



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
UNIVERSITY OF PELOPONNESE

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΡΟΗΓΜΕΝΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΔΙΚΤΥΑ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΡΟΗΓΜΕΝΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ
ΤΟΥΡΙΣΤΙΚΗΣ ΠΥΛΗΣ ΒΑΣΕΙ ΤΕΧΝΟΛΟΓΙΩΝ WEB 2.0**

ΒΟΡΓΙΑΣ ΓΕΩΡΓΙΟΣ

ΑΜ: 2008105

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΤΣΕΛΙΚΑΣ ΝΙΚΟΛΑΟΣ

Περίληψη

Αντικείμενο της διπλωματικής εργασίας είναι η ανάπτυξη ενός τουριστικού portal, το οποίο θα παρέχει πληροφορίες στους χρήστες τέτοιες ώστε να μπορούν να οργανώσουν καλύτερα την επίσκεψή τους στη γεωγραφική περιοχή την οποία καλύπτει το portal. Οι πληροφορίες της περιοχής προβάλλονται σε δυναμικό χάρτη, ο οποίος βασίζεται στο χαρτογραφικό υπόβαθρο της Google. Η ανάπτυξη της εφαρμογής βασίζεται στη χρήση ελεύθερου λογισμικού και για τη βάση δεδομένων αλλά και για τη διαχείριση του περιεχομένου της ιστοσελίδας. Επίσης γίνεται χρήση κώδικα που βασίζεται στις γλώσσες προγραμματισμού PHP και Javascript.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	7
1.1	Σκοπός και στόχος της Διπλωματικής	7
1.2	Δομή της Διπλωματικής	7
2	WEB 2.0 Εφαρμογές και ιστορία	9
2.1	Ιστορική αναδρομή του παγκοσμίου Ιστού	9
2.2	WEB 2.0 – Η νέα γενιά του Παγκόσμιου Ιστού	9
2.3	Βασικές αρχές WEB 2.0.....	11
2.4	Mashup εφαρμογές	12
2.4.1	Ορισμός	12
2.4.2	WEB APIS	14
3	Τεχνολογίες ανάπτυξης της εφαρμογής	15
3.1	Βάσεις δεδομένων	15
3.2	Τεχνολογίες AJAX	16
3.3	Γλώσσες σήμανσης ερωτημάτων.....	17
3.3.1	HTML	18
3.3.2	XML	18

3.3.3	CSS	18
3.4	Γλώσσες Scripting	19
3.4.1	Java Script	19
3.4.2	PHP.....	21
4	Υλοποίηση παρουσίαση της εφαρμογής	22
4.1	Παρουσίαση portal	22
4.1.1	Σελίδα του δυναμικού χάρτη.	23
4.1.2	Εισαγωγή δυναμικού περιεχομένου στο χάρτη.....	27
4.2	Σύστημα κράτησης δωματίων	29
4.2.1	Σχεδιασμός της Βάσης δεδομένων	29
4.2.2	Σελίδα κράτησης δωματίου	31
5	Διαχείριση περιεχομένου του portal	33
5.1	CMS JOOMLA!	33
5.2	Διαχειριστική σελίδα	34
6	Σενάριο χρήσης	38
7	Επίλογος – Συμπεράσματα	44
	Αναφορές - Βιβλιογραφία	45
	Παράρτημα – Πηγαίος κώδικας	46

Κεφάλαιο 1: Εισαγωγή

Τα τελευταία χρόνια, η τεχνολογία γύρω από τους ηλεκτρονικούς υπολογιστές αναπτύσσεται με ταχείς ρυθμούς. Ο Παγκόσμιος Ιστός γνωρίζει ιδιαίτερη εξέλιξη, και πλέον αποτελεί αναπόσπαστο κομμάτι της κάθε επιχείρησης. Ολοένα και περισσότεροι χρήστες δικτυώνονται, περιηγούνται στο διαδίκτυο, και συναναστρέφονται με άλλους χρήστες, μέσω των ηλεκτρονικών κοινοτήτων κοινωνικής δικτύωσης.

Η παρούσα εργασία ασχολείται με την εκμετάλλευση της υπάρχουσας τεχνολογίας του διαδικτύου, με σκοπό την τουριστική ανάδειξη μιας γεωγραφικής περιοχής. Ο συνδυασμός του τουριστικού τομέα, με τον τομέα του διαδικτύου, και τα γεωγραφικά πληροφοριακά συστήματα παρέχουν νέες δυνατότητες στο τελικό χρήστη.

1.1 Σκοπός και στόχος της Διπλωματικής

Ο σκοπός της παρούσας εργασίας είναι να κατασκευαστεί ένα τουριστική πύλη portal όπου ο τελικός χρήστης να έχει τη δυνατότητα να αντλεί διάφορα δεδομένα (σημεία τουριστικού ενδιαφέροντος, φωτογραφίες, video, μετεωρολογικά δεδομένα κτλ.) για μια τουριστική περιοχή τα οποία θα του είναι χρήσιμα ώστε να μπορεί να οργανώσει καλύτερα την επίσκεψη του στη περιοχή αυτή.

Έχει δοθεί ιδιαίτερη βαρύτητα στην παρουσίαση των πληροφοριών αυτών να γίνεται σε "δυναμικό" χάρτη ώστε η να υπάρχει άμεση οπτική παρουσίαση τους και σύνδεσή τους με την ευρύτερη γεωγραφικές περιοχή. Παράλληλα δίνεται η δυνατότητα στο κάθε χρήστη της εφαρμογής να ενημερώσει το χάρτη με τα δικά του σημεία ενδιαφέροντος.

1.2 Δομή Διπλωματικής

Το υπόλοιπο της διπλωματική εργασίας είναι οργανωμένο ως ακολούθως:

Το πρώτο κεφάλαιο αποτελεί τη εισαγωγή της διπλωματικής εργασίας και περιγράφει συνοπτικά το σκοπό της εργασίας , και δίνεται η δομή του τόμου.

Στο δεύτερο Κεφάλαιο με τίτλο «WEB 2.0 Εφαρμογές και ιστορία» ο αναγνώστης έχει τη δυνατότητα να γνωρίσει το web 2.0. Γίνεται μια αναφορά στις αρχές που το διέπουν και παρουσιάζονται οι βασικές WEB 2.0 τεχνολογίες.

