

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

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



Πτυχιακή εργασία

**Τίτλος: Τουριστική Παρουσίαση της Λευκάδος
μέσο του Google Earth**

Ευστάθιος Κατωπόδης (ΑΜ: 1197)

Ηράκλειο - Ημερομηνία

Επόπτης Καθηγητής: Δρ. Μαλάμος Αθανάσιος

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή της εργασίας μου κύριο Μαλάμο Αθανάσιο που μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο αντικείμενο καθώς και για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία .

Επίσης, θα ήθελα να αφιερώσω αυτήν την εργασία στην οικογένεια μου για την στήριξη της,υλική και πνευματική για τη διεκπεραίωση των σπουδών μου .

Abstract

The purpose of this thesis is to design and implement an integrated system for dynamic management of the wider area of the tourist island of Lefkada. This thesis aims at using modern Web 2.0. technologies to create a web application, which means he can collect information on accommodation, entertainment, beaches, attractions and everything else could help him in organizing his vacations.

The display points of interest will be to map dynamically. This application uses the application programming interface of Google, Google Maps API, which provide a number of tools for viewing and manipulation of maps. Thus, the user can select a visual way around the map as it wishes.

Specifically:

- It can manage the data of each point (marker) average database SQL.
- The data point will be stored in a SQL. That is, the administrator can change data such as geographic location (longitude latitude) and may delete some point.
- The file will update the map based on the data point (longitude latitude)
- Ability to view the map using technology GOOGLE MAP API
- Read a SQL file
- Dynamic allocation and placement of objects on the map.
- Presentation of data. Display of signs.
- Signs displayed dynamically. It can check each time it is activated (checkbox) and displays the appropriate spot on the map.
- Give the final result on website

Σύνοψη

Αντικείμενο της συγκεκριμένης πτυχιακής εργασίας αποτελεί η σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος δυναμικής διαχείρισης της ευρύτερης τουριστικής περιοχής της νήσου Λευκάδας. Η πτυχιακή εργασία έχει σαν στόχο τη χρήση σύγχρονων Web 2.0. τεχνολογιών για τη δημιουργία μίας διαδικτυακής εφαρμογής, μέσω της οποίας ο ενδιαφερόμενος θα μπορεί να συλλέξει πληροφορίες σχετικά με χώρους διαμονής, διασκέδασης, παραλίες, αξιοθέατα και ό,τι άλλο θα μπορούσε να τον βοηθήσει στην οργάνωση των διακοπών του.

Η απεικόνιση των σημείων ενδιαφέροντος γίνονται σε χάρτη δυναμικά. Η εφαρμογή αυτή χρησιμοποιεί τη διεπαφή προγραμματισμού εφαρμογών της Google, Google Maps API, οι οποίες παρέχουν έναν αριθμό εργαλείων για την προβολή και το χειρισμό χαρτών. Έτσι, ο χρήστης μπορεί να επιλέξει με visual τρόπο την περιοχή του χάρτη που επιθυμεί.

Συγκεκριμένα:

- Δίνει δυνατότητα διαχείρισης των δεδομένων κάθε σημείου (marker) μέσω βάσης δεδομένων SQL.
- Τα δεδομένα του σημείου θα αποθηκεύονται σε αρχείο SQL. Δηλαδή, ο διαχειριστής θα μπορεί να μεταβάλει δεδομένα, όπως το γεωγραφικό σημείο (longitude latitude) και θα μπορεί να διαγράψει κάποιο σημείο.
- Το αρχείο θα ενημερώνει το χάρτη με βάση τα στοιχεία του σημείου (longitude latitude)
- Δυνατότητα προβολής του χάρτη με την χρήση τεχνολογίας GOOGLE MAP API
 - Ανάγνωση SQL αρχείου
 - Δυναμική κατανομή και τοποθέτηση αντικειμένων στο χάρτη.
 - Παρουσίαση των δεδομένων. Απεικόνιση των σημείων.
 - Τα σημεία απεικονίζονται δυναμικά. Θα μπορεί να ελέγχει κάθε φορά αυτό που είναι ενεργοποιημένο (checkbox) και θα εμφανίζει το ανάλογο σημείο πάνω στο χάρτη.
- Δίνει το τελικό αποτέλεσμα σε ιστοχώρο (web site).

Πίνακας Περιεχομένων

Ευχαριστίες.....	2
Abstract.....	3
Σύνοψη.....	4
Πίνακας Περιεχομένων.....	5
Πίνακας Εικόνων.....	7
ΚΕΦΑΛΑΙΟ 1.....	8
1.1 Περίληψη.....	8
1.2 Κίνητρο για τη διεξαγωγή της εργασίας.....	8
1.4 Δομή εργασίας.....	9
ΚΕΦΑΛΑΙΟ 2.....	10
Μεθοδολογία Υλοποίησης.....	10
2.1 Σχέδιο δράσης για την εκπόνηση της εργασία.....	10
2.2 Γλώσσες προγραμματισμού.....	10
2.2.1 HTML.....	10
2.2.1.1 DOCTYPE.....	11
2.2.1.1.1 HTML 4.01 Strict.....	11
2.2.1.1.2 Transitional.....	11
2.2.2 CSS.....	12
2.2.2.1 Εισαγωγή ορισμάτων CSS.....	12
2.2.2.2 Κλάσεις CSS.....	12
2.2.2.3 JavaScript.....	13
2.2.2.3.1 Ιεραρχία JavaScript.....	13
2.2.3 PHP.....	13
2.2.3.1 Ιστορία της PHP.....	14
2.2.3.2 Τι είναι η PHP.....	14
2.2.3.3 Πλεονεκτήματα της PHP.....	14
2.2.3.4 Που Χρησιμοποιείται ένα PHP script.....	15
2.2.3.5 Τι μπορεί να κάνει η PHP.....	16
2.2.3.6 Συναρτήσεις.....	16
2.2.3.6.1 GET.....	16
2.2.3.6.2 POST.....	16
2.2.4 SQL (Structured Query Language).....	16
2.2.4.1 Η ιστορία της SQL.....	16
2.2.4.2 DML (Data Manipulation Language) DDL (Data Definition Language) DCL (Data Control Language).....	17
2.2.4.3 DML Γλώσσα Διαχείρισης Δεδομένων.....	17
ΚΕΦΑΛΑΙΟ 3.....	19
Ανάλυση προβλήματος και εγκατάσταση Wamp server.....	19
3.1 Ανάλυση προβλήματος.....	19
3.2 Απαιτήσεις συστήματος.....	19
3.3 Σχεδιασμός υλοποίησης.....	19
3.4 Τεχνολογία.....	19
3.4.1 Εγκατάσταση Wampp.....	19
3.4.2 PhpMyAdmin.....	22
3.4.3 Τι είναι το WAMP.....	25
ΚΕΦΑΛΑΙΟ 4.....	26
4.1 Χρήση του Google Map Api σε διαδικτυακές εφαρμογές.....	26

4.1.1 Διεπαφή Προγραμματισμού Εφαρμογών Google Maps	26
4.1.1.1. The GMap class.....	27
4.1.1.2 Γεγονότα (Events)	27
4.1.1.3 Ρυθμιστές (Controls)	28
4.1.1.4 Overlays	28
4.1.1.5 The Info Window	30
4.1.1.6 Προσφερόμενες υπηρεσίες του Google Maps API	30
4.2 Δικαιώματα χρήσης	30
ΚΕΦΑΛΑΙΟ 5	32
5.1 Γεωγραφικά συστήματα πληροφοριών GIS (Geographical Information Systems) .	32
5.2 Ιστορία των GIS	32
5.3 Κατηγορίες GIS.....	33
5.4 Χρήσεις των GIS	33
5.6 Συμπέρασμα.....	35
ΚΕΦΑΛΑΙΟ 6	36
6.1 Πλοήγηση στην εφαρμογή.....	36
6.1.1 Αρχική οθόνη.....	36
6.1.2 Ενδιαφέροντα σημεία(ομάδες και υποομάδες).....	37
6.1.3 Δυνατότητες Εφαρμογής	37
6.1.4 Δενδροειδή δομή	40
6.1.5 Επιλογή ενδιαφέροντος σημείου.....	41
6.1.6 Εκτύπωση χαρτών	41
ΚΕΦΑΛΑΙΟ 7	42
7.1 Υλοποίηση Εφαρμογής.....	42
7.2 Περιγραφή – Κατασκευή βάσης δεδομένων της διεπιφάνειας.....	42
7.3 Κώδικας Εφαρμογής	45
7.3.1 Το αρχείο marker_functions3.js	45
7.3.2 Το αρχείο fortwse_all_markers.php.....	46
7.3.3 Το αρχείο vres_nodes.php που επικοινωνεί με τη βάση δεδομένων και μας επιστρέφει τις υποκατηγορίες για την επιλεγμένη κύρια κατηγορία.....	47
7.3.4 Το αρχείο load_tree.php που επικοινωνεί με τη βάση δεδομένων και μας επιστρέφει την δενδροειδή δομή που θέλουμε σε κατηγορίες/ υποκατηγορίες σε xml format.	48
7.3.5 Η σελίδα με τη διεπιφάνεια, leukada_markers.html.....	49
7.3.6 Κώδικας SQL για κατασκευή βάσης με δυο πίνακες για την εφαρμογή.....	51
ΚΕΦΑΛΑΙΟ 8 – ΕΠΙΛΟΓΟΣ	53
8.1 Αποτελέσματα	53
8.2 Συμπεράσματα.....	53
8.3 Εξέλιξη Συστήματος	53
8.4 Πρακτική αξιοποίηση	54
Βιβλιογραφία	55
Παράρτημα – Παρουσίαση	56

Πίνακας Εικόνων

Εικόνα : 1 Setup Wizard	20
Εικόνα : 2 accept the ageement.....	20
Εικόνα : 3 Finish Setup.....	20
Εικόνα : 4 Start WampServer	21
Εικόνα : 5 Localhost & phpMyAdmin.....	21
Εικόνα : 6 Wamp Server Home Page (επιλογή localhost).....	21
Εικόνα : 7 Δημιουργία νέας βάσης	22
Εικόνα : 8 Εισαγωγή δεδομένων στο πίνακα hot_spot.....	23
Εικόνα : 9 Εισαγωγή δεδομένων στο πίνακα spot_categories	23
Εικόνα : 10 Σχέδιο βάσης δεδομένων.....	24
Εικόνα : 11 Παράδειγμα με τράπεζα	24
Εικόνα : 12 Παράδειγμα με τράπεζα από τις βάσεις hot_spot & spot_categories	25
Εικόνα : 13 Η Λευκάδα στο Google maps	26
Εικόνα : 14 Αρχική σελίδα www.lefkadamap.gr	36
Εικόνα : 15 Ομάδες – Υποομάδες - Σημεία	37
Εικόνα : 16 Επιλογή «Χάρτης».....	38
Εικόνα : 17 Επιλογή «Δορυφόρος».....	38
Εικόνα : 18 Επιλογή «Υβριδικός»	39
Εικόνα : 19 Μενού Πλοήγησης – Zoom in-out.....	39
Εικόνα : 20 Zoom in-out.....	40
Εικόνα : 21 Δενδροειδή δομή	40
Εικόνα : 22 Επιλογή ενδιαφέροντος σημείου	41
Εικόνα : 23 Δομή πίνακα hot_spots.....	43
Εικόνα : 24 Ο πίνακας hot_spots	43
Εικόνα : 25 Δομή πίνακα hot_spots.....	44
Εικόνα : 26 Ο πίνακας spots_categories.....	44
Εικόνα : 27 Εξαγωγή των δεδομένων της βάσης	45

ΚΕΦΑΛΑΙΟ 1

Σε αυτό το κεφάλαιο θα γίνει η περιγραφή του στόχου της πτυχιακής εργασίας, των γλωσσών προγραμματισμού και της τεχνολογίας που έχει χρησιμοποιηθεί για την υλοποίηση της.

1.1 Περίληψη

Ο βασικός στόχος της παρούσας πτυχιακής είναι η δημιουργία μιας ιστοσελίδας με ενδιαφέροντα σημεία για τη Λευκάδα. Η δημιουργία και ο σχεδιασμός της θα γίνει με γνώμονα την απλότητα και την ευχρηστία προκειμένου ο χρήστης να μπορεί να περιηγηθεί και να βρίσκει αυτό που ψάχνει, με όσο το δυνατόν πιο εύκολο, γρήγορο και κατανοητό τρόπο.

Η ιστοσελίδα δίνει την δυνατότητα σε οποιονδήποτε χρήστη να περιηγηθεί μεταξύ διαφόρων κατηγοριών και να αναζητήσει τα σημεία που τον ενδιαφέρουν έτσι ώστε να δει πληροφορίες για αυτά. Έχει σχεδιαστεί και δομηθεί με τέτοιο τρόπο έτσι ώστε να είναι απλή στο χρηστή και τα σημεία να είναι χωρισμένα σε κατηγορίες – υποκατηγορίες για την ευκολία των χρηστών της.

Η κατασκευή της ιστοσελίδας έχει γίνει με τέτοιο τρόπο έτσι ώστε να είναι δυναμική η ενημέρωση των σελίδων της και αυτό οφείλεται στο συνδυασμό php και sql.

1.2 Κίνητρο για τη διεξαγωγή της εργασίας

Στην εποχή που ζούμε το διαδύκτιο έχει γίνει ένα μεγάλο κομμάτι της ζωής μας όπου απευθυνόμαστε για συλλογή πληροφοριών, ανεύρεση διαφόρων ενδιαφερόντων σημείων για επίσκεψη, όπου και αν υπάρχουν.

Έτσι θα ήταν πάρα πολύ χρήσιμο να βρίσκουμε ένα εργαλείο το οποίο θα μας έδινε την δυνατότητα να βρίσκουμε αυτό που μας ενδιαφέρει γρήγορα, εύκολα και θα μπορούσε να προσαρμοστεί πάνω σε όλες μας τις ανάγκες με μια και μόνο επίσκεψη στην ιστοσελίδα.

Αυτό μετά την έρευνα μας βρήκαμε ότι ήταν το Google Map Api. Έτσι με μια ιστοσελίδα που θα παρείχαμε την δυνατότητα σε πολλούς χρήστες να μπορούν να βρίσκουν τα διάφορα ενδιαφέροντα σημεία που θέλουν να επισκεφτούν μέσω αυτής της σελίδας εύκολα και γρήγορα.

1.3 Στόχοι της εργασίας και σκοποί

Στόχος αυτής της εργασίας είναι η δημιουργία μιας ιστοσελίδας ώστε να δίνει την δυνατότητα σε οποιονδήποτε χρήστη να περιηγηθεί μεταξύ διαφόρων κατηγοριών και να αναζητήσει τα σημεία που τον ενδιαφέρουν έτσι ώστε να δει πληροφορίες για αυτά.

Έχει σχεδιαστεί και δομηθεί με τέτοιο τρόπο έτσι ώστε να είναι ευκολόχρηστη και τα σημεία να είναι χωρισμένα σε κατηγορίες και υποκατηγορίες για την ευκολία των χρηστών της.

Σκοπός της εργασίας αυτής είναι να μάθουμε για το σχεδιασμό και προγραμματισμό μιας ιστοσελίδας χρησιμοποιώντας τα εργαλεία της Google και η κατανόηση του τρόπου λειτουργίας του Google Map Api. Επίσης είναι και η κατανόηση κάποιων βασικών γλωσσών προγραμματισμού που απαιτούνται για την δημιουργία ιστοσελίδων. Σε αυτό το σημείο βοήθησαν διάφορα εργαλεία που παρέχει το Google.

Πιο συγκεκριμένα στόχος της πτυχιακής είναι η υλοποίηση ενός ιστοτόπου ο οποίος θα διαβάζει δεδομένα που σχετίζονται με ενδιαφέροντα σημεία της Λευκάδος από μία βάση δεδομένων και παρουσιάζει στο χάρτη τα ενδιαφέροντα σημεία με τις πληροφορίες που μας ενδιαφέρουν από τη βάση.

Για την υλοποίηση της εφαρμογής, έχουν χρησιμοποιηθεί τα παρακάτω :

1. HTML για την δημιουργία του web site και την ενσωμάτωση της PHP.
2. PHP για την δημιουργία και επεξεργασία του SQL .
3. CSS για την σχεδίαση του web site.
4. SQL για την διαχείριση των δεδομένων κάθε σημείου.

1.4 Δομή εργασίας

Η δομή της εργασίας μας είναι οι εξής στα πιο κάτω κεφάλαια που ακολουθούν:

- (α) Στο δεύτερο κεφάλαιο αναφερόμαστε στην μεθοδολογία που χρησιμοποιήσαμε για την υλοποίηση της πτυχιακής και αναφερόμαστε περιληπτικά σε θεωρίες των γλωσσών προγραμματισμού που χρησιμοποιήσαμε.
- (β) Στο τρίτο κεφάλαιο γράφουμε πιο λεπτομερή την έρευνα που κάναμε γύρω από τον wampserver και την εγκατάσταση του. Επίσης αναφερόμαστε στην γενικότερη ιδέα του phpmyadmin και στο τι μας έκανε να το επιλέξουμε για την υλοποίηση της πτυχιακής μας.
- (γ) Το τέταρτο κεφάλαιο μιλά για την ανάλυση που κάναμε στο πρόβλημα που καλούμαστε να λύσουμε ,τις διεπαφές προγραμματισμού εφαρμογών Google Maps API και πώς ενσωματώνονται στην ιστοσελίδα μας.
- (δ) Στο πέμπτο κεφάλαιο αναφερόμαστε στα γεωγραφικά συστήματα πληροφοριών στα πλεονεκτήματα καθώς και τη χρήση που έχουν στις μέρες μας σαν σύστημα αναφοράς.
- (ε) Το έκτο κεφάλαιο μας δείχνει τις δυνατότητες της εφαρμογής και πιο συγκεκριμένα το πως μπορούμε να βρούμε ένα σημείο πάνω στο χαρτη.Ακόμη γίνεται παρουσίαση των πληροφοριών του σημείου που μας ενδιαφέρει.
- (ζ) Το έβδομο κεφάλαιο αναφερόμαστε στο δεύτερο κομμάτι της πτυχιακής μας για το πώς δηλαδή γράψαμε τα ενδιαφέροντα σημεία για τη Λευκάδα στο χάρτη και γενικά για τα στάδια υλοποίησης της ιστοσελίδας.Συγκεκριμένα πως κατασκευάστηκε η βάση και τι περιέχει και το κώδικα της εφαρμογής.
- (η) Τέλος στο όγδοο κεφάλαιο αναγράφονται στα αποτελέσματα, τα συμπεράσματά την εξέλιξη της πτυχιακής εργασίας και την πρακτική εφαρμογή που μπορεί να έχει.

ΚΕΦΑΛΑΙΟ 2

Μεθοδολογία Υλοποίησης

2.1 Σχέδιο δράσης για την εκπόνηση της εργασίας

Για την εκπόνηση της πτυχιακής εργασίας θα χρειαστεί κάποιος web server που θα φιλοξενεί τις ιστοσελίδες και την βάση δεδομένων, μια γλώσσα προγραμματισμού για την ανάπτυξη των CSS scripts, ένας web browser για την προβολή, καθώς και κάποιο σύστημα βάσεων δεδομένων για την ανάπτυξη των βάσεων δεδομένων.

Ο web server που θα χρησιμοποιήσω για την φιλοξενία της εφαρμογής είναι ο Apache wamp Server ενώ το σύστημα διαχείρισης για τη βάση δεδομένων θα είναι η phpmySQL.

Η εφαρμογή θα αποτελείται από στατικές και δυναμικές ιστοσελίδες. Οι στατικές ιστοσελίδες θα αναπτυχθούν με την χρήση της HTML ενώ η ανάπτυξη δυναμικών ιστοσελίδων απαιτεί προγραμματισμό διαδικτύου PHP.

Η μεθοδολογία που θα ακολουθήσουμε όσον αφορά την ανάπτυξη της εφαρμογής ορίζονται από την ανάλυση του λογισμικού του Google Map Api και ξεκινά με τον προσδιορισμό και την ανάλυση των απαιτήσεων από όπου θα ορισθούν οι περιπτώσεις χρήσης και οι λειτουργίες που θα αναπτυχθούν.

Στην συνέχεια ορίζονται τα βήματα που πρέπει να γίνουν για την υλοποίηση της κάθε κατηγορίας και υποκατηγορίας των ενδιαφερόντων σημείων πάνω στο χάρτη. Παράλληλα με τα παραπάνω ορίζονται τα δεδομένα που απαιτεί η εφαρμογή και γίνεται η σχεδίαση των βάσεων δεδομένων που απαιτούνται για τα ενδιαφέροντα σημεία.

Αφού γίνουν όλα τα παραπάνω γίνεται η κωδικοποίηση όπου αναπτύσσονται οι λειτουργίες που απαιτεί το λογισμικό. Μετά την κωδικοποίηση ακολουθεί έλεγχος .

2.2 Γλώσσες προγραμματισμού.

2.2.1 HTML

Η HTML (HYPERTEXT MARKUP LANGUAGE), είναι μία γλώσσα η οποία «λεει» στον υπολογιστή πως θα πρέπει να εμφανίσει μία ιστοσελίδα. Τα αρχεία στα οποία γράφεται αυτή η γλώσσα είναι απλά αρχεία κειμένου (ASCII), τα οποία μπορούν να δημιουργηθούν ακόμη και με το Notepad (Σημειωματάριο) των Windows έχοντας απλώς την κατάληξη .htm ή .html. Περιλαμβάνουν μέσα τα λεγόμενα tags, τα οποία χρησιμοποιούνται σε κάποιο πρόγραμμα εμφάνισης ιστοσελίδων (π.χ. Firefox), για να δημιουργηθούν και να εμφανιστούν οι ιστοσελίδες.

