height="30"><td>Περιοχή</td><td>: </td><td><input type="text" name="perioxi"
          align="right"></td></tr>
        <tr height="30"><td>Οδός</td><td>: </td><td><input type="text" name="odos"
          align="right"></td></tr>
        <tr height="30"><td>Αριθμός</td> <td>: </td><td><input type="text" name="arithmos"
          align="right"></td></tr>
        <tr height="30"><td>Αριθμός πιστωτικής</td><td>: </td><td><input type="text"
          name="ar_pistotikis" align="right"></td></tr>

```

```

<tr height="30"><td>Username</td><td>: </td><td><input type="text" name="username"
align="right"></td></tr>
<tr height="30"><td>Password</td><td>: </td><td><input type="password" name="password"
align="right"></td></tr>
<tr height="30"><td colspan="3" align="center"><input name="submit" type="submit"
onClick="MM_validateForm('onoma','R','epitheto','R','e-
mail','RisEmail','tilefono','NisNum','arithmos','NisNum','ar_pistotikis','RisNum','username','R','
password','R');return document.MM_returnValue" value="Υποβολή" align="right"></td></tr>
</table>
</form>
</td></table>
</td></tr>
<tr class="art-footer"><td></td></tr>
</table>
</body>
</html>

```

Ο κώδικας για την δημιουργία της εφαρμογής *insert.php*:

```

<?php
require("config_syst_kath.php");
// Opens a connection to a MySQL server
$connection=mysql_connect ($dbhost, $dbuser, $dbpass);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($dbname, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// Get values from form
$onoma=$_POST['onoma'];
$epitheto=$_POST['epitheto'];
$ar_pistotikis=$_POST['ar_pistotikis'];
$email=$_POST['e-mail'];
$tilefono=$_POST['tilefono'];
$perioxi=$_POST['perioxi'];
$sodos=$_POST['odos'];
$arithmos=$_POST['arithmos'];
$username=$_POST['username'];
$password=$_POST['password'];

$result = mysql_query("SELECT username FROM users");

while($row = mysql_fetch_assoc($result))
{
    if($username==$row['username'])
        echo "Username : {$row['name']} is not available<br>" ;
}

```

```

if (!$onoma || !$epitheto || !$ar_pistotikis || !$email || !$tilefono || !$perioxi || !$odos || !$arithmos ||
!$username || !$password)
{
echo 'You have not entered all the required details.<br/>';
echo "<a href='forma.php'>Please go back and try again.</a>";
exit;
}
if (!get_magic_quotes_gpc())
{

}
// Insert data into mysql
$sql = "INSERT INTO users (onoma, epitheto, ar_pistotikis, e-mail, tilefono, perioxi, odos, arithmos,
username, password)VALUES('".$onoma."', '".$epitheto."', '".$ar_pistotikis."', '".$email."',
'".$tilefono."', '".$perioxi."', '".$odos."', '".$arithmos."', '".$username."', '".$password."')";
$result=mysql_query($sql);

// if successfully insert data into database, displays message "Successful".
if($result){
echo "Successful";
echo "<BR>";
echo "<a href='main_a.php'>Back to main page</a>";
}

else {
echo "ERROR";
}

// close connection
mysql_close();
?>

```

Ο κώδικας για την δημιουργία της εφαρμογής *genxml.php*:

```

<?php
require("config_map.php");

function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('","','&quot;',$xmlStr);
$xmlStr=str_replace('","','&#39;',$xmlStr);
$xmlStr=str_replace("&","&amp;",$xmlStr);
return $xmlStr;
}

// Opens a connection to a MySQL server
$connection=mysql_connect ($dbhost, $dbuser, $dbpass);
if (!$connection) {
die('Not connected : ' . mysql_error());
}

// Set the active MySQL database

```

```

$db_selected = mysql_select_db($dbname, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM points WHERE 1";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// Start XML file, echo parent node
echo '<markers>';

// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ' ;
    echo 'title="' . parseToXML($row['title']) . '" ' ;
    echo 'address="' . parseToXML($row['address']) . '" ' ;
    echo 'lat="' . $row['lat'] . '" ' ;
    echo 'lng="' . $row['lng'] . '" ' ;
    echo 'city="' . $row['city'] . '" ' ;
    echo 'category="' . $row['category'] . '" ' ;
    echo '/>';
}