Στο τρίτο κεφάλαιο με τίτλο «Τεχνολογίες ανάπτυξης της εφαρμογής» εξετάζονται οι τεχνολογίες, καθώς και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη του portal.

Στο τέταρτο κεφάλαιο με τίτλο «Υλοποίηση παρουσίαση της εφαρμογής» γίνεται μια αναλυτική περιγραφή του portal και των δυνατοτήτων που προσφέρει στους χρήστες. Αναλύεται η δομή του portal, δίνεται η δομή της βάσης δεδομένων, καθώς και τα κυριότερα κομμάτια του κώδικα που αναπτύχθηκε. Επίσης παρουσιάζονται οι υπηρεσίες που προσφέρονται προς τον χρήστη και ο τρόπος που εμφανίζονται προς αυτόν.

Στο πέμπτο κεφάλαιο με τίτλο «Διαχείριση περιεχομένου του portal» γίνεται μια εκτενής αναφορά στις δυνατότητες διαχείρισης περιεχομένου που παρέχει η εφαρμογή μας. Παρουσιάζεται το Content Management System Joomla!, καθώς επίσης και η διαχειριστική σελίδα που αναπτύχθηκε, με σκοπό τη διαχείριση του περιεχομένου που προβάλλεται στον χάρτη.

Στο έκτο κεφάλαιο με τίτλο «Σενάριο χρήσης» παρουσιάζεται ένα υποθετικό σενάριο χρήσης του portal. Μέσω παραδειγμάτων, ο αναγνώστης αποκτά μια πλήρη εικόνα του portal και των προσφερόμενων υπηρεσιών.

Στο έβδομο κεφαλαίο με τίτλο «Επίλογος – Συμπεράσματα» παρουσιάζονται τα συμπεράσματα της διπλωματικής εργασίας καθώς και οι προτάσεις για μελλοντικές επεκτάσεις που μπορεί να υπάρξουν.

Τέλος, παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή και υλοποίηση της εργασίας καθώς και το μεγαλύτερο μέρος του κώδικα που αναπτύχθηκε.

Κεφάλαιο 2: WEB 2.0 Εφαρμογές και ιστορία.

2.1 Ιστορική αναδρομή του παγκοσμίου Ιστού.

Η τεχνολογία του παγκόσμιου ιστού δημιουργήθηκε το 1989 από τον Βρετανό *Sir Tim Berners Lee* ο οποίος εκείνη την εποχή εργαζόταν στον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (CERN) στην Γενεύη της Ελβετίας. Το όνομα που έδωσε στον ιστό είναι *World Wide Web* (σε πολλούς γνωστό και ως *www*). Ο παγκόσμιος ιστός είναι ένα σύστημα εγγράφων υπερκειμένου που περιέχονται στο διαδίκτυο. Με έναν web browser μπορεί κανείς να δει ιστοσελίδες που περιέχουν κείμενο, εικόνες, βίντεο και άλλα πολυμέσα, καθώς και να πλοηγηθεί ανάμεσά τους με την χρήση των υπερ-συνδέσεων.

Η προβολή μιας ιστοσελίδας στον παγκόσμιο ιστό ξεκινά πληκτρολογώντας την διεύθυνση (URL) της σελίδας σε έναν web browser, ή ακολουθώντας μια υπερ-σύνδεση από κάποια ιστοσελίδα. Ο web browser στην συνέχεια ξεκινά μια σειρά από μηνύματα επικοινωνίας στο παρασκήνιο έτσι ώστε να επικοινωνήσει με τον διακομιστή και να μας εμφανίσει την ιστοσελίδα. Αρχικά, ο browser στέλνει μια αίτηση στην παγκόσμια βάση δεδομένων που είναι γνωστή ως σύστημα ονομάτων τομέα (DNS), ώστε να γίνει μετάφραση του ονόματος της ιστοσελίδας που πρέπει να εμφανίσει, σε μια διεύθυνση IP. Γνωρίζοντας την IP διεύθυνση του διακομιστή ο web browser ζητάει την ιστοσελίδα στέλνοντας μια αίτηση HTTP προς στον διακομιστή. Στην περίπτωση μιας τυπικής ιστοσελίδας, ο κώδικας HTML που λαμβάνεται από τον web browser αναλύεται (parsed) αμέσως, και στην συνέχεια στέλνονται επιπλέον αιτήσεις για τις εικόνες και τα διάφορα άλλα στοιχεία που αποτελούν τμήμα τις ιστοσελίδας.

Ενώ λαμβάνονται αυτά τα αρχεία από τον διακομιστή, ο browser μπορεί να εμφανίζει σταδιακά την ιστοσελίδα, όπως αυτή περιγράφεται από τις γλώσσες HTML, CSS, κλπ. Οι περισσότερες ιστοσελίδες περιέχουν συνδέσμους προς άλλες σχετικές σελίδας και ίσως downloads, έγγραφα, ή διάφορους πόρους του διαδικτύου.

2.2 WEB 2.0 Μια νέα γενιά στον Παγκόσμιο Ιστό

Μια από τις τελευταίες εξελίξεις στο χώρο του διαδικτύου αποτελεί και η εξέλιξη του παραδοσιακού web σε αυτό που ονομάζετε Web 2.0.