Όλα τα tags της HTML, δηλώνονται ως εξής: < ονομασία του tag >. Όλα τα tags της HTML έχουν μία αρχή και ένα τέλος. Στο ενδιάμεσο παρεμβάλλεται το περιεχόμενο του tag. Η αρχή κάθε tag δηλώνεται με <tag> ενώ το τέλος με </tag>.

Γενικά το περιεχόμενο μίας ιστοσελίδας είναι tags τα οποία περιέχουν άλλα tags κ.λ.π. Πάντα τα tags τα οποία δημιουργούνται (ανοίγουν), θα πρέπει αφού πληκτρολογηθεί το περιεχόμενό τους να κλείνουν (τελειώνουν </>). Υπάρχουν εξαιρέσεις, κάποια tags δεν κλείνουν ποτέ. Πρέπει λοιπόν όλα τα tags τα οποία ανοίγουν αφού πληκτρολογηθεί το περιεχόμενό τους να κλείνουν, εκτός βέβαια των tags τα οποία δεν πρέπει εξ' ορισμού να κλείσουν.

Παρακάτω περιγράφεται η σημασία των tags που πάντα πρέπει να υπάρχουν σε μια σελίδα HTML.

<html> TAG με ονομασία HTML

Στην κορυφή του κώδικα βρίσκεται πάντα το tag αυτό, το οποίο δηλώνει ότι ο κώδικας που ακολουθεί παρακάτω είναι κώδικας HTML.

<head> TAG με ονομασία HEAD

Ακριβώς μετά το tag html, βρίσκεται το tag head. Ανάμεσα στο άνοιγμα (< >) και στο κλείσιμο (</ >) του tag, αυτού, υπάρχει περιεχόμενο που αφορά την ιστοσελίδα όπως:

1. Ο τίτλος της
2. Η γλώσσα που υποστηρίζει η σελίδα (π.χ. ελληνικά).
3. Διάφορα keywords κ.α. Το περιεχόμενο αυτό, δεν πρέπει να γραφτεί σε κανένα άλλο μέρος του κώδικα.

</head> TAG με ονομασία HEAD. Σε αυτό το σημείο το tag head κλείνει.

<body> TAG με ονομασία BODY. Από αυτό το σημείο και έπειτα, γράφεται ο κώδικας που αφορά την εμφάνιση της ιστοσελίδας μας στον browser π.χ. τα κείμενα, οι εικόνες, οι φόρμες, κ.λ.π. θα εισάγονται πάντα με τη βοήθεια άλλων tags.

</body> TAG με ονομασία BODY. Σε αυτό το σημείο το tag body, κλείνει και πάντα πριν κλείσει το tag HTML.

</html> TAG με ονομασία HTML. Πάντα στο τέλος της σελίδας το tag με ονομασία HTML κλείνει, δηλώνοντας ότι δεν υπάρχει πλέον άλλος κώδικας HTML.

2.2.1.1 DOCTYPE

Επίσης πρέπει να περιέχει κάποιο τύπο DOCTYPE . Το DOCTYPE (document type declaration) είναι πολύ βασικό στην κατασκευή ιστοσελίδων – websites. Η χρήση του είναι απαραίτητη όταν θέλουμε να πληρούμε τα παγκόσμια πρότυπα W3C και να πετυχαίνουμε μέγιστη συμβατότητα με τα προγράμματα πλοήγησης διαδικτύου “browsers”.

Υπάρχουν διάφοροι τύποι DOCTYPE αντίστοιχα για HTML – XHTML και το καθένα λειτουργεί ανάλογα με την χρήση του. Αναλυτικότερα:

2.2.1.1.1 HTML 4.01 Strict

Σύμφωνα με τα Web standards το HTML 4.01 Strict DOCTYPE είναι αυτό που πρέπει να χρησιμοποιούμε στον σχεδιασμό web εφαρμογών εάν θέλουμε να πετύχουμε 100% εγκυρότητα (valid). Ουσιαστικά είναι μια «μειωμένη» έκδοση της HTML 4.01 που δίνει έμφαση περισσότερο στην δομή και όχι στην παρουσίαση. Τα Deprecated elements και χαρακτηριστικά (attributes) όπως frames, και link targets δεν επιτρέπονται στην HTML 4 Strict. Το βασικό πλεονέκτημα είναι πως με τη χρήση του συγκεκριμένου DOCTYPE ενημερώνουμε τους browsers να κάνουν «tender» την ιστοσελίδα με τον πιο αυστηρό και έγκυρο (valid) τρόπο πετυχαίνοντας μέγιστη συμβατότητα και εγκυρότητα.

Το μειονέκτημα όμως είναι πως είναι αρκετά περιοριστικό και δεσμευτικό σε θέματα παρουσίασης, δεν επιτρέπονται απλές βασικές λειτουργίες όπως το target=”_blank” και θέλει αρκετή εξοικείωση από την HTML 4.01 Transitional.

2.2.1.1.2 Transitional

Το HTML 4.01 Transitional DOCTYPE πιθανώς να είναι το πιο διαδεδομένο DOCTYPE στο διαδίκτυο. Η αρχική του χρήση ήταν για την μετάβαση από παλαιότερες εκδόσεις HTML σε HTML 4.01. Τώρα πλέον χρησιμοποιείται κυρίως γιατί υποστηρίζει την χρήση παλαιών deprecated tags τα οποία δεν υποστηρίζονται στην Strict έκδοση. Ουσιαστικά προσφέρει περισσότερη ελευθερία στους web designers και προωθεί καλύτερα τον τρόπο παρουσίασης μιας ιστοσελίδας ενώ πληροί τα παγκόσμια πρότυπα W3C. Χρησιμοποιώντας το συγκεκριμένο DOCTYPE ενημερώνετε τους browsers ότι χρησιμοποιείτε την τελευταία έκδοση HTML αλλά θα πρέπει να προσέξετε το cross browser compatibility (συμβατότητα με τα προγράμματα πλοήγησης διαδικτύου) ώστε να έχετε συμβατότητα με τους browsers.

2.2.2 CSS

Τα CSS (Cascading Style Sheets), είναι ένα σύνολο από ορίσματα, διαφορετικά από αυτά που συναντάμε στην HTML, τα οποία αναπτύχθηκαν με στόχο την καλύτερη διαχείριση της εμφάνισης των ιστοσελίδων. Μπορούμε μέσα από τα CSS, να ορίσουμε γραμματοσειρές, χρώματα, στοίχιση, backgrounds, κ.λ.π.

Ουσιαστικά αποτελούν κλάσεις αντικειμένων τα οποία παρέχουν μεθόδους για υλοποίηση σε κάποια tags της HTML.

Για παράδειγμα μία κλάση CSS μπορεί να είναι η main , η οποία ορίζει ότι η γραμματοσειρά που θα χρησιμοποιηθεί είναι η Verdana, μεγέθους 11, bold και το κείμενο θα είναι στοιχισμένο στο κέντρο. Για να εφαρμόσουμε αυτή την κλάση και τις ιδιότητες – μεθόδους της μέσα στην HTML, χρησιμοποιούμε εντός όλων σχεδόν των tags της HTML που αφορούν σε εμφάνιση κειμένου, πινάκων, γραμμών ή στηλών πινάκων, links, φορμών κ.λ.π. την παράμετρο class="όνομα κλάσης CSS". Για παράδειγμα αν θέλουμε σε μία παράγραφο να εφαρμόσουμε τις ιδιότητες της κλάσης main που αναφέραμε παραπάνω πρέπει να γράψουμε:

```
<p class="main">Το κείμενό μας</p>
```

Τα ορίσματα των CSS, μπορούν να εισαχθούν ως κώδικας τόσο μέσα στην HTML, όσο και σε ξεχωριστό αρχείο με κατάληξη .css το οποίο στη συνέχεια αναφέρεται ως link στην HTML.

2.2.2.1 Εισαγωγή ορισμάτων CSS

Για να ορίσετε ότι σε μία σελίδα HTML, θα εισαχθεί κώδικας CSS πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<style type="text/css">
<!--
Ακολουθεί ο κώδικας των CSS.
-->
</style>
```

Στην περίπτωση που ο κώδικας των CSS, έχει γραφτεί σε ξεχωριστό αρχείο με κατάληξη .css τότε αντίστοιχα πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<link href=" διαδρομή και όνομα αρχείου.css " rel="stylesheet" type="text/css">
```

(μπορούμε να έχουμε περισσότερα από ένα αρχεία css στην ίδια σελίδα , μεγαλύτερη προτεραιότητα έχει αυτό που γράφτηκε τελευταίο.)

2.2.2.2 Κλάσεις CSS

Παραπάνω αναφέραμε ότι τα CSS, αποτελούνται από κλάσεις και ιδιότητες αυτών των κλάσεων. Ένα παράδειγμα μπορεί να είναι το παρακάτω το οποίο έχει δύο κλάσεις τις body και header οι οποίες ορίζουν διαφορετικές ιδιότητες.

```
body {
    background-color:#FFFFCC;
}
#header {

    background-image:url(images/logo.jpg);
    width: auto;
```

```
height: 130px;
margin: 10;
padding: 10;
}
```

Κοινώς για να ορίσετε μία κλάση CSS, εντός ενός αρχείου ή εντός της ίδιας της HTML σελίδας, όπως αναφέρθηκε παραπάνω πρέπει να εισάγετε τα εξής:

```
.όνομα κλάσης {
ιδιότητες κλάσης
}
```

2.2.2.3 JavaScript

Η JavaScript είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα (αλληλεπίδραση, διαδραστικότητα) στις ιστοσελίδες μας και είναι ανταγωνιστική της γλώσσας προγραμματισμού VBScript. Δημιουργήθηκε από την εταιρεία Netscape και το αρχικό της όνομα ήταν LiveScript. Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεσθεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript.

2.2.2.3.1 Ιεραρχία JavaScript

Η JavaScript οργανώνει όλα τα στοιχεία μιας web σελίδας σε μια ιεραρχία. Κάθε στοιχείο της σελίδας βλέπεται σαν αντικείμενο. Κάθε αντικείμενο έχει τις δικές του ιδιότητες (properties) και μεθόδους (methods). Με τη βοήθεια του JavaScript μπορείτε εύκολα να διαχειριστείτε τα αντικείμενα

·
Ιεραρχία της JavaScript σε μια web σελίδα.

Έχουμε δύο εικόνες (images), ένα link και μία φόρμα (form) με δύο πεδία κειμένου και ένα κουμπί. Από πλευράς JavaScript, το παράθυρο του browser θεωρείται σαν ένα αντικείμενο, εν ονόματι window. Το αντικείμενο window περιέχει συγκεκριμένα στοιχεία, όπως η StatusBar. Μέσα στο παράθυρο, μπορεί να γίνει η 'φόρτωση' ενός HTML έγγραφου (ή άλλου τύπου αρχείου - προς το παρόν, θα περιοριστούμε σε HTML αρχεία). Αυτή η σελίδα είναι το αντικείμενο document. Αυτό σημαίνει ότι το αντικείμενο document αναπαριστά το HTML έγγραφο το οποίο 'βλέπει' ο χρήστης εκείνη τη στιγμή.

Το document είναι ένα πολύ σημαντικό αντικείμενο στο JavaScript - θα χρησιμοποιείται συχνά. Οι ιδιότητες (properties) του document είναι για παράδειγμα το χρώμα του φόντου.

Αλλά το πιο σημαντικό είναι ότι όλα τα αντικείμενα HTML είναι ιδιότητες του document. Αντικείμενα είναι για παράδειγμα τα links, ή οι φόρμες.

2.2.3 PHP

Η PHP είναι μια γλώσσα script από την πλευρά του διακομιστή, σχεδιασμένη ειδικά για το Web. Μέσα σε μια HTML σελίδα μπορεί να ενσωματωθεί ο PHP κώδικας, που θα εκτελείται κάθε φορά που κάποιος θα επισκέπτεται τη σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης.

2.2.3.1 Ιστορία της PHP

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. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

2.2.3.2 Τι είναι η PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor.

Είναι μια server-side (εκτελείται στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα, η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), άντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα.

Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

2.2.3.3 Πλεονεκτήματα της PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Allaire Cold Fusion.

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως :

1. Υψηλή απόδοση
2. Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
3. Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
4. Χαμηλό κόστος
5. Ευκολία μάθησης και χρήσης
6. Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερή περιγραφή αυτών των πλεονεκτημάτων.

1. Απόδοση

Η PHP είναι πολύ αποτελεσματική. Με ένα φθινό διακομιστή μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies (<http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

2. Ολοκλήρωση με Βάσεις δεδομένων

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, μπορεί κάποιος να συνδεθεί κατευθείαν με τις βάσεις δεδομένων PostgreSQL, mSQL, Oracle, dbm, filePro, Informix, InterBase, Sybase, μεταξύ άλλων. Χρησιμοποιώντας το Open Database Connectivity Standard (ODBC) μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC. Αυτό περιλαμβάνει και τα προϊόντα της Microsoft products.

3. Ενσωματωμένες Βιβλιοθήκες

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. Μπορεί κάποιος να δημιουργήσει εικόνες GIF δυναμικά, να συνδεθεί με άλλες υπηρεσίες δικτύων, να στείλει ηλεκτρονικό ταχυδρομείο, να δουλέψει με cookies και να δημιουργήσει PDF έγγραφα: όλα αυτά με λίγες γραμμές κώδικα.

4. Κόστος

Η PHP είναι δωρεάν. Μπορεί κάποιος να 'κατεβάσει' την τελευταία έκδοση από το <http://www.php.net>, χωρίς χρέωση.

2.2.3.4 Που Χρησιμοποιείται ένα PHP script.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting.

Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Απαιτούνται τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή (parser) (CGI ή server module), ένα webserver (εξυπηρετητή σελίδων) και ένα web browser ("φυλλομετρητή"). Πρέπει κάποιος να 'τρέξει' τον webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορεί να προσπελάσει τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.

- Command line scripting.

Μπορεί κάποιος να φτιάξει ένα PHP script για να το 'τρέξει' χωρίς server ή browser. Απαιτείται μόνο ο PHP μεταγλωττιστής για να χρησιμοποιηθεί με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε *nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.

- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών).

Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν γνωρίζει την PHP πολύ καλά και θέλει να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές, μπορεί επίσης να χρησιμοποιήσει το PHP-GTK για αυτού του είδους τα προγράμματα. Έχει επίσης τη δυνατότητα να γράφει cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή. Αν κάποιος ενδιαφέρεται για το PHP-GTK, θα πρέπει να επισκεφθεί την δική του ιστοσελίδα.

Επίσης, η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers,

Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής. Έτσι με την PHP, υπάρχει η ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, μπορεί κάποιος ελεύθερα να χρησιμοποιήσει συναρτησιακό (procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα.

Με την PHP δεν είναι κάποιος περιορισμένος να εξάγει HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορεί επίσης να Πραγματοποιηθεί εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. 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 σας περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή.

2.2.3.5 Τι μπορεί να κάνει η PHP.

Οτιδήποτε. Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορεί να κάνει τα ίδια πράγματα με οποιοδήποτε άλλο CGI πρόγραμμα, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies.

2.2.3.6 Συναρτήσεις.

Στην PHP, για να «μεταφερθούν» οι τιμές που έχουν οι μεταβλητές μας από μία σελίδα σε μία άλλη, πρέπει να αποθηκευθούν προσωρινά. Αυτό γίνεται με 2 μεθόδους την GET και την POST.

2.2.3.6.1 GET

Όταν χρησιμοποιείται η μέθοδος GET τα δεδομένα είναι σε μορφή text και είναι ορατά από τον καθένα στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου GET είναι περιορισμένη. Έτσι για να περάσουμε μια μεταβλητή που θα μας δίνει το όνομα κάποιου γίνεται ως εξής: (selida.php?name=Stathis).

2.2.3.6.2 POST

Όταν χρησιμοποιείται η μέθοδος POST τα δεδομένα μεταδίδονται με ξεχωριστή σύνδεση στο script και δεν είναι ορατά στο URL που καλεί το script. Επιπλέον η ποσότητα των δεδομένων που μπορούν να μεταδοθούν με τη χρήση της μεθόδου POST. Τα δεδομένα στέλνονται κυρίως από φόρμες, όπου μέσα στο \$_POST[""] βάζουμε το όνομα (name) του πεδίου που μας στέλνει η φόρμα.

Οι πληροφορίες που στέλνονται από την φόρμα μας με την μέθοδο POST είναι «αόρατες», και δεν έχουν όρια στο μέγεθος των πληροφοριών που μπορούν να στείλουν.

2.2.4 SQL (Structured Query Language)

2.2.4.1 Η ιστορία της SQL

Η SQL βασίστηκε σε ιδέες που προτάθηκαν στις αρχές του 1970 από τον ερευνητή Tedd Codd των εργαστηρίων IBM San Josi Research Laboratories. Δημιουργήθηκε με σκοπό να παρέχει μια ημι – φυσική γλώσσα για το IBM System Relational

database system. Αρχικά οι σχεσιακές βάσεις δεδομένων είχαν πολλούς και πιο αποδοτικούς ανταγωνιστές π.χ. συστήματα διαχείρισης δεδομένων τα οποία βασίζονταν σε δικτυακά μοντέλα δεδομένων. ωστόσο, όλο και πιο αποδοτικά σχεσιακά συστήματα άρχισαν να εμφανίζονται στα τέλη του 1980 κι αυτό οδήγησε στην επικράτηση των σχεσιακών βάσεων δεδομένων και της SQL.

2.2.4.2 DML (Data Manipulation Language) DDL (Data Definition Language) DCL (Data Control Language)

Η δομημένη γλώσσα ερωταποκρίσεων SQL (Structured Query Language, SQL) χρησιμοποιείται για τη διαχείριση των δεδομένων μιας βάσης δεδομένων. Η διαχείριση περιλαμβάνει τόσο τη δημιουργία και μεταβολή των πινάκων της εφαρμογής, όσο και την καταχώρηση και ανάκτηση δεδομένων με βάση συγκεκριμένα κριτήρια επιλογής. Μια τυπική γλώσσα SQL περιλαμβάνει τις επόμενες δομικές μονάδες:

- Γλώσσα ορισμού δεδομένων (Data Definition Language, DDL)

Η γλώσσα αυτή περιλαμβάνει εντολές που μας επιτρέπουν να υλοποιήσουμε πίνακες, σχέσεις ανάμεσα σε πίνακες και γενικά όλη τη δομή μιας βάσης δεδομένων.

- Γλώσσα χειρισμού δεδομένων (Data Manipulation Language, DML)

Η γλώσσα αυτή επιτρέπει τη διαχείριση των δεδομένων της εφαρμογής, όπως την εισαγωγή, διαγραφή, ανάκτηση και τροποποίηση δεδομένων.

- Ορισμός όψεων της βάσης (View Definition)

Επιτρέπει τη δημιουργία όψεων της βάσης δεδομένων οι οποίες ορίζονται ως εικονικοί πίνακες (virtual tables) οι οποίοι περιέχουν δεδομένα από ένα ή περισσότερους πίνακες της βάσης.

- Ορισμός εξουσιοδοτήσεων (Authorization)

Επιτρέπει τη δημιουργία ομάδων χρηστών και την απόδοση διαφορετικών δικαιωμάτων πρόσβασης σε κάθε έναν από αυτούς, προκειμένου η κάθε ομάδα χρηστών, να διαχειρίζεται μόνο τα δικά της δεδομένα.

- Διαχείρισης ακεραιότητας (Integrity)

Επιτρέπει το λεπτομερή έλεγχο των δεδομένων που καταχωρούνται στη βάση, έτσι ώστε να μην παραβιάζονται οι κανόνες ακεραιότητας (integrity constraints) που έχουμε ορίσει και οι οποίοι όταν τηρούνται απομακρύνουν τον κίνδυνο καταχώρησης ασυνεπών δεδομένων (inconsistent data).

- Η γλώσσα διαχείρισης δεδομένων (Data Control Language, DCL) χειρίζεται τις εξουσιοδοτήσεις των δεδομένων

2.2.4.3 DML Γλώσσα Διαχείρισης Δεδομένων

Η γλώσσα χειρισμού δεδομένων (Data Manipulation Language DML), επιτρέπει την διαχείριση των δεδομένων των πινάκων της βάσης, και πιο συγκεκριμένα, την εισαγωγή, διαγραφή και τροποποίηση των εγγραφών των πινάκων. Επιπλέον, έχουμε την δυνατότητα να ανακτήσουμε από τους πίνακες, δεδομένα τα οποία πληρούν κάποια κριτήρια. Η πραγματοποίηση αυτών των διαδικασιών, γίνεται χρησιμοποιώντας τις εντολές INSERT, DELETE, UPDATE και SELECT.

- Η εντολή SELECT

Η δύναμη της SQL είναι η εντολή SELECT. Χρησιμοποιείται για να ανακαλεί δεδομένα κάνοντας ερωτήματα στην βάση δεδομένων, επιλέγοντας γραμμές από έναν πίνακα, που ταιριάζουν με συγκεκριμένα κριτήρια.

Η σύνταξη της SELECT:
SELECT <πεδία>
FROM <πίνακες>
WHERE <κριτήριο>

- Η εντολή INSERT, εισάγει μια νέα γραμμή (ή και περισσότερες)

Η σύνταξη της INSERT
INSERT INTO SYN
VALUES (4,300,103);

Με αυτή την εντολή εισάγαμε μια νέα γραμμή στον πίνακα SYN. Όμως πρέπει να γνωρίζουμε τις στήλες του SYN και την σειρά τους.

- Η εντολή UPDATE, ενημερώνει τα δεδομένα σε μια ή περισσότερες σειρές.

Η σύνταξη της UPDATE
UPDATE SYN
SET EURO = EURO * 10
WHERE AA = 3;

Η εντολή αλλάζει την 3η συναλλαγή.

- Η εντολή DELETE, διαγράφει μια ή περισσότερες γραμμές.
Σίγουρα αυτή η εντολή πρέπει να χρησιμοποιείται με προσοχή.