// End XML file
echo '</markers>';
?>

```

Ο κώδικας για την δημιουργία της εφαρμογής *find.php*:

```

<?php
session_start();
require("config_map.php");

// Connect to server and select database.
mysql_connect("$dbhost", "$dbuser", "$dbpass")or die("cannot connect");
mysql_select_db("$dbname")or die("cannot select DB");

// address sent from form
$address=$_POST["address"];

// To protect MySQL injection (more detail about MySQL injection)
$address = stripslashes($address);
$address = mysql_real_escape_string($address);

$sql="SELECT * FROM points WHERE address='".$address.'";

$result=mysql_query($sql) or die(mysql_error());

```

```
$row = mysql_fetch_array( $result );

if ($row){

    $lat=$row['lat'];
    $lng=$row['lng'];
    $_SESSION['lat'] = $lat;
    $_SESSION['lng'] = $lng;

    header("location:main.php");
}
else
echo "Address not found!";
echo '<a href="main_a.php"> Back to main page</a>';
?>
```

Ο κώδικας για την δημιουργία της εφαρμογής *login_success.php*:

```
// Check if session is not registered , redirect back to main page.
```

```
<?php
session_start();
if(!isset($_SESSION[$myusername])){
echo 'Login Successful';
header("location:main_a.php");
}
?>
```

Ο κώδικας για την δημιουργία της εφαρμογής *checklogin.php*:

```
<?php
session_start();
require("config_syst_kath.php");

// Connect to server and select databse.
mysql_connect("$dbhost", "$dbuser", "$dbpass")or die("cannot connect");
mysql_select_db("$dbname")or die("cannot select DB");

// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

$encrypted_mypassword=md5($mypassword);

$sql="SELECT * FROM users WHERE username='".$myusername.'" and
password='".$mypassword.'";

$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
```

```
if($count==1){

// Register $myusername, $mypassword and redirect to file "login_success.php"
session_register("myusername");
session_register("mypassword");

header("location:login_success.php");
}
else {
echo "Wrong Username or Password! ";
echo '<a href="main_a.php">Go back to main page to register</a>';
}
?>
```

Ο κώδικας για την δημιουργία της εφαρμογής *style.css*:

```
@charset "iso-8859-1";
/* CSS Document */

.style1 {
    font-size: 38px;
    font-family: Tahoma, Arial, Helvetica, sans-serif;
    font-weight: inherit;
    color: #03B7ED;
    text-align: justify;
    position: static;
}
.style2 {
    font-size: 16px;
    font-family: Verdana, Geneva, Arial, Helvetica, Sans-Serif;
    color: #03B7ED;
    text-align: center;
}
.art-header-jpeg
{
    position: inherit;
    z-index: -1;
    top: 0;
    left: 0;
    width: 900px;
    height: 150px;
    background-image: url('eikones/header.jpg');
    background-repeat: repeat-x;
    background-position: center;
    font-size: 24px;
}
.art-footer
{
    position: inherit;
    z-index: -1;
    background-repeat: repeat-x;
    background-image: url('eikones/footer.png');
    background-position:center;
```