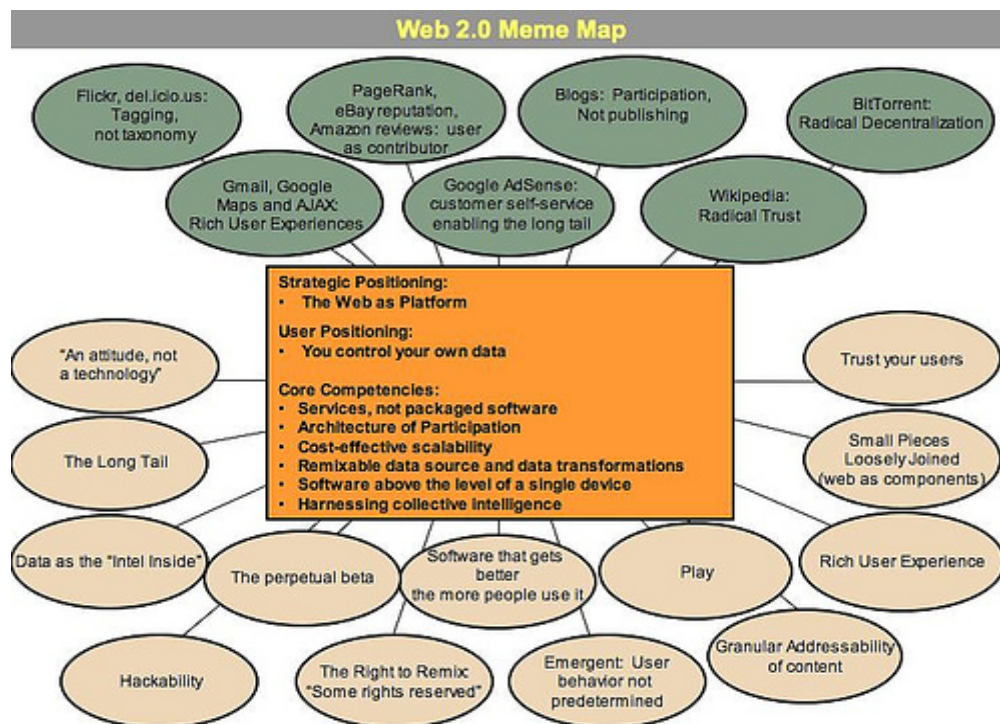
Η έννοια του Web2.0 χρησιμοποιήθηκε για πρώτη φορά σε ένα συνέδριο για την ανταλλαγή ιδεών μεταξύ του εκδοτικού οίκου O'Reilly και του Medialive International.

Ο όρος Web 2.0 χρησιμοποιείται για να περιγράψει τη νέα γενιά του Παγκόσμιου Ιστού η οποία στηρίζεται στην όλο και μεγαλύτερη δυνατότητα των χρηστών του διαδικτύου να μοιράζονται πληροφορίες και να συνεργάζονται online, και να αλληλεπιδρούν με τις ιστοσελίδες.

Παρόλο που ο όρος WEB 2.0 δίνει την αίσθηση ότι αποτελεί ένα καινούργιο πρωτόκολλό του WEB αυτό δεν συμβαίνει, αλλά στην ουσία πρόκειται για αλλαγές στον τρόπο που χρησιμοποιούνται ήδη υπάρχουσες τεχνολογίες, αλλά και στον τρόπο που οι σχεδιαστές πληροφοριακών συστημάτων, καθώς και οι χρήστες χρησιμοποιούν το διαδίκτυο.

Η κυρίαρχη τάση είναι να χρησιμοποιείται ως το μέσο (**πλατφόρμα**) πάνω στο οποίο θα τρέχουν οι εφαρμογές και υπηρεσίες, πολλές από τις οποίες μέχρι τώρα έτρεχαν τοπικά στους Η/Υ.

Το Web2.0 βρίσκει εφαρμογή σε βασικούς τομείς της ανθρώπινης δραστηριότητας όπως για παράδειγμα στην οικονομία, στο marketing, στην εκπαίδευση κτλ.



Εικόνα 1: Χάρτης του web 2.0

Βασικό στοιχείο όλων αυτών είναι ο κοινωνικός τους χαρακτήρας, ο οποίος ενισχύεται μέσα από την αλληλεπίδραση των χρηστών και την συμμετοχή τους στη διαμόρφωση της πληροφορίας.

Μιλάμε για μία δυναμική διαδικτυακή πλατφόρμα στην οποία μπορούν να αλληλεπιδρούν χρήστες χωρίς εξειδικευμένες γνώσεις σε θέματα

υπολογιστών και δικτύων. Οι κυριότερες εφαρμογές του Web 2.0 περιλαμβάνουν τις εξής :

- **Ta blogs**, είναι στην ουσία ιστοσελίδες που χρησιμοποιούνται για την καταχώρηση προσωπικών πληροφοριών και απόψεων των χρηστών. Οι καταχωρήσεις εμφανίζονται συνήθως με χρονολογική σειρά και με την τελευταία καταχώρηση να εμφανίζεται πρώτη.
- **Wikis**, είναι ιστοσελίδες των οποίων το περιεχόμενο μπορεί να τροποποιηθεί από τους χρήστες. Κάθε φορά που ένας χρήστης τροποποιεί το περιεχόμενο ενός wiki το περιεχόμενο (παλαιότερη έκδοση της ιστοσελίδας) εξακολουθεί να είναι διαθέσιμο και υπάρχει η δυνατότητα να επαναφερθεί. Τα wikis είναι διαδεδομένα στο χώρο των επιχειρήσεων και των οργανισμών.
- **Ιστοσελίδες**, στις οποίες ο χρήστης έχει τη δυνατότητα να "ανεβάσει" φωτογραφίες, video, (πχ. Youtube.com, myspace.com) τραγούδια ακόμα και να επικοινωνεί με φίλους του ή να κάνει νέες γνωριμίες (πχ facebook.com)
- **RSS (Real Simple Syndication) feeds**, δίνουν στους χρήστες τη δυνατότητα να λαμβάνουν πληροφορίες από ιστοσελίδες την ώρα που αυτές δημοσιεύονται χωρίς να απαιτείται να γίνει επίσκεψη στην ιστοσελίδα. Οι πληροφορίες αυτές εκτός από τον Η/Υ μπορεί να λαμβάνονται και από πλήθος άλλων συσκευών που έχουν πρόσβαση στο διαδίκτυο όπως κινητά τηλέφωνα , PDA's, κτλ.
- **Mashup**, εφαρμογές που είναι αποτέλεσμα της "μίξης" εφαρμογών κα πληροφοριών από διαφορετικές ιστοσελίδες.

2.3 Βασικές αρχές WEB 2.0

Παρακάτω παρουσιάζονται οι κυριότερες σχεδιαστικές αρχές και τεχνολογίες που χρησιμοποιούνται στο WEB 2.0 αυτές είναι:

- **Πλούσια και διαδραστικά interfaces χρηστών** βασισμένα σε τεχνολογία flash, Javascript κτλ.
- **Χρήση CSS (cascading Style Sheets)** για να διαχωρίζονται τα δεδομένα πληροφορίας από τα δεδομένα μορφοποίησης σε μια ιστοσελίδα. Με αυτό τον τρόπο ο χρήστης βλέπει δεδομένα ανάλογα με το CSS που ο ίδιος έχει.
- **Χρήση RSS (Real Simple Syndication) feeds** όπου παρέχεται η δυνατότητα οι χρήστες να λαμβάνουν νέες πληροφορίες από τις ιστοσελίδες χωρίς να είναι απαραίτητο να τις επισκεφθούν.
- **Αρχιτεκτονικές SOA (Service Oriented Architecture)** που επιτρέπουν το διαμοιρασμό και την επαναχρησιμοποίηση υπηρεσιών-εφαρμογών από διαφορετικά προγράμματα λογισμικού και **SaaS (Software as a Service)** όπου οι εφαρμογές είναι εγκατεστημένες σε κεντρικό server στο δίκτυο και οι χρήστες τις χρησιμοποιούν μέσω browser ανεξαρτήτως Η/Υ.
- **Χρήση ανοικτού λογισμικού** π.χ. **Linux** σαν λειτουργικό σύστημα, **Apache** σαν Web server, **MySQL** σαν βάση δεδομένων και **PHP, Pearl, Python** σαν server side γλώσσες.