Η σύνταξη της UPDATE
DELETE *
FROM PEL
WHERE ID = 104;

ΚΕΦΑΛΑΙΟ 3

Ανάλυση προβλήματος και εγκατάσταση Wamp server

3.1 Ανάλυση προβλήματος

Για τις ανάγκες επίλυσης του προβλήματος μας χρειάζεται να μάθουμε κάποιες βασικές αρχές κάποιων βασικών γλωσσών προγραμματισμού, μελέτη για το περιβάλλον και δομή των τύπων για το πώς δομείτε μια δένδροειδή δομή, η γνώση για το Google Map Api και την βασική ρύθμιση και διαχείριση του.

Η βασική μας θεωρία για την υλοποίηση αυτής της ιστοσελίδας είναι το όσο το πιο δυνατό να καταφέρουμε να προβάλλουμε τα ενδιαφέροντα σημεία μας με απλό και κατανοητό τρόπο για όλους τους πιθανούς χρήστες.

Η τελική απαίτηση από την δημιουργία της ιστοσελίδας είναι να είναι φιλική προς τους χρήστες και να μπορεί να τους εξυπηρετήσει έτσι ώστε να βρίσκουν εύκολα τα ενδιαφέροντα σημεία που υπάρχουν διαθέσιμα.

3.2 Απαιτήσεις συστήματος

Οι απαιτήσεις του συστήματος από την πλευρά μας είναι να έχουμε ένα server μαζί με μια βάση δεδομένων mySql και πρόσβαση μέσω ftp. Επίσης η καλή χρήση και γνώση των δυνατοτήτων του Google Map Api. Από την πλευρά του χρήστη είναι να υπάρχουν όλα τα απαραίτητα στοιχεία μιας σύνδεσης internet καθώς και μια οποιαδήποτε συσκευή έχει πρόσβαση μέσω αυτού.

3.3 Σχεδιασμός υλοποίησης

Βήματα που ακολουθήθηκαν

- Μελέτη από άλλες σελίδες του διαδικτίου. Παρατηρούμε ότι βασικό στόχο τους είναι να είναι απλές και φιλικές προς το χρήστη (π.χ google).
- Έρευνα για την δομή της δένδροειδούς δομής, βασικά πεδία που απαιτούνται όπως τίτλος,κατηγορία, χαρακτηριστικά και όνομα γεωγραφικό μήκος και πλάτος καθώς και φωτογραφίες και επεξεργασία αυτών.
- Εκμάθηση της γλώσσας προγραμματισμού php,javascript και css.
- Τι είναι το Google Map Api, τι προσφέρει και ποιες οι δυνατότητες του.
- Πως μπορούμε να αλλάξουμε τις θεματικές παραλλαγές του.
- Υλοποίηση ιστοσελίδας.

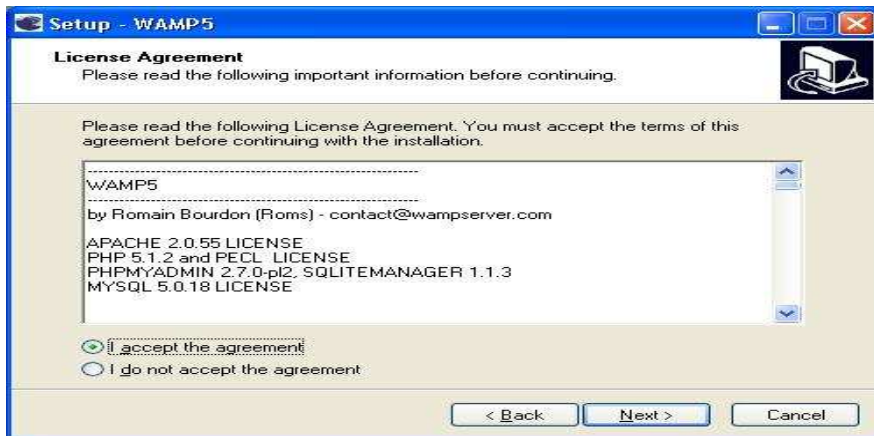
3.4 Τεχνολογία

3.4.1 Εγκατάσταση Wamp

Έχω εγκαταστήσει το αρχείο wamp.exe από τον ιστοχώρο <http://www.wampserver.com/en/download.php>.Είναι η ελαφριά έκδοση και συνιστάται για 'γρήγορη' εργασία χρησιμοποιώντας μόνο PHP και MySQL.



Εικόνα : 1 Setup Wizard



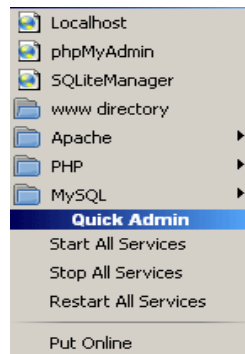
Εικόνα : 2 accept the agreement



Εικόνα : 3 Finish Setup

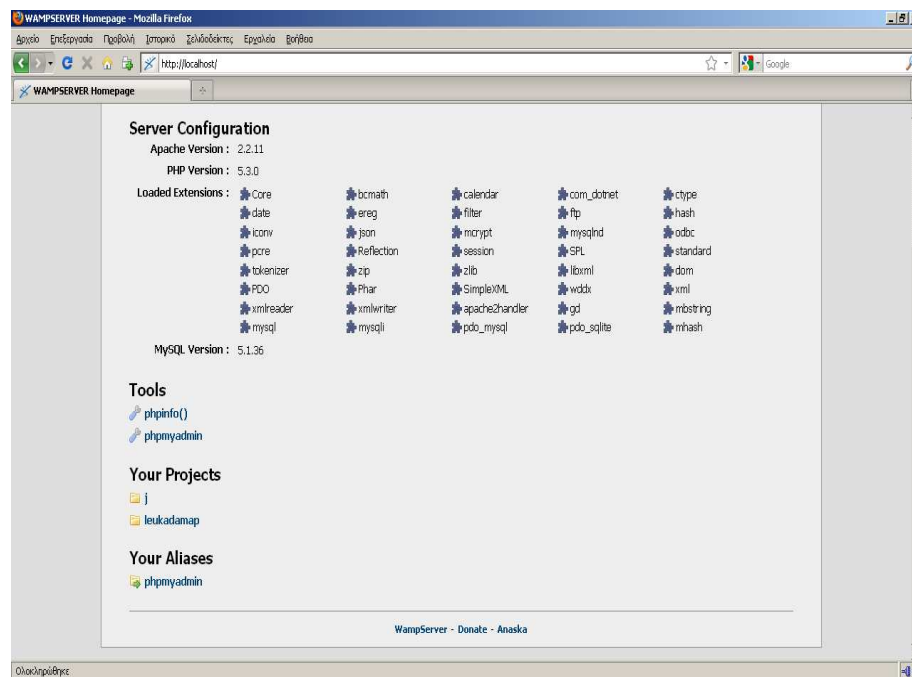


Εικόνα : 4 Start WampServer



Εικόνα : 5 Localhost & phpMyAdmin

Κάτω δεξιά στις συντομεύσεις εμφανίζετε το εικονίδιο του WampServer,με ένα κλικ βλέπουμε τη παραπάνω εικόνα .



Εικόνα : 6 Wamp Server Home Page (επιλογή localhost)

Παραπάνω βλέπουμε τα projects που έχουμε αποθηκεύσει στη διαδρομή που ορίσαμε στην εγκατάσταση. Από προεπιλογή είναι στο φάκελο C:/wamp/www/project name. Το δικό μας project name είναι leukadamap.

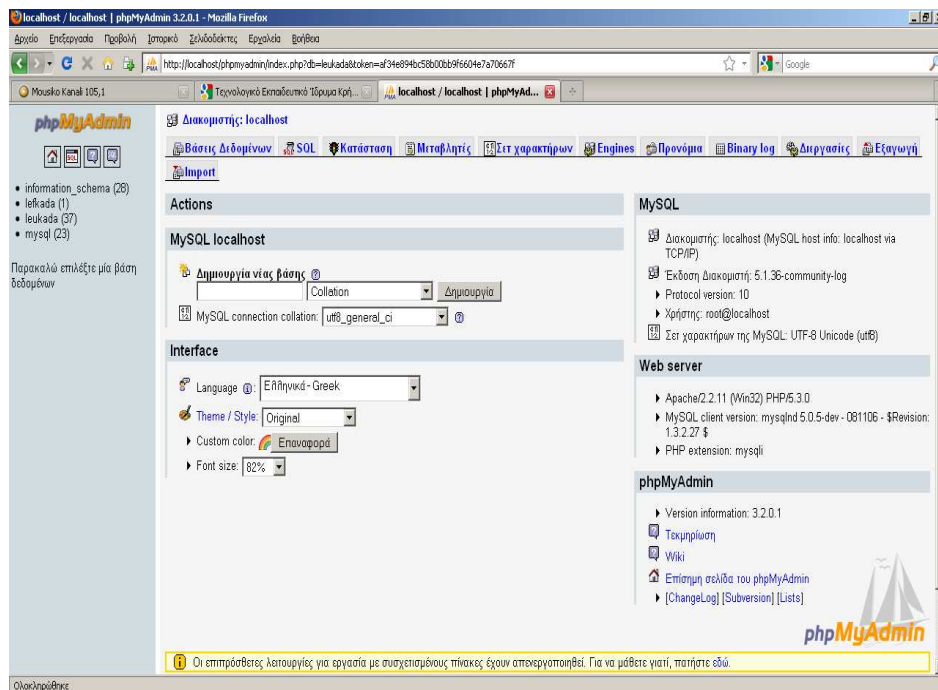
3.4.2 PhpMyAdmin

Το PhpMyAdmin είναι ένα εργαλείο γραμμένο σε php με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Το phpMyAdmin μπορεί να χειρίζεται πλήρως βάσεις δεδομένων, πίνακες, πεδία πινάκων αλλά και ένα ολόκληρο MySQL Server. Υποστηρίζει 54 γλώσσες, μεταξύ των οποίων και τα ελληνικά και είναι λογισμικό ανοιχτού κώδικα.

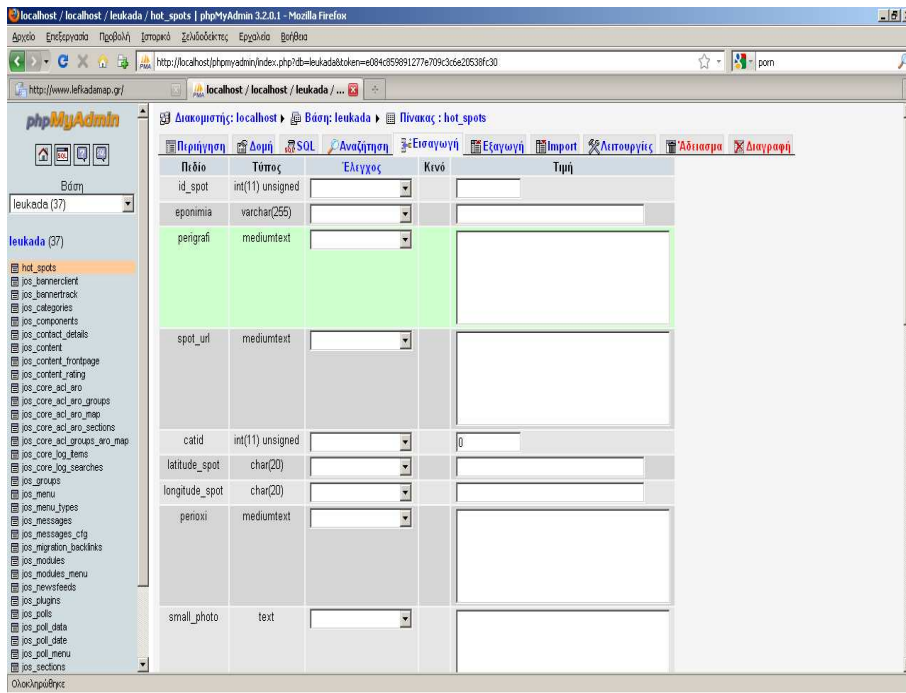
Δυνατότητες του PhpMyAdmin:

Το PhpMyAdmin μπορεί να :

- Δημιουργεί και να διαγράφει βάσεις δεδομένων
- Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
- Κάνει συντήρηση της βάσης
- Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
- Εκτελεί ερωτήματα SQL ακόμα και ομαδικά (batch)
- Διαχειρίζεται κλειδιά σε πεδία
- Φορτώνει αρχεία κειμένου σε πίνακες
- Δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσης)
- Εξάγει δεδομένα σε μορφή CVS, XML και LATEX
- Διαχειρίζεται πολλούς διακομιστές
- Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματα τους
- Ελέγχει την αναφορική δραστηριότητα των δεδομένων των MyISAM πινάκων
- Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων
- Εκτελεί αναζητήσεις σε όλη τη βάση δεδομένων ή μέρος αυτής
- Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
- Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση του MySQL

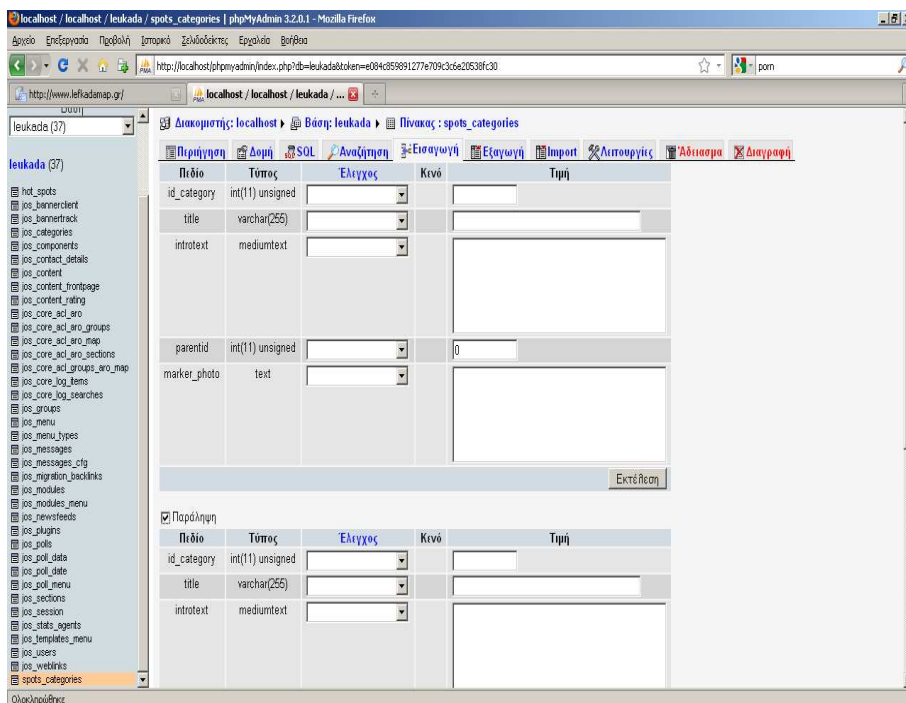


Εικόνα : 7 Δημιουργία νέας βάσης



Εικόνα 8 Εισαγωγή δεδομένων στο πίνακα hot_spot

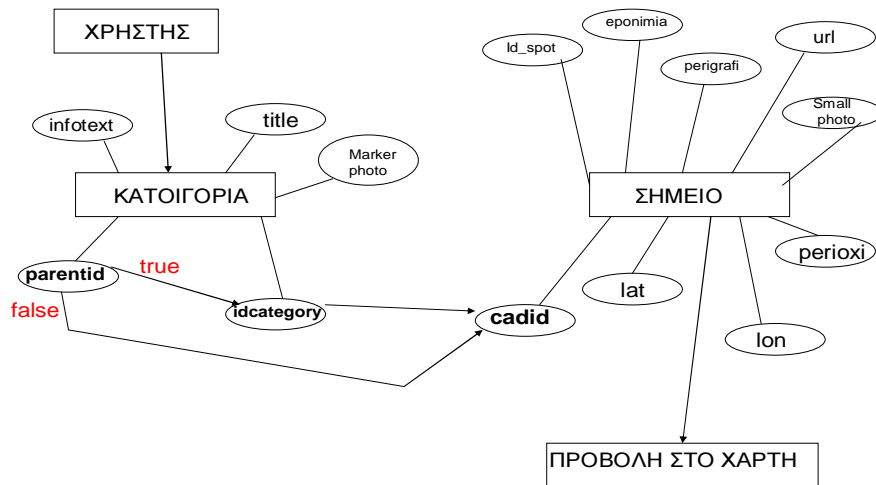
Παραπάνω βλέπουμε τα παιδιά που μετά τη συμπλήρωσή τους μας παρουσιάζονται σαν πληροφορίες στο balloon του κάθε σημείου.



Εικόνα 9 Εισαγωγή δεδομένων στο πίνακα spot_categories

Εδώ έχουμε μια όψη του πίνακα spot_categories από όπου διαχειριζόμαστε τις ομάδες – υποομάδες του δένδρου για κάθε σημείο.

ΣΧΕΔΙΑΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ



Εικόνα 10 Σχέδιο βάσης δεδομένων

Παραπάνω υπάρχει η δομή της βάσης δεδομένων και η λογική σχεδίαση της. Πιο συγκεκριμένα ο πίνακας κατηγορία (spot_categories) περιέχει τα χαρακτηριστικά για τη κατηγορία που έχει ενταχθεί στη δενδροειδή δομή. Το πεδίο parentid αποτελεί τη κατηγορία και το πεδίο idcategory αποτελεί την υποκατηγορία στο δένδρο.

Όταν λιπών ο χρήστης επιλέξει μια κατηγορία ή υποκατηγορία από το δένδρο (τα δυο αυτά πεδία συνδέονται με το πεδίο cadid του πίνακα σημείο (hot_spots)), έχουμε προβολή του ενδιαφέροντος σημείου και των χαρακτηριστικών του στο χάρτη. Η προβολή στο χάρτη γίνεται μέσω Php και περιγράφεται σε παρακάτω κεφάλαια.

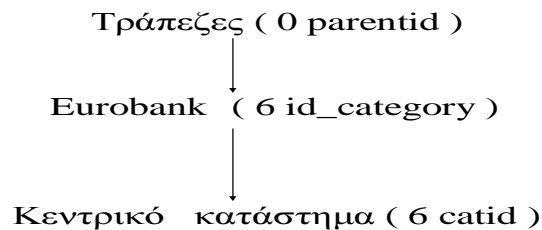
Βλέπουμε στο παρακάτω παράδειγμα ότι το πεδίο parentid της τράπεζας είναι 0, αυτό τη δηλώνει στο δένδρο σαν κατηγορία ενώ το ίδιο πεδίο της Eurobank είναι 3 όπου αυτό σημαίνει ότι μιλάμε για υποκατηγορία.

Απο την άλλη πλευρά το idcategory της τράπεζας είναι το 3, αυτό τη συνδέεται με το parentid της Eurobank που όπως είπαμε είναι και αυτό 3.

Για να αντιστοιχήσουμε την υποκατηγορία Eurobank με τα σημεία πρέπει όλα τα cadid να έχουν τη τιμή 6 όπως και το 'Κεντρικό κατάστημα'.

Εάν είχαμε ένα πιο απλό παράδειγμα χωρίς υποκατηγορίες αλλά μόνο με κατηγορίες θα μπορούσε να σχετίζεται με την επιλογή 'παραλίες'. Εδώ θα είχαμε για το πεδίο 'παραλίες' σαν parentid 0 και το idcategory 10, οπότε όλες οι παραλίες του πίνακα hot spots σαν πεδίο cadid θα πρέπει να έχουν τη τιμή 10.

Παράδειγμα



Εικόνα 11 Παράδειγμα με τράπεζα

id_category	title	introtext	parentid	marker_photo
17	Κινηματογράφοι		0	blesmile.png
18	Αρχαιολογικοί Χώροι		0	blue.png
16	Νοσοκομεία		0	mm_20_red.png
14	Bar - Club		0	mm_20_green.png
15	ΑΤΜ		0	mm_20_red.png
12	Δημόσιοι Χώροι		0	blue.png
13	Μουσεία		0	blue.png
11	Fitness		2	red_image1.png
10	Παραλίες-Beaches		0	mm_20_white.png
9	BP		7	bp.png
8	Shell		7	shell.png
7	Βενζινοδίκια		0	red_image.png
6	Eurobank		3	eurobank.png
5	Αγροτική		3	agrotiki.png
4	Εμπορική		3	emponiki.png
3	Τράπεζες		0	prasino.png

id_spot	επωνυμία	περιγραφή	spot_url	catid	latitude_spot	longitude_spot	περίοδος
1	Κατεπώδης Σπίρος	Ξυλουργικές Εργασίες Περιορισμένης Κοιμητικής Πλατείας	http://carpenterskas.webs.com	25	38.829971359226874	20.698145627975464	
2	Αγροτική	Κοιμητική Πλατεία	http://www.agrotiki.gr	5	38.83369051486683	20.708616871969604	
3	Κοιμητική Κατάστημα	Κοιμητική Αγορά	http://www.eurobank.gr	6	38.8320231870997	20.705891847610474	
4	Κατάστημα Νεοχωρίου	Μυλωνογιάνη 35	http://www.emponiki.gr	4	38.7031	20.589	
5	Κοιμητική Κατάστημα	Διονυσίου 35	http://www.emponiki.gr	4	38.830389251573465	20.7042396068573	
6	Μηχανήλης Ευστάθιος	Κολοκοτρώνη 54	http://www.michailidis.gr	22	38.82870930944846	20.70525884623296	
7	Γαϊτόνος Δημήτριος	Μοργουίου 24	http://www.gaitanos.gr	23	38.68	20.619	
8	Κούρας Ιωάννης	Στεφανοπούλου 125	http://www.koutras.gr	22	38.82381134145626	20.70000171861377	
9	Καστανίδης Μιχαήλ	Ελευθερίου 23	http://www.kastanidis.gr	8	38.689	20.637	

Εικόνα 12 Παράδειγμα με τράπεζα από τις βάσεις hot_spot & spot_categories

3.4.3 Τι είναι το WAMP

Το WampServer είναι ένα web Windows περιβάλλον ανάπτυξης. Σας επιτρέπει να δημιουργήσετε εφαρμογές web με Apache, PHP και η βάση δεδομένων σε MySQL. Επίσης, έρχεται με το phpMyAdmin για να διαχειρίζεστε εύκολα τις βάσεις δεδομένων σας.