```
width: 900px;
height: 40px;
bottom: 0;
left: 0;
}
.style4 {
font-family: Verdana, Geneva, Arial, Helvetica, Sans-Serif;
font-size: 14px;
color: #99CC00;
}
.style5 {
font-family: Verdana, Geneva, Arial, Helvetica, Sans-Serif;
font-size: 14px;
}
.art-menu
{
position: static;
z-index: -1;
background-repeat: repeat-x;
background-image: url('eikones/menu.jpg');
background-position: top;
width: 180px;
height: 30px;
bottom: 0;
left: 0;
}
```

ΠΑΡΑΡΤΗΜΑ Β: Παρουσίαση



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πληροφοριακό σύστημα για την καθοδήγηση χρηστών με χρήση **GIS**

Μανδουράρη Δήμητρα AM: 2178

Εισαγωγή

○ Στόχος

Κατασκευή δυναμικής ιστοσελίδας με χάρτη για την καθοδήγηση των χρηστών

○ Εργαλεία Υλοποίησης

- Γλώσσες προγραμματισμού: Html, Javascript, PHP, MySQL, CSS
- Πακέτο λογισμικού EasyPHP
- Macromedia Dreamweaver



EasyPHP

Περιλαμβάνει

- ένα διακομιστή Apache
- μια βάση δεδομένων MySQL
- την επέκταση της PHP
- εργαλεία εύκολης ανάπτυξης



Γεωγραφικά Συστήματα Πληροφοριών (GIS)

- ένα εργαλείο "έξυπνου χάρτη"
- επιτρέπει στους χρήστες του
 - να αποτυπώσουν μια περίληψη του πραγματικού κόσμου
 - να δημιουργήσουν διαδραστικά ερωτήσεις χωρικού ή περιγραφικού χαρακτήρα
 - να αναλύσουν τα χωρικά δεδομένα
 - να τα προσαρμόσουν και να τα αποδώσουν σε αναλογικά μέσα ή σε ψηφιακά μέσα



Google Maps API

Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface – API)

- δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη σε ιστοσελίδες με τη χρήση Javascript
- δωρεάν υπηρεσία



Google Maps API

○ Events

(κλικ του ποντικιού, διπλό κλικ, κίνηση του ποντικιού -> Javascript καθοδηγεί τα events)

○ Controls

(π.χ GLargeMapControl3D: τρισδιάστατο κουμπί ελέγχου εστίασης και κίνησης του χάρτη)

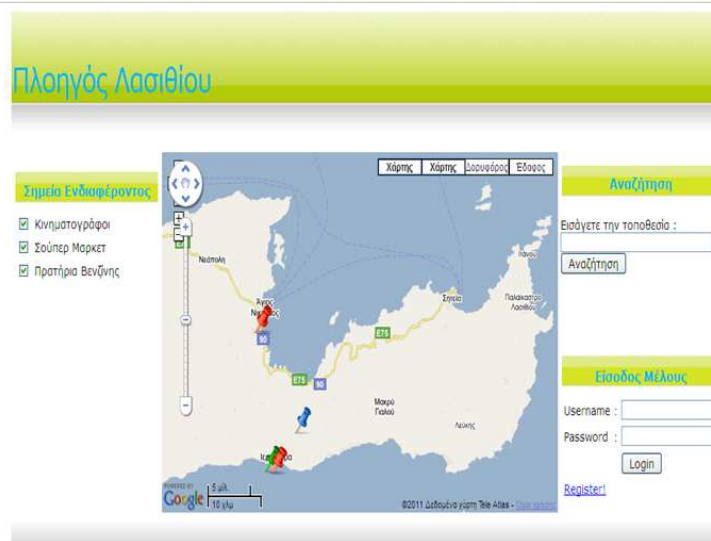
○ Overlays

(σημεία, γραμμές, περιοχές, χάρτης , info windows)

Βάσεις Δεδομένων

- Η βάση *maps*
Δεδομένα που αφορούν το χάρτη
(σημεία, διευθύνσεις...)
- Η βάση *syst_kathodigisis*
Δεδομένα που αφορούν τους χρήστες
(όνομα, επίθετο, username, password...)
και τα σημεία ενδιαφέροντος
(επωνυμία, ώρες λειτουργίας...)

Η αρχική σελίδα



Φόρμα συμπλήρωσης μέλους

Πλοηγός Λασιθίου

Όνομα :

Επίθετο :

E-mail :

Τηλέφωνο :

Διεύθυνση :

Περιοχή :

Οδός :

Αριθμός :

Αριθμός πιστωτικής :

Username :

Password :

Παράδειγμα αναζήτησης

Πλοηγός Λασιθίου

Σημεία Ενδιαφέροντος

- Κινηματογράφοι
- Σούπερ Μάρκετ
- Πρατήρια Βενζίνης

Αναζήτηση

Εισάγετε την τοποθεσία :

Είσοδος Μέλους

Username :

Password :

[Register!](#)



Συμπεράσματα

- Η ανάπτυξη ενός πληροφοριακού συστήματος (GIS) με σκοπό την καθοδήγηση
- Το σύγγραμμα της πτυχιακής αποτελεί ένα εκπαιδευτικό βοήθημα για τον αναγνώστη, διότι αναπτύσσονται όλα τα θεωρητικά θέματα και τα εργαλεία που χρησιμοποιήθηκαν και άπτονται του αντικειμένου της πτυχιακής.

ΠΑΡΑΡΤΗΜΑ Γ: Περίληψη

Πληροφοριακό σύστημα για την καθοδήγηση χρηστών με χρήση GIS