2.4 MASHUP's

Ένα ιδιαίτερο τμήμα του WEB 2.0 αποτελούν οι mashup εφαρμογές.

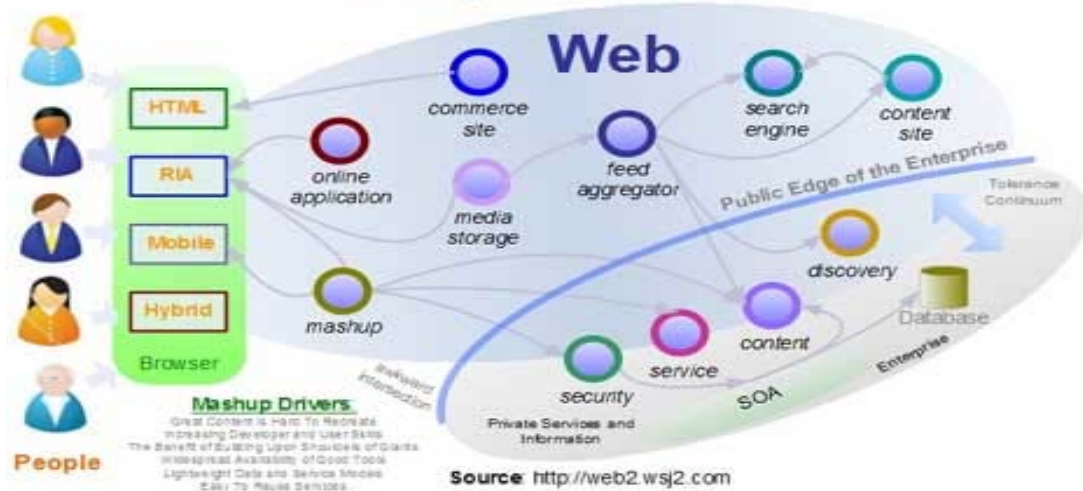
Ο όρος έχει δανεισθεί από τη μουσική βιομηχανία όπου mashup ονομάζεται η μίξη ενός τραγουδιού με την μουσική υπόκρουση ενός άλλου ώστε να συντεθεί ένα νέο.

2.4.1 Ορισμός

Ομοίως με την μουσική βιομηχανία, ο όρος MashUp αναφέρεται και σε διαδικτυακές εφαρμογές οι οποίες συγκεντρώνουν δεδομένα από διαφορετικές πηγές, τα επεξεργάζονται για να τα φέρουν σε μια νέα, ενιαία μορφή και τα παρουσιάζουν με νέους διαφορετικούς τρόπους. Ως πηγή μπορεί να χρησιμοποιηθεί υλικό οποιασδήποτε μορφής όπως άρθρα, πολυμέσα, χάρτες, εφαρμογές, ακόμα και στοιχεία που προέρχονται από βάσεις δεδομένων στο διαδίκτυο.

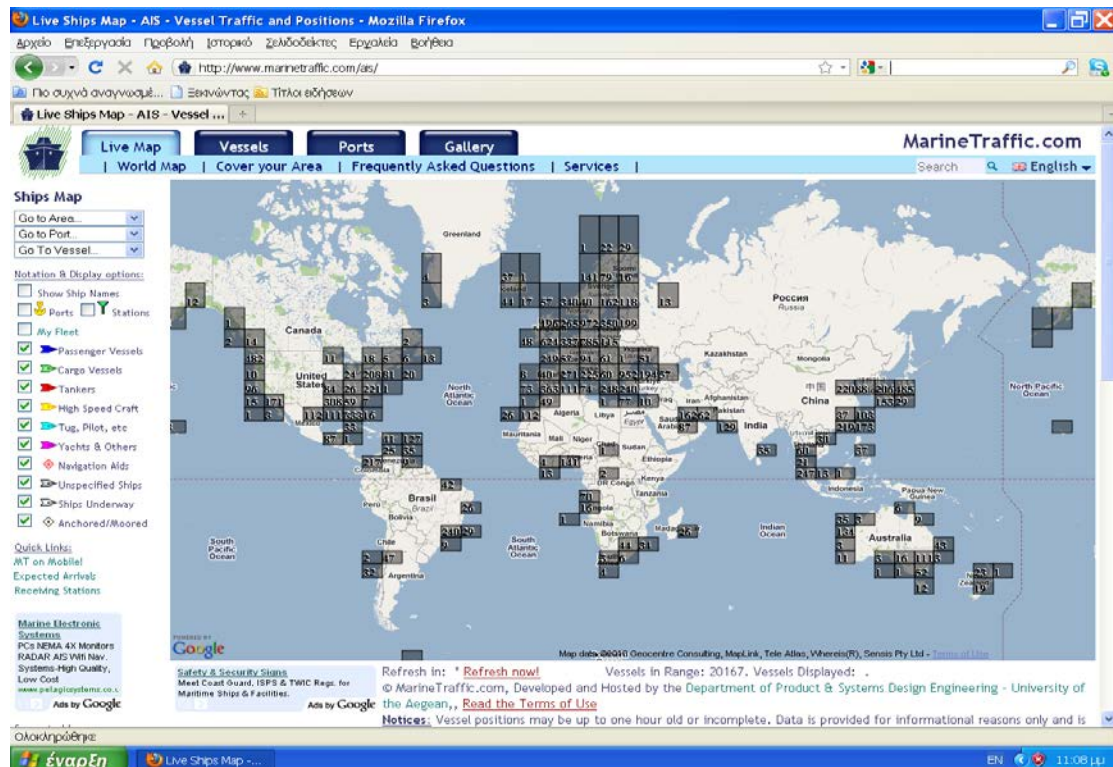
Ο στόχος ενός MashUp είναι ο συνδυασμός διάφορων αρχικών πηγών πληροφοριών ώστε να μπορέσει να προσφέρει μια νέας μορφής πληροφορία η οποία δεν ήταν διαθέσιμη από τις αρχικές ξεχωριστές πηγές δεδομένων, και να την παρουσιάσει στο χρήστη, σύμφωνα με τις εξατομικευμένες ανάγκες του.