Το WampServer εγκαθιστάτε αυτόματα και η χρήση του είναι πολύ διαισθητική. Θα μπορείτε να συντονιστείτε τον server σας χωρίς καν να αγγίζετε την ρύθμιση αρχεία. Το WampServer είναι η μόνη συσκευασμένη λύση που θα σας επιτρέψει να αναπαραγάγετε διακομιστή παραγωγής. Μόλις WampServer είναι εγκατεστημένο, έχετε τη δυνατότητα να προσθέσετε όσες Apache, MySQL και PHP κυκλοφορίες όπως θέλετε. Το WampServer έχει επίσης trayicon για τη διαχείριση των server σας και τις ρυθμίσεις της.

ΚΕΦΑΛΑΙΟ 4

Στο Κεφάλαιο 4 της Πτυχιακής Εργασίας παρουσιάζονται οι διεπαφές προγραμματισμού εφαρμογών Google Maps API οι οποίες παρέχουν τη δυνατότητα ενσωμάτωσης χαρτών και γεωγραφικών πληροφοριών στις ιστοσελίδες και το πως ενσωματώνονται σε μια ιστοσελίδα.

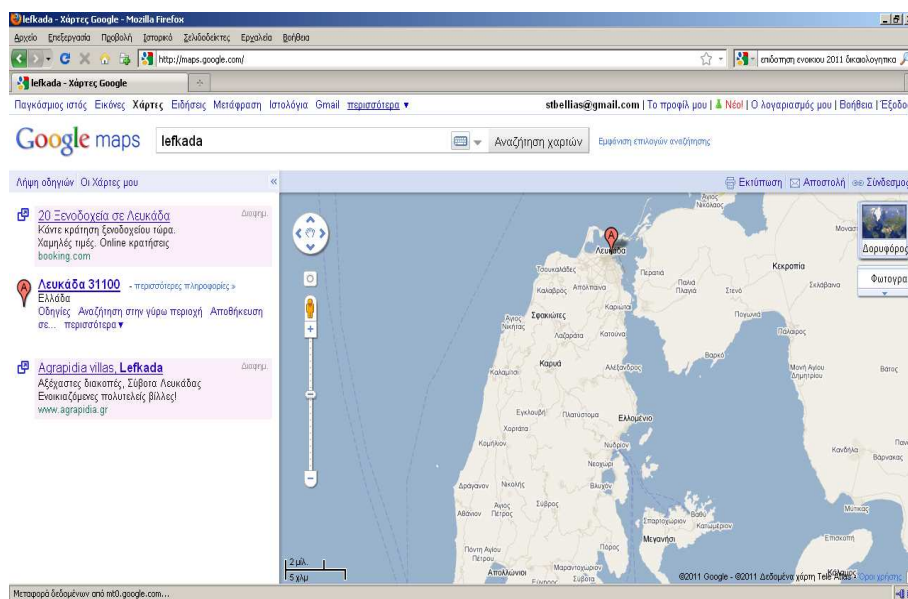
4.1 Χρήση του Google Map Api σε διαδικτυακές εφαρμογές

Οι Χάρτες Google μας προσφέρουν μια γκάμα λειτουργιών τις οποίες μπορούμε να χρησιμοποιήσουμε για να χαρτογραφήσουμε τον κόσμο γύρω μας. Μπορούμε να δημιουργήσουμε έναν χάρτη για κοινή χρήση με φίλους, ο οποίος να επισημαίνει ένα συγκεκριμένο μέρος ή διαδρομή, να διορθώσουμε ένα λάθος το οποίο βλέπουμε στους Χάρτες Google ή να προσθέσουμε χαρακτηριστικά χάρτη σε συγκεκριμένες περιοχές και να γίνουμε ένας πολίτης χαρτογράφος.

Η εμφάνιση των περιοχών σε χάρτη θα γίνει με τη χρήση των Google maps και συγκεκριμένα με το Google map Api .

Είναι μια δωρεάν διαδικτυακή υπηρεσία χαρτογράφησης και εφαρμογή τεχνολογίας που παρέχεται από την Google η κεντρική σελίδα των χαρτών είναι η παρακάτω

<http://maps.google.com>



Εικόνα : 13 Η Λευκάδα στο Google maps

Επιπλέον δίνεται η δυνατότητα ενσωμάτωσης χαρτών σε δικτυακούς τόπους τρίτων μέσω Google map Api .Επίσης υπάρχει και η αντίστοιχη εφαρμογή που ονομάζεται Google earth , η οποία είναι διαθέσιμη για Microsoft Windows, MacOSx, Linux, SymbianOS και IphoneOS προσφέροντας περισσότερα οπτικά χαρακτηριστικά.

4.1.1 Διεπαφή Προγραμματισμού Εφαρμογών Google Maps

Η Google δημιούργησε τους Google χάρτες API για να διευκολύνει την ενσωμάτωση των χαρτών Google στους ιστοχώρους από τον καθένα και αποτελεί ελεύθερη υπηρεσία. Με τη χρησιμοποίηση των Google χαρτών API μπορούμε να ενσωματώσουμε τους πλήρεις χάρτες Google

σε έναν εξωτερικό ιστοχώρο. Με την προσθήκη κώδικα Google JavaScript σε μια ιστοσελίδα μπορούμε να δημιουργήσουμε το δικό μας Google maps προσαρμοσμένο στις ανάγκες μας. Μπορεί κανείς με τις λειτουργίες που παρέχονται να σχεδιάσει δείκτες πάνω στο χάρτη, ή να αναπτύξει ακόμα πιο πολύπλοκες εφαρμογές.

Το πλεονέκτημα από τη χρήση των υπηρεσιών αυτών είναι ότι το Google Maps API είναι μια δωρεάν υπηρεσία, διαθέσιμη για οποιοδήποτε ιστότοπο (web site) που είναι ελεύθερος για το κοινό. Το μόνο που χρειάζεται κάποιος για να χρησιμοποιήσει τις υπηρεσίες είναι να εγγραφεί στο σύστημα της Google για να του παραχωρηθεί ένας κωδικός πρόσβασης στα API. Ο διαδικτυακός τόπος του Google maps Api είναι <http://code.google.com/apis/maps>

Κάθε κλειδί είναι έγκυρο μόνο για ένα συγκεκριμένο όνομα τομέα.

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 πολλαπλών εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες.

Βασικό στοιχείο οποιασδήποτε Google Maps API εφαρμογής είναι ο ίδιος ο χάρτης, ο οποίος φορτώνεται, 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, τα οποία αναλύονται παρακάτω.

Ένας εύχρηστος ιστοτόπος για την εύρεση συντεταγμένων περιοχής είναι ο παρακάτω <http://www.getlatlon.com/>

4.1.1.1. The GMap class

Πρόκειται για την βασική κλάση. Ένα αντικείμενο της κλάσης GMap αντιστοιχεί σε ένα χάρτη στην σελίδα μας. Μπορεί κανείς να δημιουργήσει όσα στιγμιότυπα αυτής της κλάσης επιθυμεί (ένα για κάθε χάρτη στη σελίδα). Όταν δημιουργούμε ένα νέο στιγμιότυπο χάρτη, καθορίζουμε ένα στοιχείο στη σελίδα το οποίο θα περιέχει το χάρτη. Ο χάρτης στη συνέχεια, σαν μέγεθος του χρησιμοποιεί το μέγεθος του στοιχείου που τον περιλαμβάνει εκτός και αν το ορίσουμε διαφορετικά. Η κλάση GMap παρέχει μεθόδους χειρισμού του κέντρου του χάρτη και του επιπέδου zoom, καθώς και μεθόδους για προσθήκη ή αφαίρεση διαφόρων overlays (όπως για παράδειγμα στιγμιότυπα των κλάσεων GMarker και Gpolyline.). Επιπλέον, παρέχει μεθόδους που μας δίνουν τη δυνατότητα να ανοίξουμε ένα «παράθυρο πληροφοριών» το οποίο θα περιέχει λεπτομέρειες για αυτό πάνω στο χάρτη.

4.1.1.2 Γεγονότα (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 Αντικειμένου 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 ακούν (listen) και απαντάνε (respond) στα DOM γεγονότα ανεξάρτητα από τις ιδιαιτερότητες του κάθε φυλλομετρητή. Παραδείγματα 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.

4.1.1.3 Ρυθμιστές (Controls)

Όπως μπορεί να διαπιστώσει κανείς και στη σελίδα του Google Maps (<http://maps.google.com>) υπάρχουν διάφορα στοιχεία στη διεπιφάνεια με το χρήστη (user interface) που επιτρέπουν την αλληλεπίδρασή του με το χάρτη. Τα στοιχεία αυτά είναι γνωστά ως ρυθμιστές (controls) και μπορούν να περιληφθούν στις Google Maps εφαρμογές σε οποιαδήποτε σημείο πάνω στο χάρτη. Τα κυριότερα είναι:

- **GLargeMapControl3D**: Τοποθετεί ένα μεγάλο τρισδιάστατο κουμπί ελέγχου εστίασης και κίνησης του χάρτη, όπως χρησιμοποιείται τώρα στη σελίδα του Google Maps. 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.

4.1.1.4 Overlays

Τα overlays είναι αντικείμενα πάνω στο χάρτη τα οποία έχουν οριστεί με βάση τις γεωγραφικές συντεταγμένες, και έτσι μετακινούνται όταν ο χρήστης μετακινεί το χάρτη ή κάνει zoom σε αυτόν ή όταν αλλάζει ο τρόπος προβολής (η Google παρέχει διάφορους τρόπους προβολής ενός χάρτη, για παράδειγμα δορυφορική προβολή κτλ). Το Google Maps API περιλαμβάνει δύο τύπους overlays, τα markers που είναι εικονίδια πάνω στο χάρτη και τα polylines που είναι γραμμές που κατασκευάζονται από μία σειρά σημείων. 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. Οι markers είναι σχεδιασμένοι να είναι διαδραστικοί, δηλαδή επάνω τους πραγματοποιούνται γεγονότα, όπως ένα κλικ ή δεξί κλικ, και συχνά προβάλλετε παράλληλα με αυτούς 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 προσκήνιο και μια άλλη ως σκιά τοποθετημένες σε συγκεκριμένες θέσεις μεταξύ τους. Παρότι οι δείκτες έχουν μεγάλη χρησιμότητα, η προσθήκη πολλών από αυτούς σε ένα 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 ποιο 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 γραμμικές (drawing), τις γεωδαισικές (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 γεωδαισικές polylines εμφανίζονται ως καμπυλωτές γραμμές λαμβάνοντας υπόψιν 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 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 ίδιος ο χάρτης

Αναπαρίσταται 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.

4.1.1.5 The Info Window

Κάθε σημείο έχει ένα μόνο «παράθυρο πληροφοριών», το οποίο εμφανίζει HTML περιεχόμενο σε ένα παράθυρο πάνω από το χάρτη. Το παράθυρο πληροφοριών μοιάζει με ένα «συννεφάκι» σε ένα βιβλίο comic. Αποτελείται από μια περιοχή με το περιεχόμενο των πληροφοριών η οποία λεπταίνει στην άκρη και γίνεται σαν δείκτης που δείχνει σε ένα καθορισμένο σημείο πάνω στο χάρτη. Αν κάποιος έχει χρησιμοποιήσει το Google Maps ή το Google Local, τότε πολύ πιθανό να έχει δει ένα «παράθυρο πληροφοριών» όταν κάνει κλικ σε ένα εικονίδιο (marker). Άλλο χαρακτηριστικό των παραθύρων αυτών είναι ότι δεν μπορεί κανείς να εμφανίσει περισσότερα από ένα ταυτόχρονα σε ένα δοσμένο χάρτη αλλά μπορεί να μετακινήσει το παράθυρο και να αλλάξει τα περιεχόμενά του αν αυτό είναι επιθυμητό.

Η βασική μέθοδος για ένα παράθυρο πληροφοριών είναι η `openInfoWindow`, η οποία παίρνει σαν είσοδο ένα σημείο και ένα HTML DOM element. Το παράθυρο πληροφοριών εμφανίζεται με το κείμενό του στο δοσμένο σημείο του χάρτη και εμφανίζει το DOM element στην περιοχή που περιλαμβάνει. Η μέθοδος `openInfoWindowHtml` είναι παρόμοια, αλλά παίρνει ένα HTML string σαν δεύτερο όρισμα αντί για ένα DOM element. Ομοίως, η `openInfoWindowXslt` παίρνει ένα σημείο, ένα XML DOM element, και το URL ενός XSLT αρχείου. Στη συνέχεια εφαρμόζει το μετασχηματισμό XSLT στο XML για να παράγει τα περιεχόμενα του παραθύρου. Η μέθοδος αυτή μεταφέρει το XSLT αυτόματα αν δεν έχει ήδη μεταφερθεί από τον browser του χρήστη.

Εκτός από τα παραπάνω, μπορούμε επίσης να εμφανίσουμε ένα παράθυρο πληροφοριών πάνω από ένα overlay όπως για παράδειγμα ένα εικονίδιο (marker). Για να το κάνουμε αυτό περνάμε ως τρίτο όρισμα ένα pixel offset μεταξύ του καθορισμένου σημείου και του κειμένου του παραθύρου πληροφοριών. Η κλάση `GMarker` επιτρέπει μεθόδους `openInfoWindow` οι οποίες χειρίζονται τα pixel offsets αυτόματα βασισμένες στο μέγεθος και σχήμα του εικονιδίου, και συνεπώς δε χρειάζεται να ανησυχεί ο προγραμματιστής για τον υπολογισμό των offsets στην εφαρμογή του.

4.1.1.6 Προσφερόμενες υπηρεσίες του Google Maps API

Πρόσθετες υπηρεσίες τις οποίες προσφέρει το Google Maps API είναι η μετατροπή πραγματικών διευθύνσεων ανά τον πλανήτη σε γεωγραφικές συντεταγμένες (geocoding), κάτι το οποίο είναι χρήσιμο στην τοποθέτηση δεικτών, αλλά και το αντίστροφο (reverse geocoding). Επιπλέον το Google Maps API είναι εξοπλισμένο με κρυφή μνήμη (cache memory) από τη μεριά του πελάτη (client), η οποία αποθηκεύει διάφορες απαντήσεις σε μετατροπές διευθύνσεων, επιτρέποντας ταχύτερη εξυπηρέτηση σε περίπτωση που ζητηθεί η ίδια μετατροπή διεύθυνσης στο μέλλον.

Μια άλλη υπηρεσία είναι η άποψη από δρόμο (street view), η οποία παρέχει πανοραμική θέα 360ο περιοχών για τις οποίες υπάρχουν διαδραστικές εικόνες. Κάθε οπτική της εικόνας χαρακτηρίζεται από την τοποθεσία και τον προσανατολισμό και είναι μοναδική.

Επιπλέον υπηρεσία είναι η προσθήκη του Google Earth στις Maps API εφαρμογές, το οποίο προβάλλει τον χάρτη Google τρισδιάστατα.

Η μπάρα αναζήτησης Google είναι άλλο ένα χρήσιμο εργαλείο με το οποίο αναζητούνται πόλεις, πανεπιστήμια, επιχειρήσεις κ.ά. Η μπάρα αναζήτησης είναι φιλική προς το χρήστη και συμπεριλαμβάνει στα αποτελέσματά της διαφημίσεις με σκοπό το οικονομικό κέρδος αν προστεθεί στις Google Maps εφαρμογές.

Δίνεται η δυνατότητα επιπλέον να περιληφθεί στην εφαρμογή ο υπολογισμός οδηγιών για τη μετάβαση από ένα μέρος σε ένα άλλο, χρησιμοποιώντας μια ποικιλία μέσων μεταφοράς.

4.2 Δικαιώματα χρήσης

Προκειμένου να έχει πρόσβαση ο χρήστης στα δεδομένα, πρέπει να προμηθευτεί από τη Google ένα αλφαριθμητικό κωδικό 86 ψηφίων (API key). Στην ηλεκτρονική διεύθυνση <http://code.google.com/apis/maps/signup.html> αφού συμπληρωθεί η φόρμα με τα απαραίτητα στοιχεία, ολοκληρώνεται η εγγραφή στην υπηρεσία και αποστέλλεται με email ο κωδικός. Μέρος της

διαδικασίας εγγραφής αποτελεί η αποδοχή των όρων χρήσης (Google Maps API Terms of Service). Εκεί αναφέρεται ότι :

- Το API μπορεί να χρησιμοποιηθεί μόνο για προβολή του υλικού και για καμιά άλλη χρήση.
- Δεν επιτρέπεται η αντιγραφή, διανομή, αλλοίωση ή άλλη χρήση.
- Δεν επιτρέπεται η χρήση για real time πλοήγηση
- Δεν επιτρέπεται η χρήση για διαχείριση οχημάτων.
- Δεν επιτρέπεται η χρήση για σύστημα διαχείρισης στόλου ή άλλη παρόμοια εφαρμογή.

Το API μπορεί να χρησιμοποιηθεί μόνο για υπηρεσίες που είναι γενικά προσβάσιμες στους καταναλωτές χωρίς χρέωση. Δεν μπορεί να χρησιμοποιηθεί για υπηρεσίες που απαιτούν συνδρομή ή άλλου είδους περιορισμένη πρόσβαση.

ΚΕΦΑΛΑΙΟ 5

5.1 Γεωγραφικά συστήματα πληροφοριών GIS (Geographical Information Systems)

Τα γεωγραφικά συστήματα πληροφοριών, ευρύτερα γνωστά ως GIS (Geographical Information Systems), διαχειρίζονται με αποτελεσματικό τρόπο τη γεωγραφική πληροφορία και μαζί τα δεδομένα ενός οργανισμού που μπορούν να συσχετιστούν με αυτή. Την τελευταία δεκαετία έχουν γνωρίσει μεγάλη άνθηση και έχουν βρει αρκετές πρακτικές εφαρμογές σε επιχειρήσεις και οργανισμούς. Βέβαια, στην αμερικάνικη ήπειρο αλλά και αρκετές ευρωπαϊκές χώρες, τα γεωγραφικά συστήματα πληροφοριών χρησιμοποιήθηκαν πολύ πριν τα γνωρίσουμε στην Ελλάδα. Στην ευρεία διάδοση των GIS συνέβαλαν τα ακόλουθα: το γεγονός ότι οι εταιρείες λογισμικού κατάφεραν να αναπτύξουν εκδόσεις φιλικές προς τους χρήστες τους μέσω του γραφικού περιβάλλοντος, η δημιουργία και διάθεση αξιόπιστων ψηφιακών δεδομένων που μπορούν να χρησιμοποιηθούν στα συστήματα αυτά (ψηφιακοί χάρτες), η αυξημένη υπολογιστική ισχύς των προσωπικών ηλεκτρονικών υπολογιστών (desktop PCs), καθώς και η συσχέτιση των συστημάτων GIS με τα συστήματα παρακολούθησης οχημάτων, δικτύων ή άλλων αντικειμένων πάνω στη γη, μέσω της τεχνολογίας των δορυφόρων και των τηλεπικοινωνιών.

5.2 Ιστορία των GIS

Η ιδέα της οργάνωσης και συστηματοποίησης της γεωγραφικής πληροφορίας με τη χρήση ηλεκτρονικού υπολογιστή εμφανίστηκε για πρώτη φορά στα μέσα της δεκαετίας του 1960. Η ραγδαία ανάπτυξη της τεχνολογίας οδήγησε τελικά στην εμφάνιση των γεωγραφικών συστημάτων πληροφοριών ή GIS τη δεκαετία του 1980. Η επανάσταση που έφεραν οι υπολογιστές στη διαχείριση της πληροφορίας άργησε να αγγίξει τον τομέα της χωρικής πληροφορίας κυρίως για τεχνικούς λόγους που σχετίζονται με τις αυξημένες απαιτήσεις σε γραφικά. Όταν όμως τη δεκαετία του 1990 ξεπεράστηκαν τα τεχνικά εμπόδια και όταν το κόστος των συστημάτων ηλεκτρονικών υπολογιστών έπαψε να είναι απαγορευτικό, η ευρεία χρήση τους οδήγησε στην ανάπτυξη των GIS και στη σταδιακή δημιουργία των απαραίτητων χωρικών δεδομένων για τη λειτουργία των συστημάτων αυτών.

Η τεχνολογία αυτή γνώρισε μία ευρύτατη σειρά εφαρμογών, σχεδόν σε κάθε ζήτημα όπου η παράμετρος της γεωγραφικής πληροφορίας υπεισέρχεται έμμεσα ή άμεσα. Ορίζοντας το γεωγραφικό σύστημα πληροφοριών, μπορούμε να το χαρακτηρίσουμε ως ένα οργανωμένο σύνολο από πέντε δομικά στοιχεία που περιλαμβάνουν: υλικό εξοπλισμό (hardware), λογισμικό (software), γεωγραφικά δεδομένα, ανθρώπινο δυναμικό και μεθόδους (οι ιδιαίτεροι κανόνες και οι πρακτικές λειτουργίας κάθε οργανισμού ξεχωριστά), με σκοπό τη συλλογή, καταχώριση, διαχείριση, ανάλυση, επεξεργασία και απόδοση, κάθε μορφής πληροφορίας που σχετίζεται με τη γεωγραφική πληροφορία.