Η ταχύτητα αναπτυσσόμενη τεχνολογία στις μέρες μας, μας παρέχει τη δυνατότητα μετάδοσης της πληροφορίας χωρίς ιδιαίτερο κόστος. Σκοπός της πτυχιακής αυτής ήταν ο σχεδιασμός και η υλοποίηση ενός πληροφοριακού συστήματος με το οποίο θα γίνεται η καθοδήγηση των χρηστών του ιστοτόπου. Πιο συγκεκριμένα, η καθοδήγηση αφορά περιοχές που ανήκουν στο νομό Λασιθίου.

Τα Γεωγραφικά Συστήματα Πληροφοριών γνωστά ευρέως και ως GIS (Geographic Information System), είναι συστήματα διαχείρισης χωρικών δεδομένων (spatial data) και συσχετισμένων ιδιοτήτων. Ένα GIS στην πιο αυστηρή μορφή του είναι ένα ψηφιακό σύστημα, ικανό να ενσωματώσει, αποθηκεύσει, προσαρμόσει, αναλύσει και παρουσιάσει γεωγραφικά συσχετισμένες (geographically referenced) πληροφορίες. Σε πιο γενική μορφή, ένα ΓΣΠ είναι ένα εργαλείο "έξυπνου χάρτη", το οποίο επιτρέπει στους χρήστες του :

- να αποτυπώσουν μια περίληψη του πραγματικού κόσμου
- να δημιουργήσουν διαδραστικά ερωτήσεις χωρικού ή περιγραφικού χαρακτήρα (αναζητήσεις δημιουργούμενες από τον χρήστη)
- να αναλύσουν τα χωρικά δεδομένα (spatial data)
- να τα προσαρμόσουν και να τα αποδώσουν σε αναλογικά μέσα (εκτυπώσεις χαρτών και διαγραμμάτων) ή σε ψηφιακά μέσα (αρχεία χωρικών δεδομένων, διαδραστικοί χάρτες στο Διαδίκτυο).

Για την υλοποίηση του συστήματος χρησιμοποιήθηκαν γλώσσες προγραμματισμού όπως η Javascript, η HTML, η CSS, η PHP και η MySQL, καθώς και το πακέτο λογισμικού EasyPHP, που υποστηρίζει τις δυο τελευταίες. Επίσης χρησιμοποιήθηκε η υπηρεσία Google Maps API, μία διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface - API) της Google μέσω της οποίας δόθηκε η δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη στην ιστοσελίδα.

Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορούμε να κάνουμε οτιδήποτε μπορεί ένα άλλο CGI πρόγραμμα να κάνει, όπως η συλλογή δεδομένων, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στέλνουμε και να λαμβάνουμε cookies, αλλά και πολύ περισσότερα. Η MySQL είναι ένας SQL Database Server. Ένας server είναι ένα κομμάτι λογισμικού (software) που τρέχει (εκτελείται) στο παρασκήνιο σ' έναν υπολογιστή. Μια βάση δεδομένων (database) είναι μια δομημένη συλλογή από δεδομένα. Μια βάση δεδομένων είναι έτσι δομημένη ώστε να μπορούμε να βρίσκουμε πληροφορίες εύκολα και γρήγορα ανάλογα με το είδος των δεδομένων που περιέχει. Η EasyPHP είναι ένα πλήρες πακέτο λογισμικού για τα Windows που επιτρέπει την ανάπτυξη PHP και MySQL στον προσωπικό υπολογιστή του χρήστη. Περιλαμβάνει ένα διακομιστή Apache, μια βάση δεδομένων MySQL, την επέκταση της PHP καθώς και εργαλεία εύκολης ανάπτυξης για το web site ή τις εφαρμογές μας.

Με τη JavaScript μπορούμε να φτιάξουμε σενάρια που να εκτελούν αυτόματες εργασίες, για παράδειγμα όταν μια σελίδα του Web ανοίγει ή κλείνει. Επίσης μπορούμε να κάνουμε την JavaScript να εκτελεί ενέργειες ανταποκρινόμενη σε ένα συγκεκριμένο γεγονός. Για παράδειγμα όταν ο χρήστης επιλέγει ένα κουμπί ή ένα σύνδεσμο, όταν εστιάζει από ένα στοιχείο μιας φόρμας σε ένα άλλο στοιχείο της . Τα σενάρια μπορεί να ανοίγουν νέα παράθυρα στον browser και να εμφανίζουν συγκεκριμένα HTML έγγραφα ή να παρουσιάζουν μια σελίδα επιλεγμένη από τον κατάλογο ιστορικού του browser. Μπορεί επίσης να είναι και περίπλοκες δηλαδή ένα σενάριο μπορεί να ελέγχει τα περιεχόμενα μιας φόρμας που θέλει να υποβάλει ένας χρήστης και στη συνέχεια να προειδοποιεί τον χρήστη αν τα δεδομένα είναι λάθος. Το σενάριο μπορεί να ψάξει για πληροφορίες σε μια μικρή βάση δεδομένων ή να κάνει πολύπλοκους υπολογισμούς οικονομικών στοιχείων.