The Mashup Ecosystem: Flourishing In An Increasingly Nurturing Environment



Εικόνα 2: Mashup Ecosystem

Το πιο χαρακτηριστικό και πιο διαδομένο παράδειγμα mashup είναι η χρήση των χαρτογραφικών δεδομένων από την υπηρεσία Google Maps για την προσθήκη πληροφορίας σχετικά με την θέση ακινήτων (πχ. Σπιτιών , καταστημάτων, κτλ) δημιουργώντας έτσι μια νέα και ευδιάκριτη WEB υπηρεσία που δεν παρέχετε ουσιαστικά από καμία άλλη πηγή.



Εικόνα 3: Στιγμιότυπο από την mashup εφαρμογή marinetraffic.

Τα δεδομένα που χρησιμοποιούνται ως πηγές συνήθως διατίθεται δωρεάν. Σε διαφορετική περίπτωση απαιτείται άδεια από τους δημιουργούς ή ιδιοκτήτες για την χρησιμοποίησή τους.

Τα mashup διαφοροποιούνται από την απλή ενσωμάτωση στοιχείων από ένα site σε ένα άλλο (πχ. Ενσωμάτωση ενός video), καθώς παρέχουν πρόσβαση και δυνατότητες επεξεργασίας στα δεδομένα, προσπαθώντας να δημιουργήσουν μια νέα υπηρεσία.

2.4.2 WEB APIS

Ο συνηθέστερος τρόπος για να επιτευχθεί ο συνδυασμός των δεδομένων είναι μέσω των λεγόμενων Application Programming Interfaces (API) που παρέχουν έτοιμο κώδικα για τη διασύνδεση των δικτυακών εφαρμογών.

Τα API's αποτελούνται από ένα σύνολο από ρουτίνες (routines), δομές δεδομένων (data structures), κλάσεις αντικειμένων (object classes), ή και πρωτόκολλα (protocols), που υποστηρίζουν την ανάπτυξη λογισμικού και υπηρεσιών βασισμένων σε μια πρωτογενή πηγή (υπολογιστικό σύστημα, βιβλιοθήκη, εφαρμογή).

Το API απλά ορίζει με ποιες εξωτερικές εντολές θα παρέχει την αμφίδρομη επικοινωνία με την υπηρεσία που θέλει να συνδεθεί προς αυτό χωρίς να αποκαλύπτει τον πηγαίο κώδικά του. Αυτές οι εντολές που καλούν το API, μπορεί να είναι γραμμένες σε συγκεκριμένη προγραμματιστική γλώσσα (Language-dependent API), ή σε οποιαδήποτε γλώσσα (Language-independent), αναλόγως του τύπου του API.



ProgrammableWeb.com 12/05/10

Εικόνα 4 : Κατανομή WEB APIS

Κεφάλαιο 3: Τεχνολογίες ανάπτυξης της εφαρμογής.

Οι web 2.0 εφαρμογές χρησιμοποιούν νέες τεχνολογίες και αρχές. Τεχνολογίες όπως AJAX , RSS feeds καθώς και γλώσσες προγραμματισμού όπως javascript, XML, XHTML είναι σύνηθες . Παρακάτω ακολουθεί μια συνοπτική παρουσίαση αυτών των τεχνολογιών με έμφαση σε αυτές που χρησιμοποιήθηκαν για την κατασκευή αυτής της εφαρμογής.

3.1 Βάσεις δεδομένων (MySql)

Μια βάση δεδομένων είναι μια δομημένη συλλογή από δεδομένα. Μπορεί να είναι οτιδήποτε από ένα απλό κατάλογο αγορών ως μια συλλογή φωτογραφιών ή τα απέραντα ποσά πληροφοριών σε ένα εταιρικό δίκτυο. Βάση δεδομένων είναι η οργάνωση και καταχώριση της πληροφορίας σε τρόπο ώστε να ενημερώνεται και να ανακαλείται με τον πλέον ευέλικτο τρόπο.

Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της, σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαιτέρου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως σύστημα διαχείρισης βάσης δεδομένων (DBMS).

Η χρήση του DBMS για τη διαχείριση των δεδομένων διακρίνεται από ένα πλήθος πλεονεκτημάτων :

- **Ανεξαρτησία των δεδομένων :** Οι εφαρμογές και ο κώδικας τους δεν πρέπει να εξαρτώνται από τις λεπτομέρειες που έχουν αν κάνουν με την αναπαράσταση και την αποθήκευση των δεδομένων.
- **Ταχεία πρόσβαση στα δεδομένα :** Το DBMS διαθέτει και εφαρμόζει αυτόματα μια ποικιλία τεχνικών ώστε να καταχωρεί και να ανακτά τα δεδομένα με γρήγορο και παραγωγικό τρόπο.
- **Ακεραιότητα των δεδομένων και ασφάλεια :** Το DBMS μπορεί να επιβάλει ελέγχους στη πρόσβαση των δεδομένων καθορίζοντας ποια τμήματα της βάσης θα είναι προσπελάσιμα από τις διάφορες κατηγορίες χρηστών και εφαρμογών.
- **Διαχείριση των δεδομένων :** Υπάρχει η δυνατότητα οργάνωσης των δεδομένων ώστε αυτά να καταχωρούνται μια μόνο φορά, όπως και ρύθμισης κατάλληλα της φυσικής καταχώρισης των δεδομένων με τρόπο ώστε να μεγιστοποιούνται οι επιδόσεις του συστήματος στη ανάκτηση της καταχωρημένης πληροφορίας.
- Ταυτόχρονη καταχώριση και επαναφορά από βλάβη : Το DBMS

χρονοπρογραμματίζει την ταυτόχρονη πρόσβαση στα δεδομένα, με τρόπο ώστε οι χρήστες να θεωρούν ότι μόνον ένας από αυτούς έχει πρόσβαση στη καταχωρημένη πληροφορία

Για την ανάπτυξη της εφαρμογής έχει χρησιμοποιηθεί η MySQL βάση δεδομένων. Αντίθετα με τις περισσότερες βάσεις δεδομένων η MySQL είναι ένα προϊόν ανοικτού κώδικα (ο πηγαίος κώδικας του είναι ελεύθερα διαθέσιμος για κατέβασμα στο καθένα). Οι προγραμματιστές μπορούν να αλλάξουν τον πηγαίο κώδικα έτσι ώστε να προσαρμόσουν τη MySQL στις ανάγκες τους. Ένα από τα καλά των προϊόντων ανοικτού κώδικα είναι ότι ένα ευρύ φάσμα προγραμματιστών και χρηστών συμβάλλουν την εμπειρία τους στο λογισμικό, κάνοντας το καλύτερο.