Ένα γεωγραφικό σύστημα πληροφοριών είναι ένα εργαλείο χαρτογράφησης και ανάλυσης των στοιχείων που υπάρχουν και των γεγονότων που συμβαίνουν στο γεωγραφικό χώρο. Η τεχνολογία των GIS ολοκληρώνει τις λειτουργίες των συνήθων εφαρμογών βάσεων δεδομένων, όπως αναζήτηση και στατιστική ανάλυση, με τα πλεονεκτήματα της οπτικής απεικόνισης και της γεωγραφικής ανάλυσης που προσφέρουν οι χάρτες. Οι ικανότητες αυτές διακρίνουν τα GIS από τα άλλα πληροφοριακά συστήματα και τα καθιστούν πολύτιμα σε ένα μεγάλο εύρος δημοσίων οργανισμών και ιδιωτικών επιχειρήσεων, για την επεξήγηση γεγονότων, την εκτίμηση αποτελεσμάτων, το σχεδιασμό στρατηγικών, τη λήψη αποφάσεων.

Το σύστημα GIS μοντελοποιεί το χώρο συγκεντρώνοντας και συνδυάζοντας ένα πλήθος πληροφοριών. Για το σκοπό αυτό αποθηκεύει δεδομένα σε ένα σύνολο από διαφορετικές θεματικές βαθμίδες (layers), όπως για παράδειγμα πόλεις, δρόμοι, κτίρια, αγωγοί, γεωγραφικό ανάγλυφο, λίμνες, ποτάμια, λοιπά σημεία ενδιαφέροντος. Οι θεματικές αυτές βαθμίδες συνδέονται μεταξύ τους μέσω γεωγραφικών συντεταγμένων, σε δύο διαστάσεις (γεωγραφικό μήκος και πλάτος), ακόμα και σε τρεις διαστάσεις. Αυτή είναι η απλή αλλά εξαιρετικά ισχυρή αρχή λειτουργίας των GIS που αποδεικνύεται πολύτιμη για την επίλυση πλήθους πραγματικών προβλημάτων.

Το πιο ζωτικό δομικό στοιχείο ενός GIS είναι τα δεδομένα και για το λόγο αυτό οι βάσεις δεδομένων βρίσκονται στην καρδιά ενός τέτοιου συστήματος. Το GIS διαθέτει μία βάση γεωγραφικών δεδομένων ή αλλιώς ένα ψηφιοποιημένο χάρτη ο οποίος υλοποιεί και το μοντέλο του χώρου, όπως αυτό περιγράφηκε παραπάνω με τις θεματικές βαθμίδες. Ταυτόχρονα περιλαμβάνει και μία βάση περιγραφικών δεδομένων που συνδέονται με τα δεδομένα του χώρου. Το κατάλληλο λογισμικό ενός GIS για τη διαχείριση βάσεων δεδομένων και οπτικής απεικόνισής τους, επιτρέπει το συνδυασμό αυτών των δεδομένων και την απεικόνισή τους σε μορφή συνδυασμένων πληροφοριών πάνω σε χάρτες.

5.3 Κατηγορίες GIS

Ο τρόπος με τον οποίο δομείται ο ψηφιοποιημένος χάρτης χωρίζει τα GIS σε δύο μεγάλες κατηγορίες: τα «διανυσματικά» (vector) συστήματα και τα συστήματα «μωσαϊκού» (raster ή grid).

Τα διανυσματικά συστήματα αποθηκεύουν τη γεωγραφική πληροφορία σε αναλυτική μορφή συντεταγμένων, ενώ τα συστήματα μωσαϊκού αποθηκεύουν την πληροφορία σε μορφή πλέγματος ψηφίδων. Στα διανυσματικά GIS η καταγραφή και η επεξεργασία των χωρικών πληροφοριών γίνεται με τη χρήση της γεωμετρίας των διανυσμάτων που περιλαμβάνει σημεία, γραμμές και πολύγωνα, με τα οποία αντιπροσωπεύονται αντιστοίχως τα εξής γεωμετρικά στοιχεία του χώρου: τόποι, γραμμικά στοιχεία και επιφάνειες.

Τα GIS μωσαϊκού τύπου βασίζονται στην αρχή των στοιχειωδών επιφανειών (raster, cells, pixels). Οι στοιχειώδεις επιφάνειες είναι συνήθως τετράγωνα ή παραλληλόγραμμα και ενίοτε τριγωνικής ή εξαγωνικής μορφής. Δημιουργείται δηλαδή ένα πλέγμα πάνω από μία εικόνα (συνήθως ψηφιακά σαρωμένος χάρτης), το οποίο διαχωρίζει την εικόνα σε πολύ μικρά στοιχειώδη κομμάτια - ψηφίδες, παρόμοιες με αυτές που βλέπει κανείς όταν μεγεθύνει υπερβολικά μία ψηφιακή φωτογραφία. Οι χάρτες μωσαϊκού (raster ή grid) υστερούν στη δημιουργία των χωρικών συσχετίσεων που επιτυγχάνουν οι διανυσματικοί (vector) χάρτες. Για το λόγο αυτό τα σύγχρονα GIS χρησιμοποιούν διανυσματικούς χάρτες, ενώ συνοδεύονται και από ειδικές εφαρμογές μετατροπής των χαρτών από «raster» σε «vector» μορφή.

5.4 Χρήσεις των GIS

Ενα σύστημα GIS έχει το πλεονέκτημα ότι διαχειρίζεται ξεχωριστά την αποθήκευση των δεδομένων από την οπτική αναπαράσταση των χαρτών. Αυτό έχει ως αποτέλεσμα τα ίδια δεδομένα να μπορούν να αποτυπωθούν με διαφορετικούς τρόπους. Ενδεικτικά αναφέρουμε ότι μπορούμε να μεγεθύνουμε τον ψηφιακό χάρτη ή να εμφανίσουμε συγκεκριμένα μόνο επίπεδα (layers) της ψηφιακής πληροφορίας. Ενα τέτοιο παράδειγμα συναντάται στις γνωστές εφαρμογές χαρτών της Google όπου ο χρήστης επιλέγει είτε μόνο τον πολιτικό χάρτη, είτε το γεωφυσικό χάρτη (φωτογραφία από δορυφόρο) είτε και τους δύο μαζί και πάνω σε αυτούς μπορεί να εμφανίσει οποιαδήποτε άλλη πληροφορία τον ενδιαφέρει και είναι διαθέσιμη: δρόμοι, πόλεις, σημεία τουριστικού ενδιαφέροντος κ.λπ.

Επιπροσθέτως, στα πλεονεκτήματα των GIS συγκαταλέγεται το γεγονός ότι μπορούμε να εκτελέσουμε ποικίλους υπολογισμούς με τα γεωγραφικά δεδομένα και οποιαδήποτε άλλη πληροφορία μπορεί να συνδυαστεί με αυτά, όπως για παράδειγμα είναι ο υπολογισμός των αποστάσεων μεταξύ τοποθεσιών ή και ο χρόνος μίας διαδρομής. Επίσης, μπορούμε να δημιουργήσουμε πίνακες που να περιλαμβάνουν τα διάφορα χαρακτηριστικά του ψηφιακού χάρτη ή να προσθέσουμε οποιαδήποτε επιπλέον πληροφορία πάνω στο χάρτη.

Μία σημαντική δυνατότητα που προσφέρουν τα GIS είναι το ότι προσδιορίζουν τις διαθέσιμες πληροφορίες στο γεωγραφικό χώρο. Η δυνατότητα αυτή ονομάζεται «γεωκωδικοποίηση» (geocoding) και ένα ενδεικτικό παράδειγμα χρήσης της είναι ο αυτόματος μετασχηματισμός της διεύθυνσης ενός πελάτη σε συντεταγμένες ενός σημείου στον ψηφιακό χάρτη της αντίστοιχης πόλης.

Ο σημαντικότερος παράγοντας για την επιτυχημένη υλοποίηση και χρήση ενός GIS είναι η ύπαρξη των κατάλληλων γεωγραφικών δεδομένων, τα οποία όταν συνδυαστούν με τα υπόλοιπα δεδομένα ενός οργανισμού, δύναται να υποστηρίξουν πολλές λειτουργίες ή και τη λήψη των

αποφάσεων. Τέτοια γεωγραφικά δεδομένα μπορούν είτε να αγοραστούν είτε να δημιουργηθούν εξαρχής με τη χρήση ειδικών διατάξεων ψηφιοποίησης των χαρτών. Μέχρι τις αρχές της δεκαετίας του 2000, μόνο η Γεωγραφική Υπηρεσία Στρατού (ΓΥΣ) προσέφερε αξιόπιστα και σχετικά πλήρη γεωγραφικά δεδομένα για τον Ελλαδικό χώρο, αν και αυτά υστερούσαν στο επίπεδο των αστικών περιοχών (εντός των πόλεων). Σήμερα, η ανάπτυξη των συστημάτων πλοήγησης και παρακολούθησης οχημάτων και η αυξανόμενη διείσδυση των GIS σε δημόσιο και ιδιωτικό τομέα έχουν οδηγήσει στην παραγωγή και διάθεση αξιόπιστων αλλά και οικονομικών γεωγραφικών δεδομένων για κάθε ενδιαφερόμενο.

Το πεδίο εφαρμογής των γεωγραφικών συστημάτων πληροφοριών είναι ευρύτατο καθώς μπορούν να χρησιμοποιηθούν οπουδήποτε η παράμετρος της γεωγραφικής πληροφορίας υπεισέρχεται άμεσα ή έμμεσα, όπως είναι: οι ανάγκες χαρτογράφησης, τα ζητήματα χωροταξίας, περιπτώσεις αστικών και περιφερειακών μελετών, διαχείρισης φυσικών πόρων, οικολογικών ερευνών, διαχείρισης αποβλήτων, κτηματολογίου και πολεοδομικού σχεδιασμού, μελέτης κυκλοφοριακών συνθηκών, διαχείρισης επειγόντων περιστατικών, δημογραφικά ζητήματα. Επίσης μπορούν να χρησιμοποιηθούν σε θέματα διερεύνησης και ανάλυσης μίας αγοράς, ανάλυσης των πωλήσεων μίας επιχείρησης, σχεδιασμού των δικτύων εφοδιασμού και διακίνησης, προγραμματισμού της διανομής προϊόντων, καθώς και παρακολούθησης των δρομολογίων των οχημάτων.

5.5 Χαρακτηριστικά των GIS

Σήμερα, τα γεωγραφικά συστήματα πληροφοριών χρησιμοποιούνται ολοένα και περισσότερο ως εργαλεία που υποστηρίζουν τις λειτουργίες επιχειρήσεων και οργανισμών. Αυτές οι λειτουργίες μπορεί να απαιτούν είτε απλά διαχείριση της γεωγραφικής πληροφορίας, είτε πιο σύνθετη επεξεργασία των χωρικών και περιγραφικών βάσεων δεδομένων, με σκοπό την παραγωγή διαφόρων αναφορών και τη λήψη αποφάσεων. Η διαχείριση της πληροφορίας περιλαμβάνει:

- Δημιουργία δυναμικών ψηφιακών τοπογραφικών χαρτών, που μπορούν εύκολα να ενημερώνονται και επικαιροποιούνται μέσω της ηλεκτρονικής διαχείρισης.
- Δημιουργία «ψηφιακών μοντέλων εδάφους» για τρισδιάστατη απεικόνιση των χαρτών με ανάγλυφο εδάφους, κάτι που μπορεί να χρησιμοποιηθεί στη διαχείριση των ορυκτών πόρων ή των υδάτινων πόρων και στη διαχείριση των φυσικών πόρων γενικότερα.
- Διαχείριση κτηματολογικών και πολεοδομικών δεδομένων για τη δημιουργία κτηματολογίου ή δασολογίου και για την πραγματοποίηση αναλύσεων που σχετίζονται με την αστική και περιφερειακή ανάπτυξη.
- Ανάλυση, σχεδιασμό και διαχείριση δικτύων, όπως είναι για παράδειγμα η εύρεση συντομότερων διαδρομών, ο σχεδιασμός δρομολογίων, η παρακολούθηση της κυκλοφορίας, η ομαδοποίηση των κόμβων ενός δικτύου έτσι ώστε αυτοί να εξυπηρετούνται από συγκεκριμένα σημεία (clustering and allocation problems).
- Εκτέλεση αναλύσεων και ερωτημάτων (queries) με βάση τα γεωγραφικά δεδομένα, όπως για παράδειγμα: ποιο κατάστημα είναι το πιο κερδοφόρο ανά περιοχή, πόσο κοντά είναι οι ανταγωνιστές; Πού είναι οι κατάλληλες θέσεις για την κατασκευή νέου καταστήματος; ποιος είναι ο συνολικός αριθμός πελατών σε απόσταση 10 χιλιομέτρων; ποια είναι η συντομότερη διαδρομή για την εκτέλεση των παραδόσεων; σε ποια σημεία ενός δικτύου εντοπίζονται δυσλειτουργίες; πόσοι οι υποψήφιοι πελάτες σε μια περιοχή; ποιο είναι το πιο κοντινό νοσοκομείο από ένα ατύχημα; ποια χωριά βρίσκονται σε απόσταση μικρότερη των 5 χιλιομέτρων από την εστία μίας δασικής πυρκαγιάς;

- Οπτική απεικόνιση των επιχειρησιακών δεδομένων σε χάρτες και παραγωγή γραφημάτων που μπορούν να υποστηρίξουν τη συγκέντρωση απολογιστικών στοιχείων, όπως για παράδειγμα συμβαίνει με τη γεωγραφική κατανομή των πωλήσεων, του κόστους ή της πελατειακής βάσης.

5.6 Συμπέρασμα

Τα γεωγραφικά συστήματα πληροφοριών διαθέτουν ένα ευρύτατο πεδίο εφαρμογών. Οι πιο συνηθισμένες εφαρμογές τους είναι στη ψηφιακή χαρτογράφηση, στην αποτύπωση των υποδομών οργανισμών τηλεπικοινωνιών, ύδατος και ηλεκτρικής ενέργειας, στη διαχείριση του φυσικού περιβάλλοντος, στη διαχείριση της κυκλοφορίας, αλλά και στην παρακολούθηση των διαδρομών οποιουδήποτε οχήματος.

Όσον αφορά την εφοδιαστική αλυσίδα, η χρησιμότητα των GIS έχει αναγνωριστεί σε ποικίλα ζητήματα προγραμματισμού και διαχείρισης. Η παρακολούθηση του στόλου των οχημάτων μίας επιχείρησης για την εκτέλεση της διανομής των προϊόντων είναι η πιο συνηθισμένη εφαρμογή των γεωγραφικών συστημάτων πληροφοριών στο πεδίο των Logistics. Επιπλέον, αρκετές επιχειρήσεις έχουν γνωρίσει τα πλεονεκτήματα των GIS στο σχεδιασμό ή επανασχεδιασμό της εφοδιαστικής αλυσίδας, στον προγραμματισμό των παραδόσεων και τη δρομολόγηση του στόλου, καθώς και ως εργαλεία παραγωγής εποπτικών αναφορών προς της διοίκηση, χρήσιμων στη λήψη αποφάσεων.

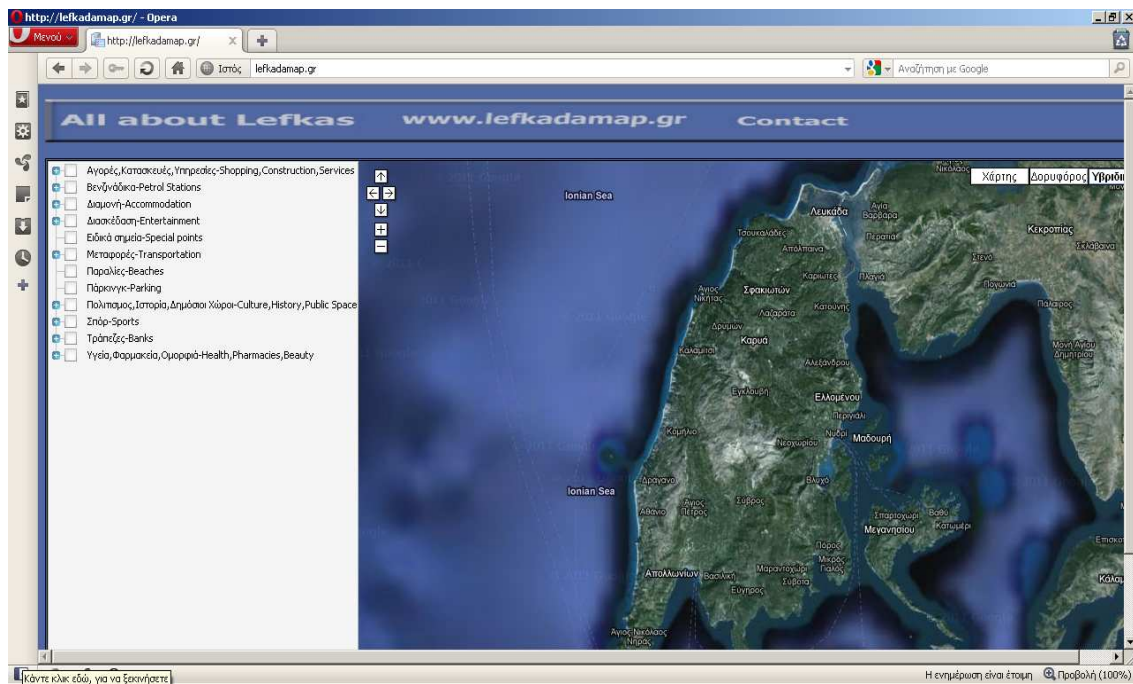
Παρόλα αυτά η χρήση των GIS σε θέματα Logistics των Ελληνικών επιχειρήσεων, υστερεί σε σχέση με την υπόλοιπη Ευρώπη. Εάν οι Ελληνικές επιχειρήσεις και οι οργανισμοί επιδιώξουν να γνωρίσουν τη λειτουργικότητα που προσφέρουν τα συστήματα GIS, ενδεχομένως να εντοπίσουν δυνατότητες εφαρμογής τους στην υποστήριξη συγκεκριμένων προβλημάτων που τους απασχολούν και τελικά να μπορέσουν να εκμεταλλευτούν τα πλεονεκτήματα της τεχνολογίας των συστημάτων αυτών, για να βελτιώσουν την απόδοση της εφοδιαστικής τους αλυσίδας.

ΚΕΦΑΛΑΙΟ 6

6.1 Πλοήγηση στην εφαρμογή

Στις επόμενες ενότητες θα γίνει μια αναλυτική περιγραφή της εφαρμογής. Θα υπάρξει παρουσίαση με εικόνες των διαφόρων λειτουργιών που θα συναντήσει ο χρήστης.

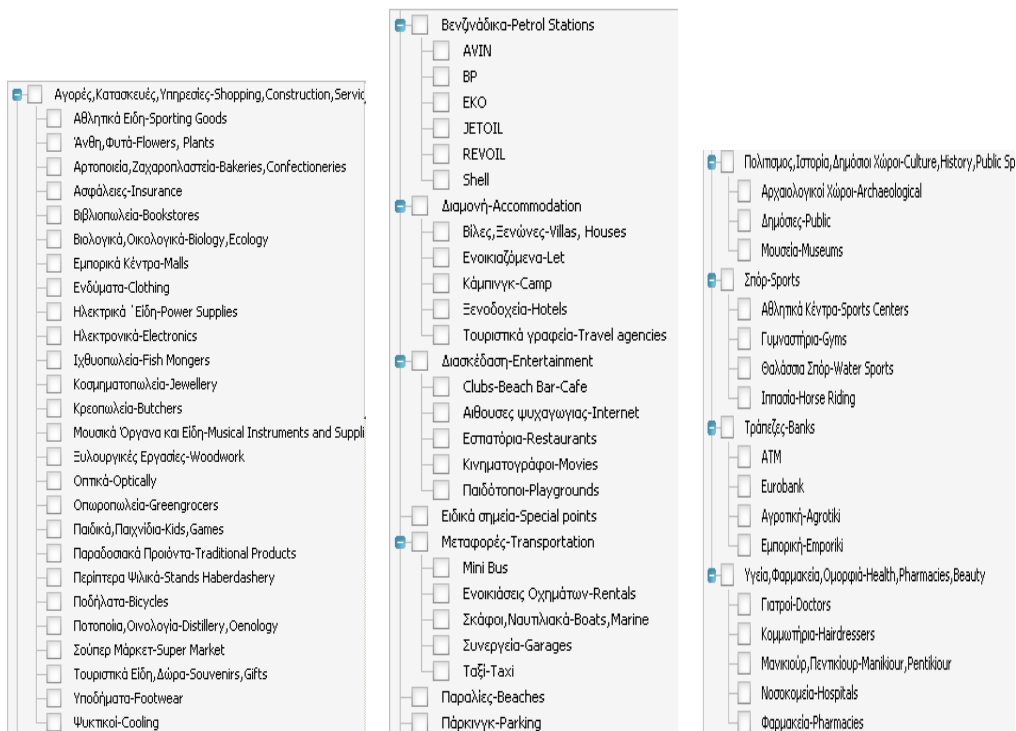
6.1.1 Αρχική οθόνη



Εικόνα : 14 Αρχική σελίδα www.lefkadamap.gr

Εμφανίζεται ο χάρτης της νήσου Λευκάδας με τη δενδροειδή δομή στα δεξιά. Η δενδροειδή δομή περιλαμβάνει ομάδες και υποομάδες ενδιαφερόντων σημείων(marker) οι οποίες υπόκεινται σε αλλαγές μέσω SQL.