Η HTML είναι μια γλώσσα προγραμματισμού που χρησιμοποιεί ένα αριθμό από tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα των σελίδα, για την εισαγωγή εικόνων, ήχου κ.α. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας. Επίσης η CSS είναι μια markup γλώσσα η οποία

περιγράφει τον τρόπο απεικόνισης εγγράφων που έχουν γραφεί σε markup γλώσσες όπως η HTML. Με την επισύναψη αρχείων CSS σε δομημένα έγγραφα (HTML, XML κλπ) ο χρήστης ή ο αναγνώστης έχει τη δυνατότητα να τροποποιήσει την παρουσίαση του εγγράφου χωρίς να κάνει αλλαγές στη δομή του, δηλαδή να προσθέσει κάποια ετικέτα κλπ. Άλλος ένας σημαντικός λόγος για τη χρήση της γλώσσας αυτής στο διαδίκτυο είναι ο διαχωρισμός του περιεχομένου από την εμφάνιση του εγγράφου, δίνοντας έτσι ένα πλεονέκτημα σε θέματα που αφορούν την πρόσβαση στο περιεχόμενο, την καλύτερη δόμηση του κώδικα, θέματα βελτιστοποίησης κλπ.

Το Google Maps API είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη σε ιστοσελίδες με τη χρήση Javascript. Το API παρέχει έναν αριθμό εργαλείων για το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες.

Βασικό στοιχείο οποιασδήποτε Google Maps API εφαρμογής είναι ο ίδιος ο χάρτης, ο οποίος φορτώνεται και αρχικοποιείται με συγκεκριμένες διαστάσεις (ύψος και πλάτος), κέντρο (γεωγραφικό πλάτος, γεωγραφικό μήκος), επίπεδο εστίασης και τύπο (κανονικό, δορυφορικό, υβριδικό). Στη συνέχεια είναι δυνατή η αλληλεπίδραση με το χάρτη μέσω διαφόρων μηχανισμών όπως είναι τα events, τα controls, τα overlays αλλά και συναρτήσεις επεξεργασίας των χαρακτηριστικών του.

Στην περίπτωση του ιστότοπου που δημιουργήθηκε, ο χάρτης φορτώνεται με όλες τις παραμέτρους που έχουν δοθεί στον κώδικα καθώς επίσης και τα markers σε διάφορα σημεία του. Με τη βοήθεια των checkboxes ο χρήστης μπορεί να επιλέξει ποία σημεία θέλει να εμφανίζονται στο χάρτη (όπως πρατήρια καυσίμων, σούπερ μάρκετ, ή κινηματογράφους). Επιπλέον ο χρήστης έχει τη δυνατότητα να εισάγει μία διεύθυνση στο πεδίο αναζήτησης και ο χάρτης να μεγεθύνεται στο σημείο αυτό βοηθώντας τον να σχηματίσει μια εικόνα για την περιοχή όπου βρίσκεται το σημείο αυτό. Ένα άλλο χαρακτηριστικό του ιστότοπου είναι η δυνατότητα εγγραφής του χρήστη.

Τέλος, το Σύστημα που δημιουργήθηκε είναι δυναμικό και παρέχει μεγάλες δυνατότητες ανάπτυξης και επέκτασης. Εκτός από τα δεδομένα, το σύστημα είναι ανοιχτό και στην προσθήκη νέων λειτουργιών. Ένα τέτοιο παράδειγμα θα μπορούσε να ήταν η προσθήκη της δυνατότητας εμφάνισης της απόστασης μεταξύ δυο σημείων ή η αναζήτηση βέλτιστης διαδρομής ή ακόμη και η διασύνδεση της εφαρμογής απευθείας σε κινητές συσκευές.