3.2 AJAX

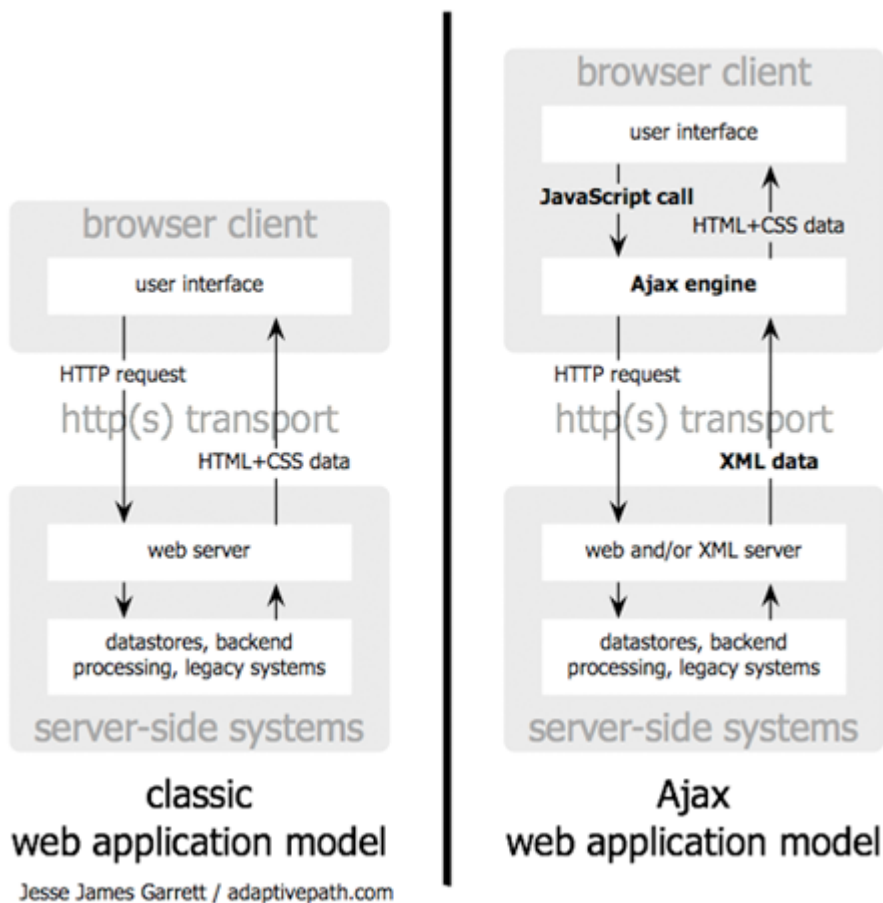
Ο όρος είναι συντομογραφία και δηλώνει την έννοια Asynchronous Javascript and XML. Όταν χρησιμοποιείται αυτή η τεχνική, δεν είναι απαραίτητη η φόρτωση ολόκληρης της ιστοσελίδας (reload), και η ανανέωση της (refresh) μετά από κάθε αλλαγή, γεγονός που δίνει το πλεονέκτημα της ταχύτητας που απαιτείται για τις διαδικτυακές εφαρμογές. Η AJAX δεν είναι μια τεχνολογία αλλά μια "μίξη" είδη υπαρχόντων τεχνολογιών και χρησιμοποιεί:

- Τις γλώσσες, επισήμανσης ιστοσελίδων XHTML και μορφοποίησης CSS, για την παρουσίαση του περιεχομένου.
- Το μοντέλο οντοτήτων DOM για τη δυναμική προβολή και την αλληλεπίδραση στις ιστοσελίδες.
- Τις γλώσσες, επισήμανσης XML και μετασχηματισμών XSLT, για την ανταλλαγή και τη διαχείριση δεδομένων αντίστοιχα.
- Το αντικείμενο XMLHttpRequest της JavaScript, για την ασύγχρονη ανάκτηση δεδομένων.
- Τη γλώσσα σεναρίων JavaScript για τη διασύνδεση και διαλειτουργικότητα των προηγούμενων τεχνολογιών.

Αν και η τυπική HTML καθώς και η Javascript χρησιμοποιείται για την δημιουργία ιστοσελίδων και τοποθεσιών, η τεχνική ajax καταλήγει σε μια web εφαρμογή, υπηρεσία δηλαδή βασισμένη στο web που εκτελεί κάποια εργασία για το χρήστη.

Ένα τυπικό αίτημα AJAX εργάζεται ως εξής:

1. Το script δημιουργεί ένα αντικείμενο XMLHttpRequest και το στέλνει στο διακομιστή.
2. Ο διακομιστής ανταποκρίνεται στο αίτημα και στέλνει τα περιεχόμενα ενός αρχείου ή την έξοδο ενός προγράμματος μέσω ενός XML αρχείου.
3. Όταν φτάσει η απάντηση τότε ξεκινά μια javascript συνάρτηση η οποία ενεργεί στα δεδομένα.
4. Το script εμφανίζει τα δεδομένα που λαμβάνονται από το διακομιστή χρησιμοποιώντας το DOM εξαλείφοντας με τον τρόπο αυτό την ανάγκη ανανέωσης της σελίδας.



Εικόνα 5: Τυπικό μοντέλο web εφαρμογών σε σύγκριση με το Ajax μοντέλο.

3.3 Γλώσσες σήμανσης ερωτημάτων.

Η γλώσσα σήμανσης είναι ένας συγκεκριμένος τρόπος γραφής κειμένου ο οποίος τηρεί κάποια συγκεκριμένα πρότυπα του οποίου το συντακτικό είναι τέτοιο ώστε να μπορούμε να το ξεχωρίσουμε από το απλό κείμενο. Από τις πιο γνωστές γλώσσες σήμανσης είναι οι ακόλουθες: HTML, XML και CSS.