6.1.2 Ενδιαφέροντα σημεία(ομάδες και υποομάδες)



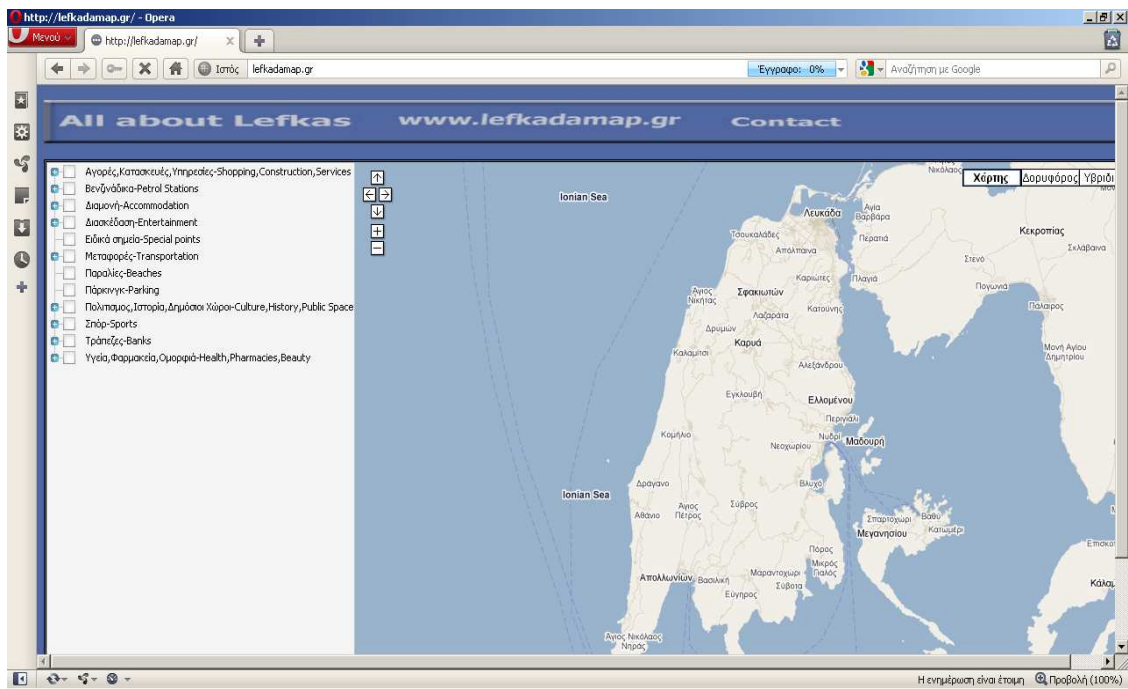
Εικόνα : 15 Ομάδες – Υποομάδες - Σημεία

6.1.3 Δυνατότητες Εφαρμογής

Χρησιμοποιώντας το Google Map Api και ενσωματώνοντας τις λειτουργίες του στη διεπιφάνεια, ο χρήστης έχει μια άμεση προβολή του χάρτη και των ενδιαφερόντων σημείων σε αυτόν. Η προβολή του χάρτη μπορεί να αλλάξει χρησιμοποιώντας τις επιλογές που παρέχονται από το Google Map Api.

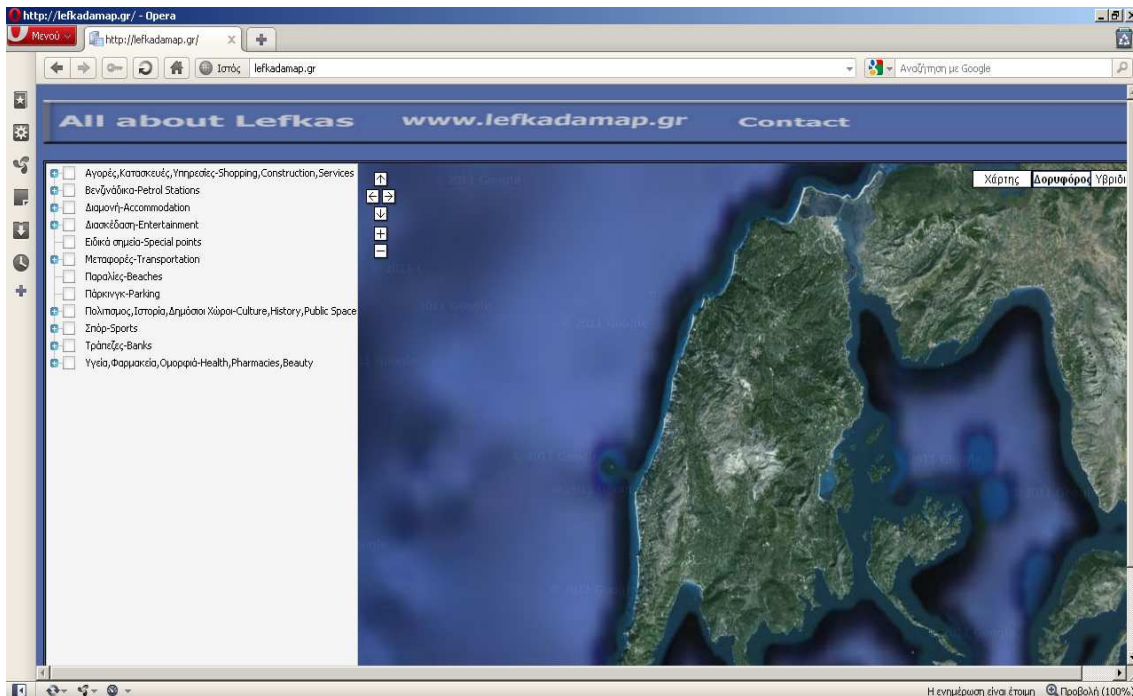
Συγκεκριμένα : Επιλέγοντας μια από τις τρεις επιλογές ο χάρτης προσαρμόζεται ανάλογα, σαν προεπιλεγμένη επιλογή είναι ο «Υβριδικός».

Χάρτης | Δορυφόρος | **Υβριδικός**



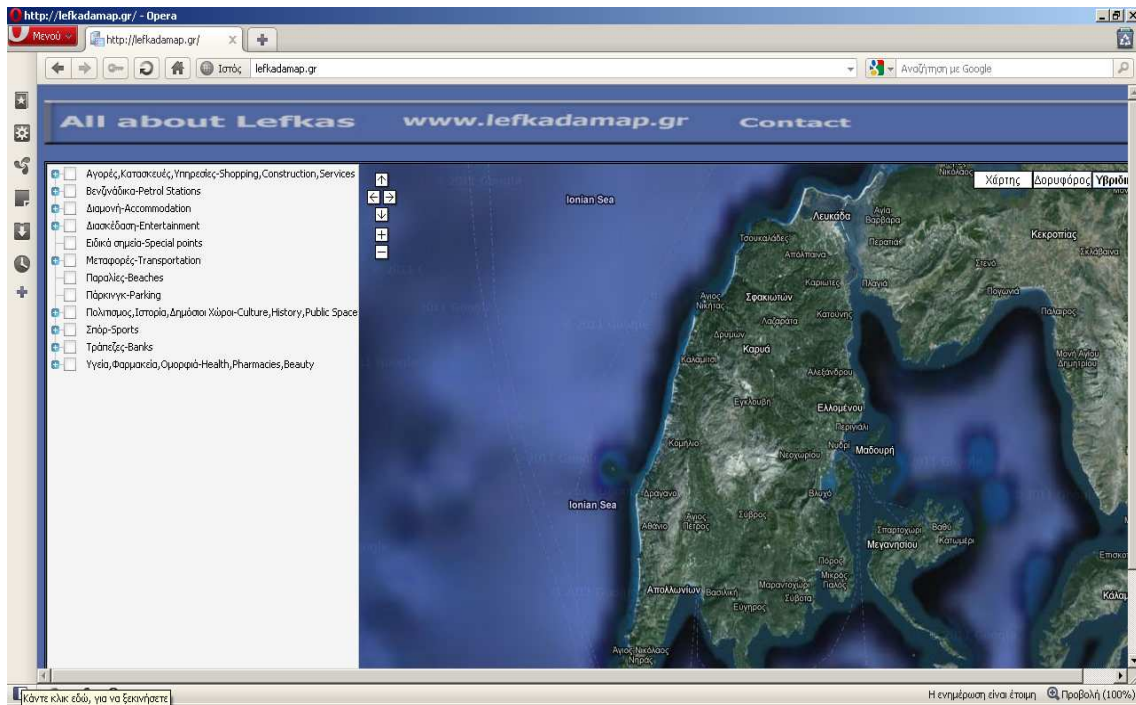
Εικόνα : 16 Επιλογή «Χάρτης»

Επιλογή «Χάρτης»:Ο χάρτης παίρνει τη μορφή προβολή χάρτη οδών που παρουσιάζεται παρακάτω, κάνοντας τη διάδραση με το χρηστή πιο λεπτομερή μας δίνει τη δυνατότητα να εστιάσουμε με λεπτομέρεια στο σημείο που μας ενδιαφέρει.



Εικόνα : 17 Επιλογή «Δορυφόρος»

Επιλογή «Δορυφόρος»:Ο χάρτης παίρνει τη μορφή προβολής εικόνας δορυφόρου που παρουσιάζεται παρακάτω χωρίς να εμφανίζει αναγραφόμενες περιοχές. Συνήθως το χρησιμοποιούμε για να έχουμε τη βέλτιστη απεικόνιση του εδάφους και της κλίσης που ενδεχόμενος να υπάρχει.



Εικόνα : 18 Επιλογή «Υβριδικός»

Επιλογή «Υβριδικός»:Ο χάρτης παίρνει τη μορφή προβολής εικόνας δορυφόρου που παρουσιάζεται παρακάτω αλλά εμφανίζει αναγραφόμενες περιοχές, ονόματα και οδούς βοηθώντας να κατανοήσουμε επιπλέον ποιες περιοχές βρίσκονται γύρω μας αλλά και παράλληλα να δούμε ονομασίες δρόμων .

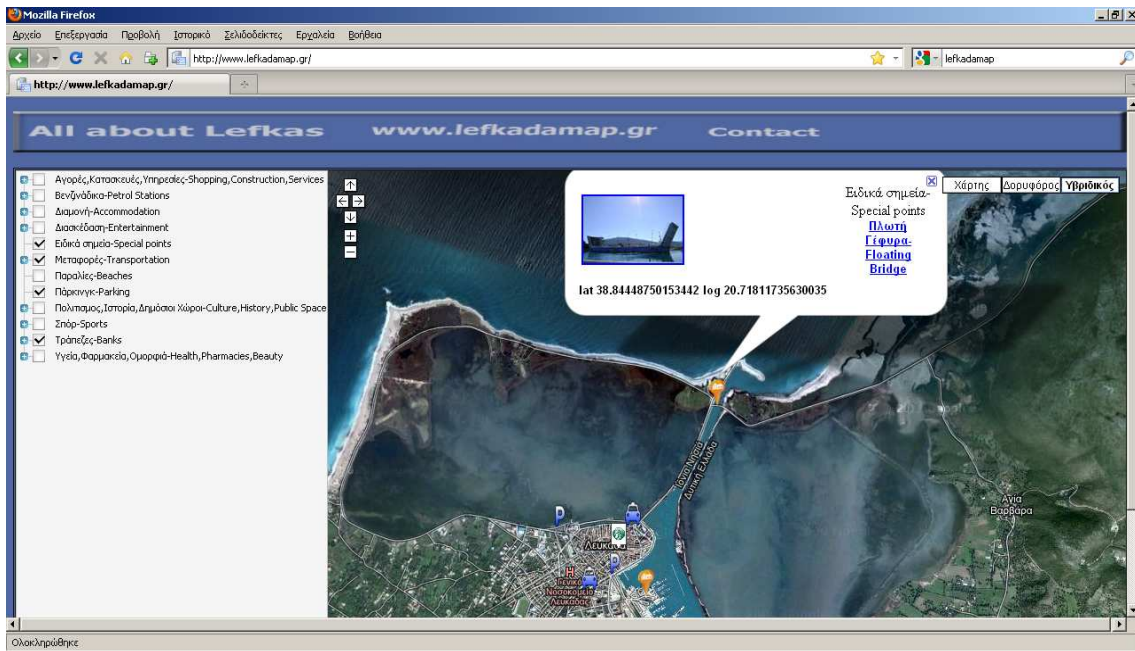
Ένα άλλο μενού πολύ χρήσιμο αλλά και διαδραστικό είναι αυτό που βρίσκεται πάνω αριστερά .



Εικόνα : 19 Μενού Πλοήγησης – Zoom in-out

Πατώντας κάθε ένα από τα βέλη ο χάρτης κινείται ανάλογα (Βόρεια, Νότια, Ανατολικά, Δυτικά) και χρησιμοποιώντας το αυξητικό(+) ή το μειωτικό(-) σύμβολο μεταβάλλουμε το zoom για να κατατοπισθούμε καλύτερα στο χάρτη. Αυτό έχει σαν αποτέλεσμα να μπορούμε να κατατοπισθούμε για την καλύτερη εύρεση του ενδιαφέροντος σημείου(marker).

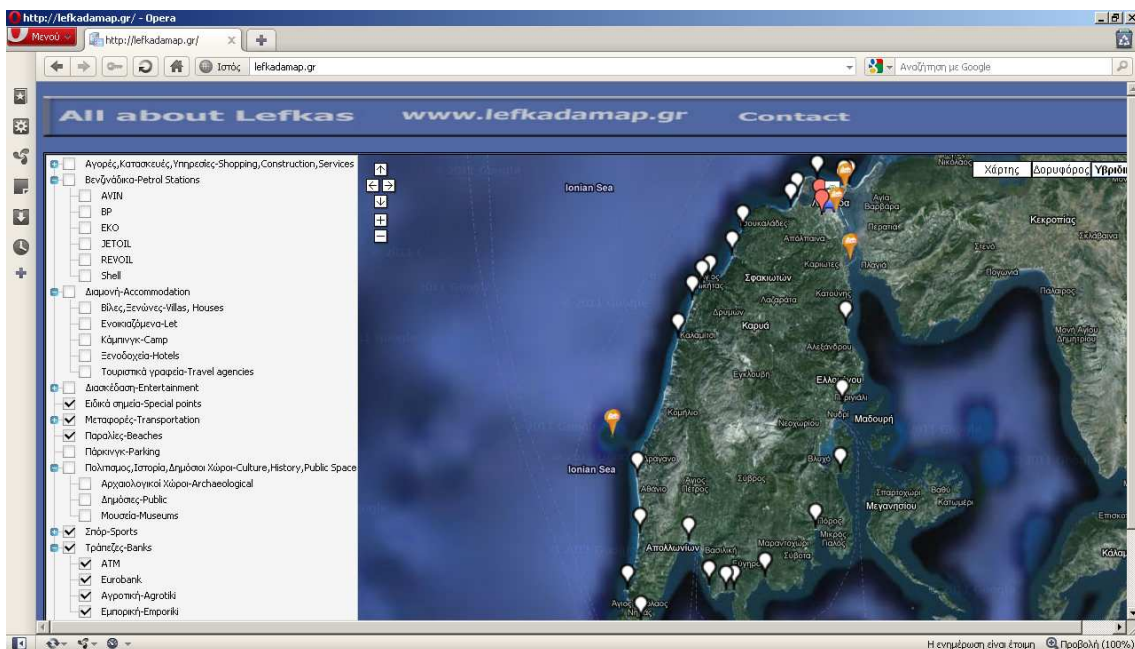
Πιο πάνω αναφέραμε το συγκεκριμένο μενού σαν διαδραστικό και αυτό έχει να κάνει με το ποντίκι. Μπορούμε δηλαδή να κάνουμε τα παραπάνω (μετακίνηση του χάρτη και zoom) χρησιμοποιώντας αριστερό κρατημένο click πάνω σε οποιοδήποτε σημείο του χάρτη για να τον μετακινήσουμε και τη ροδέλα του ποντικιού για zoom-in και zoom-out.



Εικόνα : 20 Zoom in-out

6.1.4 Δενδροειδή δομή

Όσον αφορά τις ομάδες και τις υποομάδες (ενδιαφέροντα σημεία) στα δεξιά, δίνεται στο χρήστη η δυνατότητα να επιλέξει με check box αυτήν που τον ενδιαφέρει και να προβληθεί στο χάρτη. Η προβολή του σημείου ενδιαφέροντος γίνεται με ένα μικρό εικονίδιο πάνω στο χάρτη το οποίο έχει άμεση σχέση με το λογότυπο του οργανισμού ή του φορέα. Ακόμη χρησιμοποιούνται εικονίδια διαφορετικού χρώματος για κάθε κατηγορία, ούτως ώστε να μπορεί ο χρήστης να έχει καλύτερη απεικόνιση του σημείου που τον ενδιαφέρει σε σχέση με τα υπόλοιπα.



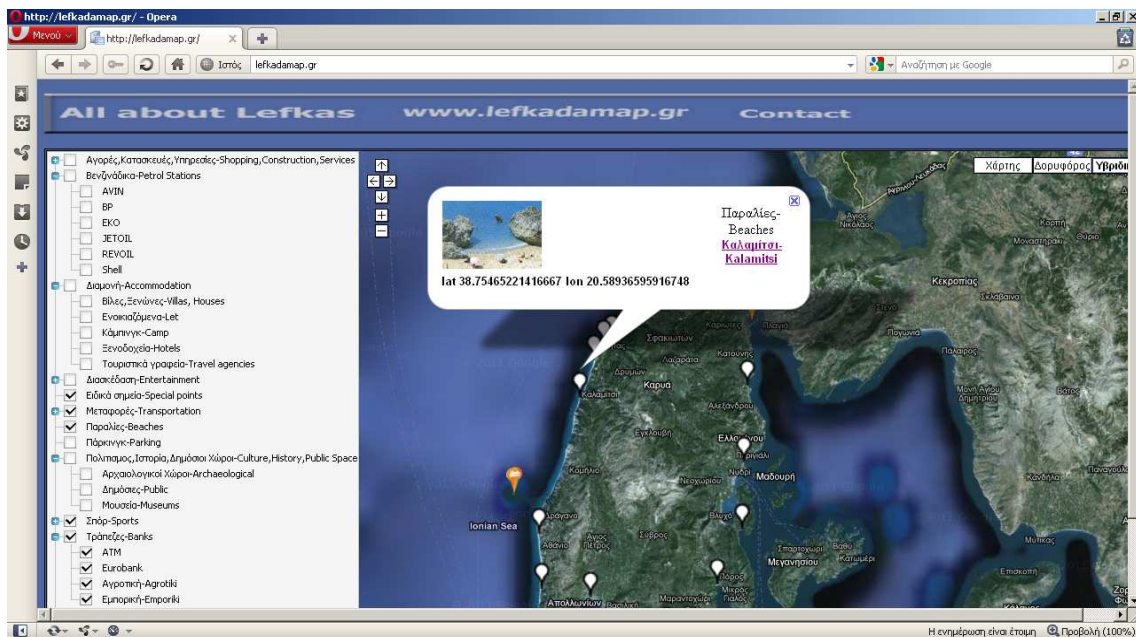
Εικόνα : 21 Δενδροειδή δομή

6.1.5 Επιλογή ενδιαφέροντος σημείου

Οι δείκτες εμφανίζονται στους Χάρτες Google για την επισήμανση μιας σημαντικής τοποθεσίας. Ο χρήστης αφού εντοπίσει το σημείο που τον ενδιαφέρει και επιλέγοντάς το κάνοντας click πάνω στο αντίστοιχο εικονίδιο, η διεπιφάνεια του παρέχει πληροφορίες μέσω ενός παράθυρου πληροφοριών «balloon» όπως διεύθυνση, εικόνα, διαδικτυακή διεύθυνση, γεωγραφικό πλάτος και μήκος του σημείου.

Στο παρακάτω παράδειγμα έχουμε επιλέξει από τη δενδροειδή δομή τα «Παράλιες» και σαν σημείο το Καλαμίτσι.

Τα παράθυρα πληροφοριών «balloon» εμφανίζονται όταν κάνετε κλικ σε ένα δείκτη και εμφανίζουν πρόσθετες πληροφορίες σχετικά με την τοποθεσία φωτογραφία και link.



Εικόνα : 22 Επιλογή ενδιαφέροντος σημείου

6.1.6 Εκτύπωση χαρτών

Μπορείτε να εκτυπώσετε χάρτες και πληροφορίες σχετικά με χάρτες που εμφανίζονται στην εφαρμογή. Για να το κάνετε αυτό, κάντε κλικ στην επιλογή Αρχείο/Εκτύπωση. Εμφανίζεται μια σελίδα εκτύπωσης με πρόσθετες επιλογές.

ΚΕΦΑΛΑΙΟ 7

7.1 Υλοποίηση Εφαρμογής

Για τη παραπάνω διεπιφάνεια χρησιμοποιήθηκαν τα ακόλουθα προγράμματα:

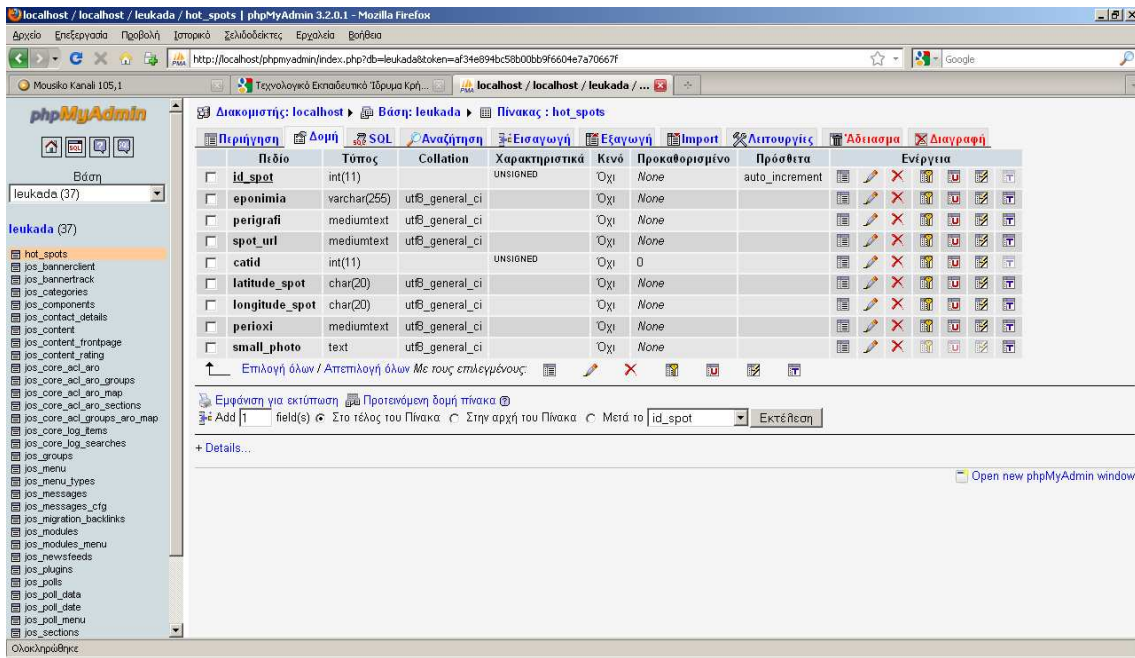
- Adobe Dreamweaver CS 3 για την κατασκευή java-scripts,css και τη δημιουργία PHP αρχείων αλλά και του τελικού αρχείου HTML.
- Adobe Photoshop CS 4 για την επεξεργασία των εικονιδίων του κάθε ενδιαφέροντος σημείου.
- WAMPServer για την υποστήριξη της διεπιφάνειας στο διαδύκτυο.
- PHPMyADMIN χρησιμοποιώντας την SQL του WAMPServer για την κατασκευή δυο πινάκων.

7.2 Περιγραφή – Κατασκευή βάσης δεδομένων της διεπιφάνειας

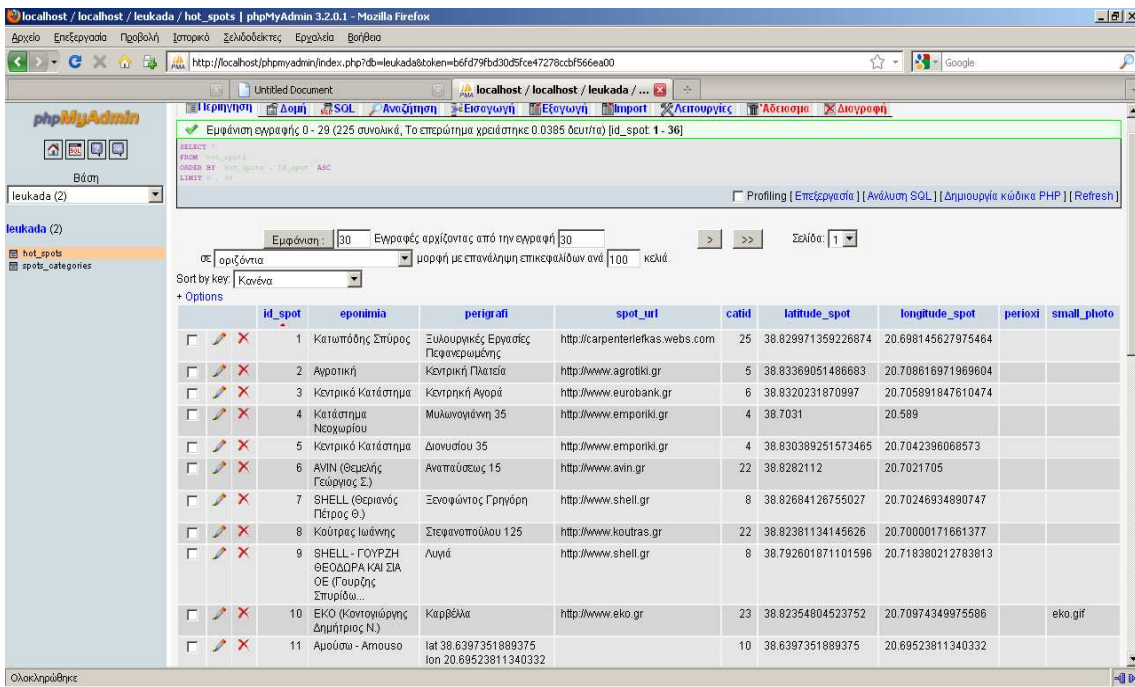
Δημιουργήθηκε η βάση δεδομένων **leukada** που αποτελείται από τους πίνακες **hot_spots** και **spots_categories**.

- Ο πίνακας **hot_spots** περιέχει την πληροφορία για κάθε σημείο ενδιαφέροντος που εμφανίζεται στο χάρτη της διεπιφάνειας και αποτελείται από τα εξής πεδία:

id_spot : το id του σημείου
eponimia : το όνομα (πχ τίτλος επιχείρησης) του σημείου
perigrifi : βοηθητικό πεδίο, το χρησιμοποιώ για την διεύθυνση
spot_url : η διαδικτυακή διεύθυνση του σημείου
catid : η κατηγορία στην οποία ανήκει το σημείο (εδώ βάζουμε το id της κατηγορίας/υποκατηγορίας σύμφωνα με την αρίθμηση του πίνακα spots_categories)
latitude_spot : το γεωγραφικό πλάτος του σημείου
longitude_spot : το γεωγραφικό μήκος του σημείου
perioxi : η ευρύτερη περιοχή στην οποία ανήκει το σημείο ενδιαφέροντος small_photo
small_photo : μικρή φωτογραφία του σημείου ενδιαφέροντος για να εμφανίζεται στο ballon με τις πληροφορίες όταν ο χρήστης κάνει κλικ πάνω στο σημείο. Το μόνο που χρειάζεται εδώ είναι το όνομα του αρχείου και όχι όλο το path (πχ simeio.jpg). Οι φωτογραφίες των σημείων θα πρέπει βέβαια πριν να έχουν τοποθετηθεί στον φάκελο spot_photos.



Εικόνα : 23 Δομή πίνακα hot_spots

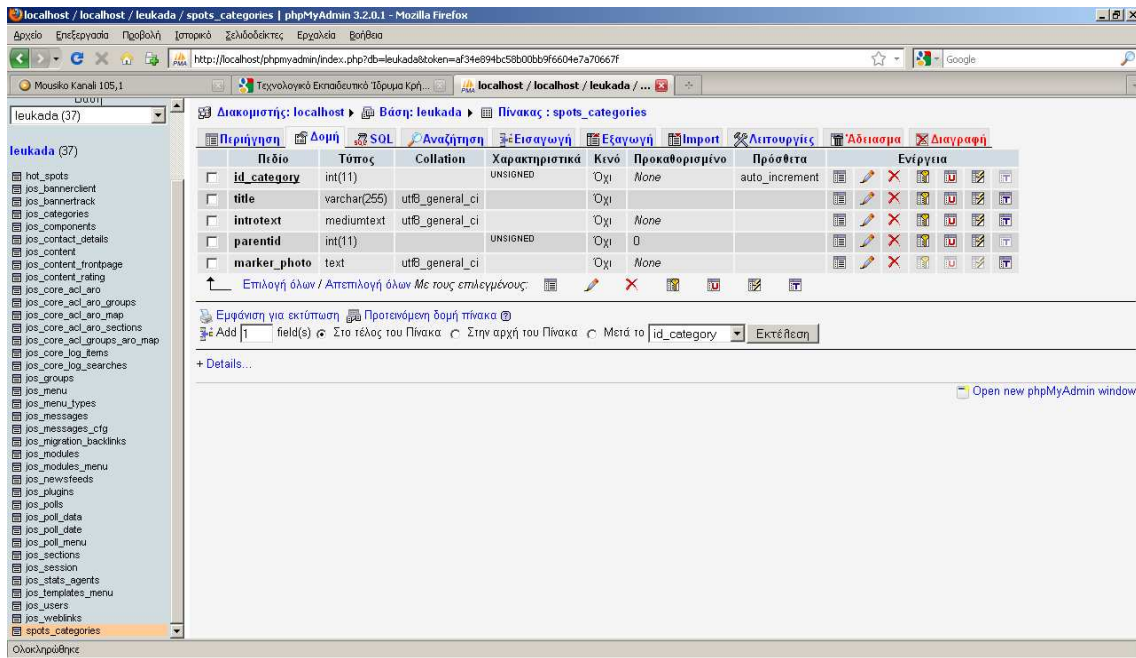


Εικόνα : 24 Ο πίνακας hot_spots

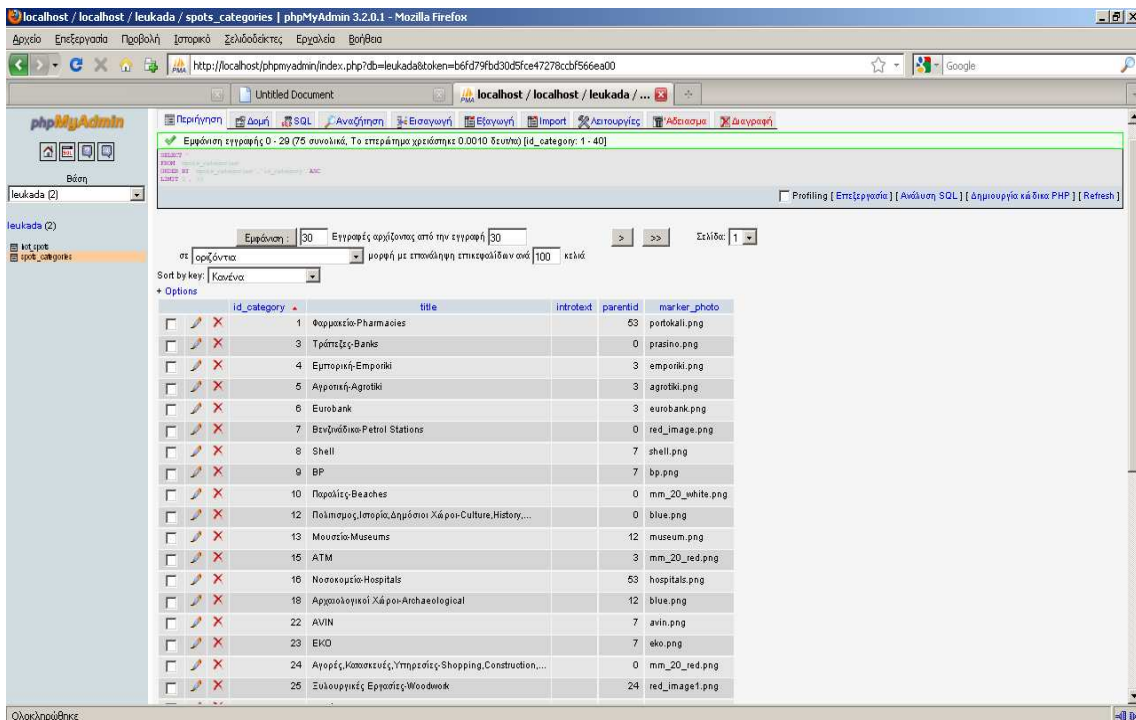
- Ο πίνακας spots_categories περιέχει την πληροφορία για κάθε κατηγορία/υποκατηγορία που εμφανίζεται στο treeview της διεπιφάνειας και αποτελείται από τα εξής πεδία:

id_category : το id της κατηγορίας/υποκατηγορίας
 title : το όνομα της κατηγορίας/υποκατηγορίας
 introtext : βοηθητικό πεδίο για τυχόν extra πληροφορίες

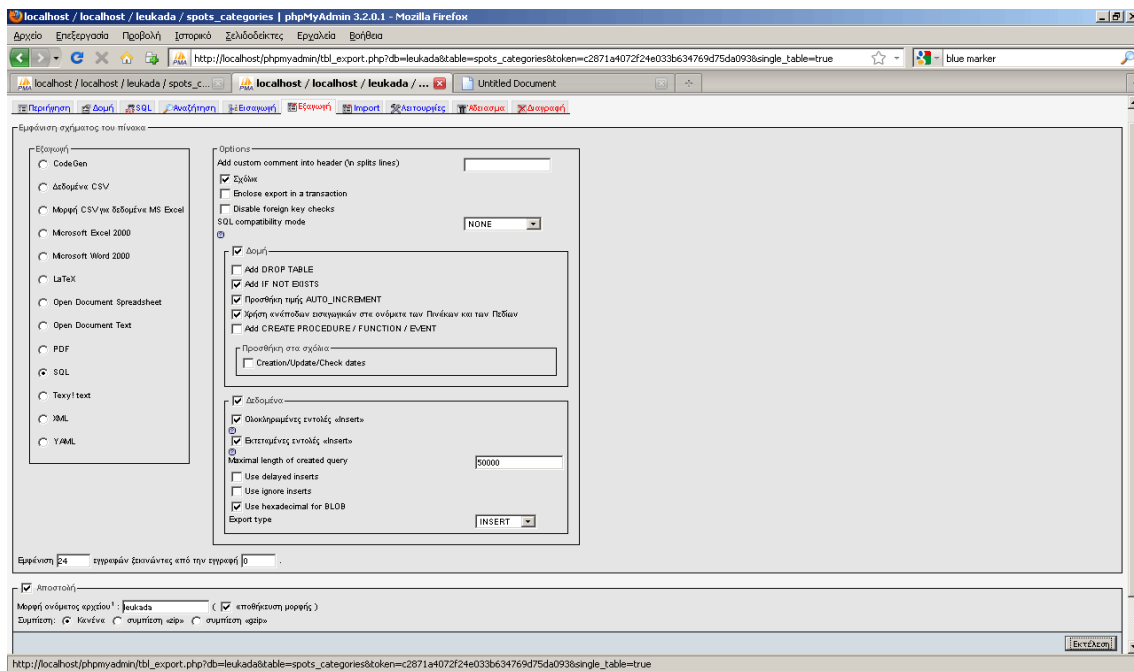
parentid : το id της γονικής κατηγορίας (το συμπληρώνουμε μόνο στις υποκατηγορίες για να δείξουμε σε ποια κατηγορία ανήκουν, αν πρόκειται για κύρια κατηγορία το parentid έχει την τιμή 0)
 marker_photo : το όνομα του αρχείου εικόνας που αναπαριστά την κατηγορία/ υποκατηγορία. Τα σχετικά εικονίδια θα πρέπει βέβαια πριν να έχουν τοποθετηθεί στον φάκελο markers_icons.



Εικόνα : 25 Δομή πίνακα hot_spots



Εικόνα : 26 Ο πίνακας spots_categories



Εικόνα : 27 Εξαγωγή των δεδομένων της βάσης

Έπειτα ακολουθεί η διαδικασία της εξαγωγής των δεδομένων της βάσης. Το αρχείο που δημιουργείται έχει κατάληξη .sql και περιέχει πληροφορίες για τη βάση δεδομένων **leukada** που αποτελείται από τους πίνακες **hot_spots** και **spots_categories**.

7.3 Κώδικας Εφαρμογής

7.3.1 Το αρχείο marker_functions3.js

Περιέχει όλες τις βοηθητικές javascript συναρτήσεις.

- Κατασκευάζω το marker και του δίνω χαρακτηριστικά

```
function create_marker(point, photo, category, keimeno) {

var red_marker = new GIcon();
red_marker.image = 'markers_icons/' + photo// red_image.png'
//red_marker.shadow = 'markers_icons/red_marker/red_shadow1.png';
red_marker.iconSize = new GSize(18,30);
red_marker.shadowSize = new GSize(36,30);
red_marker.iconAnchor = new GPoint(9,30);
red_marker.infoWindowAnchor = new GPoint(9,0);
```

- Στη συνέχεια φορτώνω όλα τα σημεία(marker) για να μην αργεί η διεπιφάνεια κατά το άνοιγμα. Τα στοιχεία των marker είναι αποθηκευμένα στη βάση μαζί με τα χαρακτηριστικά τους.

```
function load_all_markers(file){

xmlDoc=loadXMLDoc(file);

var markers= xmlDoc.documentElement.getElementsByTagName("marker");
```

- Δημιουργώ το balloon και του δίνω χαρακτηριστικά.

```
var keimeno_info_ballon= "<table width=300> <td width=30%>" + "<a href="" +
spot_url + "">" + "<img src='spot_photos/" + small_photo + ""></a>" + "</td> <td align=center>" +
title + "<br><font face='Arial' size= 2 >" + "<b><a href="" + spot_url + "">" + eponimia +
"</a></div>" + "</b>" + "</h5></font></td> </table>" + "<div align='left' style='border:0px solid
#BBBBBB; padding:5px 0px 5px 0px; width:300px;'> <font face='Arial' size= 2>" + "<b>" +
perigrifi + "</div></b></font>";
```

7.3.2 Το αρχείο fortwse_all_markers.php

Επικοινωνεί με τη βάση δεδομένων και μας επιστρέφει σε μορφή xml τις πληροφορίες για όλα τα σημεία ενδιαφέροντος που εμφανίζονται στο χάρτη.

```
//include XML Header (as response will be in xml format)
header("Content-type: text/xml");
//encoding may be different in your case
echo('<?xml version="1.0" encoding="ISO-8859-7"?>');
```

```
//start output of data
echo '<markers>';
```

```
//output data from DB as XML
```

```
$con = mysql_connect('localhost', 'root', ''); // edw vazei onoma xristi sti vasi kai kwdiko
```

```
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
```

```
mysql_query('set character set greek',$con);
mysql_query("SET perioxi,perigrifi 'greek'",$con);
```

```
mysql_select_db("leukada", $con);
```

```
$sql="SELECT * FROM hot_spots, spots_categories WHERE catid = id_category";
```

```
$res = mysql_query($sql);
if($res){
```

```
while($row=mysql_fetch_array($res)){
```

```
echo ("<marker id="" . $row['id_spot']. "">");
```

```
print("<latitude_spot><![CDATA[" . $row['latitude_spot']. "]]></latitude_spot>");
```

```
print("<longitude_spot><![CDATA[" . $row['longitude_spot']. "]]></longitude_spot>");
```

```
print("<category><![CDATA[" . $row['catid']. "]]></category>");
```

```
print("<eponimia><![CDATA[" . $row['eponimia']. "]]></eponimia>");
```

```
print("<marker_photo><![CDATA[" . $row['marker_photo']. "]]></marker_photo>");
```

```

        print("<perioxi><![CDATA[".$row['perioxi']."]]></perioxi>");
        print("<perigrafi><![CDATA[".$row['perigrafi']."]]></perigrafi>");
print("<title><![CDATA[".$row['title']."]]></title>");
print("<small_photo><![CDATA[".$row['small_photo']."]]></small_photo>");
        print("<spot_url><![CDATA[".$row['spot_url']."]]></spot_url>");

        echo ("</marker>");
    }
}else{
//error occurs
        echo mysql_errno().": .mysql_error()." at ".__LINE__." line in ".__FILE__." file<br>";
}
echo '</markers>';

mysql_close($con);?>

```

7.3.3 Το αρχείο vres_nodes.php που επικοινωνεί με τη βάση δεδομένων και μας επιστρέφει τις υποκατηγορίες για την επιλεγμένη κύρια κατηγορία.

```

<?php
$q=$_GET["q"];
error_reporting(E_ALL ^ E_NOTICE);
//include XML Header (as response will be in xml format)
header("Content-type: text/xml");
//encoding may be different in your case
echo('<?xml version="1.0" encoding="ISO-8859-7"?>');

//start output of data
echo '<markers>';

//output data from DB as XML

$con = mysql_connect('localhost', 'root', ''); // edw vazei onoma xristi sti vasi kai kwdiko

if (!$con)
{
die('Could not connect: ' . mysql_error());
}

mysql_query('set character set greek',$con);
mysql_query("SET perioxi,perigrafi 'greek",$con);
mysql_select_db("leukada", $con);
$sql="SELECT * FROM spots_categories WHERE parentid = ".$q."";
$res = mysql_query($sql);

if($res){
    while($row=mysql_fetch_array($res)){
        echo ("<marker id='".$row['id_category']."'>");
        echo ("</marker>");
    }
}
else{
//error occurs
        echo mysql_errno().": .mysql_error()." at ".__LINE__." line in ".__FILE__." file<br>";
}
}

```



```

}
echo '</markers>';
mysql_close($con);?>

```

7.3.4 Το αρχείο load_tree.php που επικοινωνεί με τη βάση δεδομένων και μας επιστρέφει την δενδροειδή δομή που θέλουμε σε κατηγορίες/ υποκατηγορίες σε xml format.

```

<?php
error_reporting(E_ALL ^ E_NOTICE);

//include XML Header (as response will be in xml format)
header("Content-type: text/xml");
//encoding may be different in your case
echo('<?xml version="1.0" encoding="ISO-8859-7"?>');

//start output of data
echo '<tree id="0">';

//output data from DB as XML

$con = mysql_connect('localhost', 'root', ""); // edw vazei onoma xristi sti vasi kai kwdiko
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_query('set character set greek',$con);
mysql_query("SET title,id_category 'greek'", $con);
mysql_select_db("leukada", $con);
$sql="SELECT * FROM spots_categories where parentid=0";
$res = mysql_query($sql);
if($res){
    while($row=mysql_fetch_array($res)){
        $side= $row['id_category'];
        if ($side == "0") {
            echo("<itemtext='".$row['title']."' id='".$row['id_category']."'></item>");
        }
        else
        {
            echo("<itemtext='".$row['title']."' id='".$row['id_category']."'>");
        }

        if ($side <> "0") {
            $sql1="SELECT * FROM spots_categories where parentid= '".$side."'";
            $res1 = mysql_query($sql1);
            if($res1){
                while($row1=mysql_fetch_array($res1)){
                    echo("<itemtext='".$row1['title']."' id='".$row1['id_category']."'></item>");
                }
            }
            echo("</item>");
        }
    }
}

```

```

    }
}else{
//error occurs
echo mysql_errno().": ".mysql_error()." at ".__LINE__." line in ".__FILE__." file<br>";
}
echo '</tree>';
mysql_close($con);
?>