3.3.1 HTML

Η γλώσσα HTML (**H**yper**T**ext **M**arkup **L**anguage) είναι η πρώτη και η πιο διαδεδομένη γλώσσα περιγραφής ιστοσελίδας την οποία αναγνωρίζουν πλέον όλοι οι browsers.

Χαρακτηριστικό της HTML είναι ετικέτες (tags), τις οποίες τις χρησιμοποιεί για να δώσει συγκεκριμένες οδηγίες στον browser για τη μορφοποίηση και το περιεχόμενο της ιστοσελίδας.

Επειδή η HTML είναι μια απλή γλώσσα σήμανσης κειμένου, δεν μπορεί να πάρει αποφάσεις ή να πραγματοποιήσει επαναλαμβανόμενες εργασίες.

3.3.2 XML

Η γλώσσα XML (**eX**tensible **M**arkup **L**anguage) περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Η βασικότερη χρήση της XML σήμερα είναι στο διαδίκτυο όπου αναπαριστά αυθαίρετες δομές δεδομένων, παρά το γεγονός ότι η σχεδίαση της εστιάζει στα κείμενα. Υπάρχει πλήρη υποστήριξη Unicode για όλες τις γλώσσες του κόσμου και υποστηρίζει την περιγραφή διεπαφής. Δηλαδή, με την χρήση της XML οι προγραμματιστές μπορούν να δημιουργήσουν μια διεπαφή περιγράφοντας την δομή της όπου στην συνέχεια μέσω αυτής της περιγραφής κατασκευάζεται η τελική διεπαφή από κάποιο λογισμικό που το υποστηρίζει.

3.3.3 CSS

Η CSS (Cascading Style Sheets) είναι μια γλώσσα περιγραφής στυλ που χρησιμοποιείται για να περιγράψει την εμφάνιση και την μορφοποίηση (presentation semantics) ενός εγγράφου που είναι γραμμένο σε μια γλώσσα σήμανσης. Η πιο κοινή εφαρμογή της γλώσσας αυτής είναι σε έγγραφα που είναι γραμμένα σε HTML και XHTML, όπου η CSS καθορίζει την εμφάνιση αυτών. Αν και ο συντάκτης του εγγράφου συνδέει τυπικά το έγγραφο με κάποιο CSS, ο αναγνώστης μπορεί να χρησιμοποιήσει κάποιο διαφορετικό CSS για να προβληθεί το έγγραφο αυτό, δίνοντας έτσι την δυνατότητα στους χρήστες να μπορούν να επιλέγουν ακόμη και τον τρόπο της δομής και της εμφάνισης ενός εγγράφου.

Η CSS είναι κυρίως σχεδιασμένη με σκοπό τον διαχωρισμό του περιεχομένου του εγγράφου (document content) από την δομή και την μορφοποίηση του εγγράφου (document presentation), συμπεριλαμβανομένων των στοιχείων διάταξης, χρωμάτων καθώς και γραμματοσειρών. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, παρέχοντας περισσότερη ευελιξία και έλεγχο στην συγγραφή της δομής και της μορφοποίησης του εγγράφου,

έτσι ώστε πολλές σελίδες να μπορούν να μοιράζονται την παραπάνω δομή και μορφοποίηση, και άρα μειώνοντας την πολυπλοκότητα σε περιεχόμενο περιγραφής δομής.

Επίσης, δίνεται η δυνατότητα σε ένα έγγραφο HTML ή XHTML, να παρουσιαστεί με τελείως διαφορετικό στυλ (και ως δομή και ως μορφοποίηση) ανάλογα το λειτουργικό και το λογισμικό του κάθε χρήστη.

3.4 Γλώσσες scripting

Μια γλώσσα scripting είναι μια γλώσσα προγραμματισμού η οποία επιτρέπει τον έλεγχο ενός ή και περισσότερων εφαρμογών λογισμικού (software applications). Τα κομμάτια κώδικα που γράφουμε σε αυτή την γλώσσα ονομάζονται *scripts* τα οποία ξεχωρίζουν από τον υπόλοιπο κώδικα μιας εφαρμογής αφού συνήθως είναι γραμμένα σε διαφορετική γλώσσα και συχνά δημιουργούνται και επεξεργάζονται από τον τελικό χρήστη (end-user).

Οι scripting γλώσσες είναι συνήθως ενσωματωμένες στις εφαρμογές στις οποίες ελέγχουν. Έχουμε δύο κύριες κατηγορίες των scripting γλωσσών

- *Client-side scripting languages* ονομάζονται οι γλώσσες αυτές που εκτελούνται στον πελάτη αντί για τον διακομιστή.
- *Server-side scripting languages* ονομάζονται οι γλώσσες αυτές που εκτελούνται στον διακομιστή και όχι στον πελάτη. Από τις πιο γνωστές είναι η Perl και η PHP, οι οποίες εκτελούνται στον διακομιστή όταν αυτός λάβει σχετική αίτηση.

Τα server-side scripts απαιτούν να εγκατασταθεί ένας διερμηνέας στον διακομιστή, και παράγουν το ίδιο αποτέλεσμα, ανεξάρτητα τον browser του πελάτη.

3.4.1 JAVA SCRIPT

Η JavaScript δημιουργήθηκε το 1995 από τον Brendan Eich, έναν μηχανικό της Netscape, και εκδόθηκε με τον Netscape 2 στις αρχές του 1996.

Αντίθετα από τις περισσότερες γλώσσες προγραμματισμού η γλώσσα JavaScript δεν έρχεται εφοδιασμένη με δυνατότητες εισόδου/εξόδου. Είναι σχεδιασμένη ως γλώσσα scripting σε ένα περιβάλλον που τη φιλοξενεί, και η ύπαρξη μηχανισμών επικοινωνίας με τον έξω κόσμο είναι ευθύνη αυτού του περιβάλλοντος. Η χρήση της είναι κυρίως για την δυναμική διαχείριση διαφόρων στοιχείων μιας ιστοσελίδας χωρίς να χρειάζεται αλληλεπίδραση client – server. Για παράδειγμα θα μπορούσε κάποιος να εμφανίσει ή να

κρύψει στοιχεία ή ομάδες στοιχείων ή και να αλλάξει το περιεχόμενο τους δυναμικά.