```

7.3.5 Η σελίδα με τη διεπιφάνεια, leukada_markers.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script src="http://maps.google.com/maps?file=api&v=1&key=ABQIAAAA6motOPHuzoaZ3-
hG0HuqPhQhL6vn9PNcXa7eSPja3EjRwozD8hSIGKiE7dXRUi2Bd92ZXURIVMMjka"
type="text/javascript">
</scrip>

<!--<scriptsrc="http://gmaps-utility-
library.googlecode.com/svn/trunk/markermanager/release/src/markermanager.js"></script> -->

<!-- edw periexontai oi sinartiseis pou xrisimopoiw -->

<script src="marker_functions3.js"></script>
</head>
<!-- fortwnw olous tous markers me to anoigma tis selidas kai sti sinexeia tous krivw (auto ginetai gia
na fortwnontai mia fora stin arxi kai na min argei i efarmogi otan tsekarw kai ksetsekarw tis
kategories -->

<body onload='load_all_markers("fortwse_all_markers.php"); hide_all_markers();'>

<!-- ta 3 parakatw arxeia xreiazontai gia to treeview -->
<link rel="STYLESHEET" type="text/css" href="codebase/dhtmlxtree.css">
<script src="codebase/dhtmlxcommon.js"></script>
<script src="codebase/dhtmlxtree.js"></script>
<table>
<tr>
<td valign="top">
<!-- edw fortwnw to xarti -->
<div id="map" style="height: 600px; width: 1000px;"> </div>

</td>
<td valign="top">
<!-- edw fortwnw to treeview -->
<div id="treeboxbox_tree2" style="width:250px; height:600px;background-
color:#f5f5f5;border :1px solid Silver;; overflow:auto;"></div>

</td>
</tr>
</table>
<script>

```

```

function toncheck(id,state) {
    var rootsAr = tree2.getSubItems(0).split(",");
    var einai_parent=0;
    for (var i = 0; i < rootsAr.length; i++) {
        if (id==rootsAr[i]) { /// edw vriskw poia einai ta parent nodes
            einai_parent=1;
        }
    }
    if (einai_parent==1) {
        if (state ==1) { //elegxw an einai tsekarimeno to parent node
            show_category_markers("vres_nodes.php?q=" + id + "&sid=" + Math.random(),id);
        }
        else
        {
            hide_category_markers("vres_nodes.php?q=" + id + "&sid=" + Math.random(),id);
            //hide_spot_marker(id);
        }
    }
    else // edw vriskw poia einai ta child nodes
    {
        if (state ==1) { //elegxw an einai tsekarimeno to kathe child node
            show_spot_marker(id);
        }
        else
        {
            hide_spot_marker(id);
        }
    }
};
tree2 = new dhtmlXTreeObject("treeboxbox_tree2", "100%", "100%", 0);
tree2.setSkin('dhx_skyblue');
tree2.setImagePath("codebase/imgs/csh_bluebooks/");
tree2.enableCheckBoxes(1);
tree2.enableThreeStateCheckBoxes(true);
tree2.setOnCheckHandler(toncheck);

<!-- edw fortwnw to tis catigories sto tree -->
tree2.loadXML("load_tree.php");</script>
<script>
    var map = new GMap2(document.getElementById("map"));
    map.setCenter(new GLatLng(38.7050,20.6500),11);
    map.setMapType(G_HYBRID_MAP);
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    map.enableScrollWheelZoom();
</script>
<br />
</body>
</html>

```

7.3.6 Κώδικας SQL για κατασκευή βάσης με δυο πίνακες για την εφαρμογή.

```
phpMyAdmin SQL Dump
-- version 3.2.0.1
-- http://www.phpmyadmin.net
--
-- Σύστημα: localhost
-- Χρόνος δημιουργίας: 03 Ιούν 2011, στις 02:50 PM
-- Έκδοση Διακομιστή: 5.1.36
-- Έκδοση PHP: 5.3.0

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

-- Βάση: `leukada`

-- Δομή Πίνακα για τον Πίνακα `hot_spots`

CREATE TABLE IF NOT EXISTS `hot_spots` (
  `id_spot` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `eponimia` varchar(255) NOT NULL,
  `perigrafì` mediumtext NOT NULL,
  `spot_url` mediumtext NOT NULL,
  `catid` int(11) unsigned NOT NULL DEFAULT '0',
  `latitude_spot` char(20) NOT NULL,
  `longitude_spot` char(20) NOT NULL,
  `perioxi` mediumtext NOT NULL,
  `small_photo` text NOT NULL,
  PRIMARY KEY (`id_spot`),
  KEY `idx_catid` (`catid`),
  KEY `catid` (`catid`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=236 ;

-- 'Άδειασμα δεδομένων του πίνακα `hot_spots`

INSERT INTO `hot_spots` (`id_spot`, `eponimia`, `perigrafì`, `spot_url`, `catid`, `latitude_spot`,
`longitude_spot`, `perioxi`, `small_photo`) VALUES
(211, '+BIKE', 'Χαλκίας Σπύρος\r\nΠεφανερωμένης 6', '', 80, '38.8321935', '20.7023104', '', ''),
(143, 'ADANI HOTEL - ΕΠΙΧΕΙΡΗΣΕΙΣ ΒΕΡΓΙΝΗΣ ΑΕ', 'Γ Κατηγορία\r\nΤσουκαλάδες\r\nlat
38.82104459286924\r\nlon 20.65509617328643', '', 38, '38.82104459286924', '20.65509617328643',
'', ''),

-- Δομή Πίνακα για τον Πίνακα `spots_categories`
--

CREATE TABLE IF NOT EXISTS `spots_categories` (
  `id_category` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `title` varchar(255) NOT NULL DEFAULT '',
  `introtex` mediumtext NOT NULL,
  `parentid` int(11) unsigned NOT NULL DEFAULT '0',
  `marker_photo` text NOT NULL,
  PRIMARY KEY (`id_category`),
  KEY `id` (`id_category`)
```

```
) ENGINE=MyISAM      DEFAULT CHARSET=utf8  ROW_FORMAT=DYNAMIC  
AUTO_INCREMENT=86 ;
```

```
--
```

```
-- 'Άδειασμα δεδομένων του πίνακα `spots_categories`
```

```
--
```

```
INSERT INTO `spots_categories` (`id_category`, `title`, `introtext`, `parentid`, `marker_photo`)  
VALUES  
(15, 'ATM', "", 3, 'mm_20_red.png'),  
(22, 'AVIN', "", 7, 'avin.png'),  
(9, 'BP', "", 7, 'bp.png'),
```

ΚΕΦΑΛΑΙΟ 8 – ΕΠΙΛΟΓΟΣ

8.1 Αποτελέσματα

Τα αποτελέσματα μετά το τέλος της πτυχιακής είναι:

- Μετά από την έρευνά μας γύρω από το διαδίκτυο και την κατασκευή ιστοσελίδων μάθαμε για το ποιες γλώσσες προγραμματισμού χρησιμοποιούνται για την κατασκευή τους.
- Μάθαμε πώς να φτιάχνουμε ιστοσελίδες με PHP και να συνδυάζονται δυναμικά με βάσεις δεδομένων.
- Μελετήσαμε τις ενότητες που χρησιμοποιούνται μέσα από το Google και το Google Map Api και πόσο σημαντικές είναι για την κατασκευή ιστοσελίδας προβολής με χάρτη .
- Χρησιμοποιήσαμε το photoshop για να φτιάξουμε την εικόνα της κεφαλίδας μας και τις υπόλοιπες φωτογραφίες του ιστοτόπου και μπορέσαμε να κατανοήσουμε πώς λειτουργεί το πρόγραμμα αυτό.
- Τέλος είδαμε για το πώς δομείτε μια δενδροειδή δομή, τι πεδία χρειάζονται για να θεωρείτε ολοκληρωμένη και πώς συνδυάζεται με τις βάσεις δεδομένων δυναμικά.

8.2 Συμπεράσματα

Η δυναμική δημιουργία του προγράμματος βοηθάει τον χρήστη να βρίσκει σε λιγότερο χρόνο τα ενδιαφέροντα σημεία που τον ενδιαφέρουν χρησιμοποιώντας τις δυνατές επιλογές που του δίνονται. Επίσης ο διαχειριστής του προγράμματος θα μπορεί εύκολα να αλλάζει τα σημεία και να τα προβάλλει στο χάρτη.

Το κομμάτι της SQL θα βρίσκεται σε κάποιο server (και στο διαδίκτυο) ώστε να έχει πρόσβαση οποιοσδήποτε έχει τα δικαιώματα σε αυτόν, ώστε να μπορεί να κάνει τις απαραίτητες αλλαγές των σημείων (πρόσθεση-αφαίρεση) ,όπως και το κομμάτι της PHP αντίστοιχα.

Επίσης και το κομμάτι που αναφέρεται στην προβολή των σημείων και του χάρτη (HTML)πρέπει να βρίσκεται σε ένα server (web) ώστε να μπορούν να έχουν πρόσβαση οι χρήστες.

8.3 Εξέλιξη Συστήματος

Είναι γνωστό ότι όλες οι σελίδες που βρίσκονται στο διαδίκτυο, αλλά και γενικότερα όλα τα προγράμματα που υπάρχουν, μπορούν να δεχτούν βελτιώσεις. Έτσι και η παρούσα πτυχιακή εργασία μπορεί να εξελιχτεί.

Καταρχήν, μπορούμε να αναφερθούμε στη διαμόρφωση της αρχικής σελίδας που βλέπει ο επισκέπτης. Θα μπορούσε να γίνει πιο φιλική και ευχάριστη προς τους χρήστες, αλλάζοντας το interface, βοηθώντας τον χρήστη στις επιλογές του. Θα μπορούσε να υπάρχει αναζήτηση για υποστήριξη και άλλων περιοχών πέραν της νήσου Λευκάδας όπως και η ανάλογη λίστα των περιοχών για την SQL ή ακόμη και να παίρνουμε περισσότερες πληροφορίες για το κάθε σημείο (εικόνα-street view,video σε μεγαλύτερη οθόνη).

Η εφαρμογή θα μπορούσε να είναι συμβατή και με κινητές συσκευές με μικρή οθόνη. Θα μπορούσαμε να κάνουμε περισσότερους ελέγχους, αλλά και περιορισμούς όσον αφορά τη δενδροειδή δομή για να μπορεί ο χρήστης, όταν διαλέγει την ομάδα ή την υποομάδα που θέλει, να του εμφανίζονται τα σημεία ομαδοποιημένα και να μην πέφτουν πολλά «marker» το ένα πάνω στο άλλο. Με αυτόν τον τρόπο, θα ήταν πιο γρήγορη και εύκολη η επιλογή των σημείων και εμφάνιση τους στο χάρτη.

Επίσης θα μπορούσαμε να είχαμε και άλλη μια εφαρμογή PHP, στην οποία θα είχε πρόσβαση μόνο ο διαχειριστής. Θα ήταν ουσιαστικά μια φόρμα εισαγωγής δεδομένων (marker) αλλά και διαγραφής. Έτσι λοιπόν επιλέγοντας κάποιο σημείο να το «διαγράψει» ή να το «μεταφέρει» σε κάποια άλλη γεωγραφική θέση, να μπορεί να αλλάζει δυναμικά και το SQL. Με αυτό τον τρόπο, θα ήταν πιο εύκολη η επεξεργασία (αλλαγή, διαγραφή,πρόσθεση) των σημείων.

8.4 Πρακτική αξιοποίηση

Σκέψου όλες εκείνες τις παραλίες που θέλεις να πας και μόλις πάτησες το πόδι σου στο νησί, τις έχει ξεχάσει. Το www.lefkadamar.gr μπορεί να σε βοηθήσει να οργανώσεις τις διακοπές σου με τα εργαλεία που σου προσφέρει. Μπορείς να δεις τα σημεία ενδιαφέροντος όλου του νησιού. Παραλίες, φαγητό, διαμονή, διαδρομές και οτιδήποτε άλλο μπορεί να φανεί χρήσιμο στη περιήγησή σου στο νησί. Μπορείς να εκτυπώσεις το χάρτη!

Εάν ξεχάσεις κάποια παραλία ενώ βρίσκεσαι στις διακοπές σου και από mobile-internet έχεις τη δυνατότητα να βρεις αυτό που σε ενδιαφέρει.

Βιβλιογραφία

[1] Ανάπτυξη web εφαρμογών με Php και MySQL Luke Welling Laura Thomson εκδόσεις Μ.Γκιούρδας.

[2] Πλήρες εγχειρίδιο της JavaScript Martin Webb, Michel Plungjan εκδόσεις Μ. Γκιούρδας.

[3] *Adobe* Design Center
<http://www.adobe.com>

[4] Google Maps API Family
<http://code.google.com/intl/el-GR/apis/maps/index.html>

[5] Sign Up the Google Maps API
<http://code.google.com/intl/el-GR/apis/maps/signup.html>

[6] Using Databases With Gmaps Apps
<http://groups.google.com/group/Google-Maps-API/web/using-databases-with-gmaps-apps>

[7] PHP/MySQL with Google Geo APIs
<http://code.google.com/intl/el-GR/apis/maps/articles/phpsqlgeocode.html>

[8] http://en.wikipedia.org/wiki/Google_Maps

[9] <http://en.wikipedia.org/wiki/JavaScript>

[10] <http://maps.google.com/>

[11] <http://code.google.com/apis/maps/>

[12] <http://maps.google.com/support/>

[13] Όροι υπηρεσίας χαρτών Google http://maps.google.com/intl/el/help/terms_maps.html

Παρουσίαση Πτυχιακής Εργασίας

**Τουριστική Παρουσίαση της Λευκάδος μέσω
του Google Earth**

**Ευστάθιος Κατωπόδης
(ΑΜ: 1197)**

Περιεχόμενα

- **Εισαγωγή.**
- **Περιγραφή των βασικότερων τεχνικών διόρθωσης της εφαρμογής.**
- **Ανάλυση της προτεινόμενης εφαρμογής.**
- **Παρουσίαση εφαρμογής.**
- **Συμπεράσματα – Εξέλιξη εφαρμογής.**

Εισαγωγή

- Απεικόνιση των σημείων ενδιαφέροντος στο χάρτη.
- Χρησιμοποιεί τη διεπαφή προγραμματισμού εφαρμογών της Google, Google Maps API.
- Τα δεδομένα του σημείου αποθηκεύονται σε αρχείο SQL.
- Δυναμική κατανομή και τοποθέτηση αντικειμένων στο χάρτη.

Γεωγραφικά συστήματα πληροφοριών GIS

- Εργαλείο χαρτογράφησης και ανάλυσης των στοιχείων.
- Ολοκληρώνει τις λειτουργίες των εφαρμογών βάσεων δεδομένων.
- Αναλύει τα χωρικά δεδομένα.
- Προσδιορίζουν τις διαθέσιμες πληροφορίες στο γεωγραφικό χώρο (γεωκωδικοποίηση).

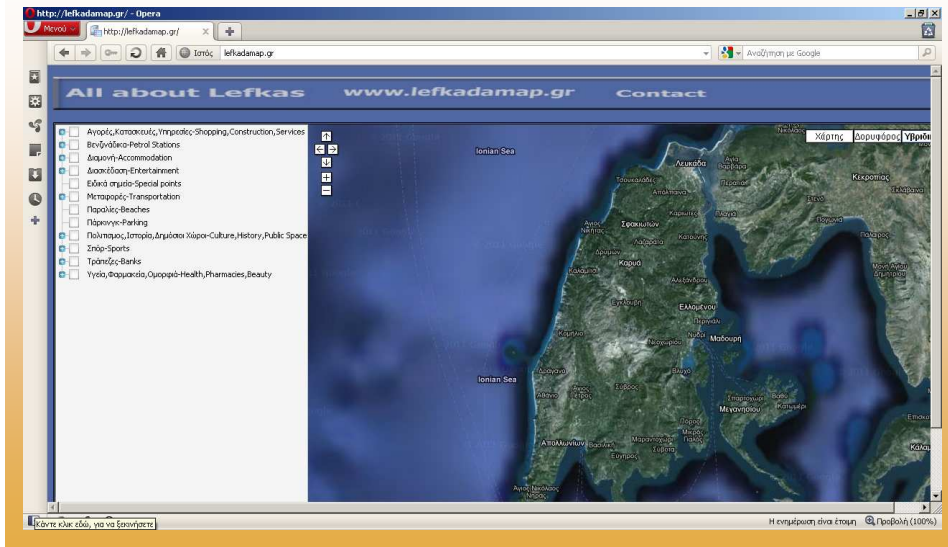
Περιγραφή των βασικότερων τεχνικών διόρθωσης της εφαρμογής

- Για την υλοποίηση της εφαρμογής, έχουν χρησιμοποιηθεί HTML,PHP,CSS,SQL.
- Το κάθε σημείο (marker) αποτελείται από lon & lat καθώς και άλλα χαρακτηριστικά που αποθηκεύονται σε SQL.
- Στην SQL υπάρχουν δύο πίνακες που υποδηλώνουν κατηγορίες-υποκατηγορίες.
- Τα marker εμφανίζονται στο χάρτη μετά την επιλογή κατηγορίας-υποκατηγορίας.

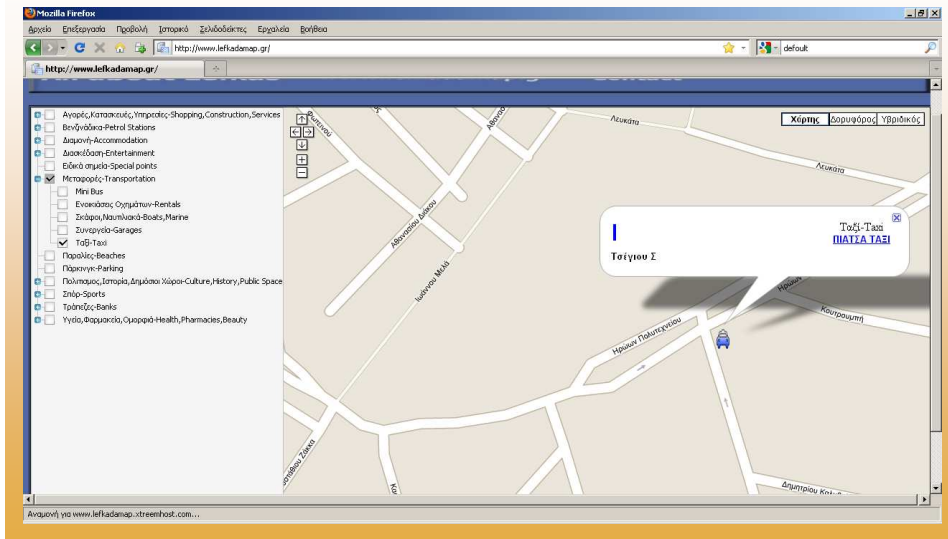
Περιγραφή των βασικότερων τεχνικών διόρθωσης της εφαρμογής

- Τα marker διαδρούν με το χρήστη και παρέχονται πληροφορίες σε balloon (link,photo,lon,lat).
- Ο χάρτης προσαρμόζεται ανάλογα «Χάρτης-Δορυφόρος-Υβριδικός».
- Ο χρήστης : μετακινεί το χάρτη,επιλογή zoom.

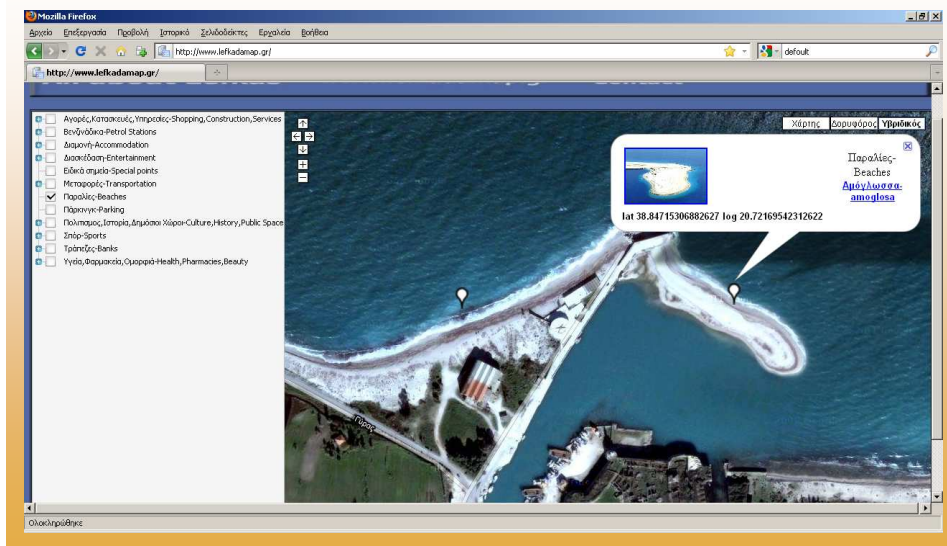
Ανάλυση της προτεινόμενης εφαρμογής



Ανάλυση της προτεινόμενης εφαρμογής



Ανάλυση της προτεινόμενης εφαρμογής



Παρουσίαση εφαρμογής

www.lefkadamap.gr

Συμπεράσματα

- Ο χρήστης βρίσκει σε λιγότερο χρόνο τα ενδιαφέροντα σημεία χρησιμοποιώντας τις δυνατές επιλογές που του δίνονται.
- Όλες οι αλλαγές στις οποίες υπόκεινται τα marker, γίνονται από την sql.
- Ο χάρτης είναι διαδραστικός και μεταβάλλεται ανάλογα.
- Η εφαρμογή εστιάζει στη προβολή ομαδοποιημένων κατηγοριών.

Εξέλιξη εφαρμογής

- Υποστήριξη και άλλων περιοχών.
- Περαιτέρω απεικόνιση ενός σημείου με άλλα marker πχ.Α.Τ.Ε.Ι., γύπεδα , μεγάλοι χώροι , πάρκα , ζωολογικοί κήποι κ.τ.λ.

Ευχαριστώ για το χρόνο σας.