Η java script είναι μια διερμηνευμένη (interpreted) γλώσσα και δεν χρειάζεται να μεταγλωττιστεί για να εκτελεστεί, ο browser εκτελεί κάθε γραμμή του script καθώς το συναντά. Ο κώδικας της javascript εμφωλεύεται μέσα στην html με τη χρήση του tag<script> ή να γίνεται χρήση link που οδηγεί σε εξωτερικό αρχείο κώδικα.

Ως διερμηνευμένη γλώσσα έχει ορισμένα πλεονεκτήματα και μειονεκτήματα που περιγράφονται παρακάτω.

Πλεονεκτήματα

- Μείωση απαιτήσεων σε hardware του server εφόσον η γλώσσα εκτελείται στον browser του client.
- Η εκτέλεση γίνεται τοπικά, και δεν χρειάζεται αλληλεπίδραση με τον Server.
- Δημιουργία καλαισθητων και διαδραστικών ιστοσελίδων.
- Ανταπόκριση στις ενέργειες του χρήστη με μεγαλύτερη ταχύτητα.

Μειονεκτήματα

- Μεγαλύτερες απαιτήσεις σε Hardware στον client.
- Προβλήματα συμβατότητας μεταξύ των browser (διαφορετικός java interpreter) .*
- Ο κώδικας δεν μπορεί να έχει πρόσβαση στο σύστημα αρχείων του server
- Ο κώδικας είναι ορατός.

Το πρότυπο της γλώσσας δεν επιβάλλει κοινή ονοματολογία για την πρόσβαση στα στοιχεία μίας ιστοσελίδας. Αυτό έχει σαν αποτέλεσμα κάθε browser να παρέχει τις δικές του μεθόδους για τη πρόσβαση στα στοιχεία αυτά. Αυτό έχει σαν αποτέλεσμα να απαιτείται η ανάπτυξη ειδικού κώδικα για την υποστήριξη κάθε browser.

3.4.2 PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο. Είναι μια server-side (εκτελείτε στην πλευρά του διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων.

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.

Η php μπορεί να συνδεθεί ως άρθρωμα στους περισσότερους web servers και να συνδεθεί με αυτούς μέσω CGI (Common Gateway Interface). Υπάρχει επίσης η δυνατότητα να κληθεί και τοπικά από τη γραμμή εντολών ώστε να εκτελέσει ένα πρόγραμμα.

Η php όπως και οι περισσότερες server side γλώσσες έχει τη δυνατότητα να συνδεθεί εντός HTML κώδικα. Ο κώδικας της αναγνωρίζεται όταν βρίσκεται εντός των αναγνωριστικών `<?php` και `?>` το κείμενο που δεν συμπεριλαμβάνεται εντός αυτών των tags αντιγράφεται αυτούσιο στην έξοδο.

Κεφάλαιο 4 : Υλοποίηση παρουσίαση της εφαρμογής

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση της εφαρμογής καθώς και του κώδικα που έχει χρησιμοποιηθεί. Για την ανάπτυξη του portal έχει γίνει χρήση κυρίως ελεύθερου λογισμικού και στηρίζεται στη χρήση της Mysql ως βάσης δεδομένων, του CMS Joomla! ως σύστημα διαχείρισης περιεχομένου, καθώς και της php ως scripting γλώσσα.

4.1 Παρουσίαση portal

Η οργάνωση της δομής του Portal είναι η ακόλουθη.

Αρχική | Είναι η αρχική σελίδα στην οποία παρουσιάζονται γενικές πληροφορίες

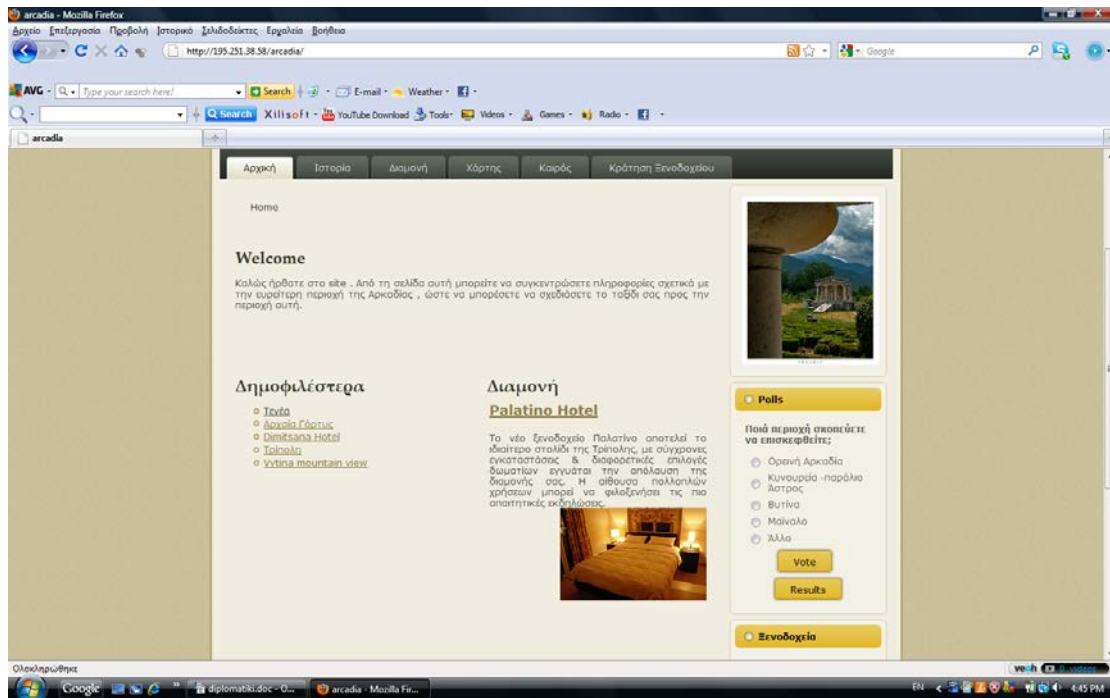
Ιστορία | Στη σελίδα αυτή παρουσιάζονται πληροφορίες για την ιστορία της τοποθεσίας που θέλουμε να προβάλουμε. Η οργάνωση του περιεχομένου είναι με τη μορφή Blog Layout.

Διαμονή | Στη σελίδα αυτή παρουσιάζονται πληροφορίες σχετικά με τις δυνατότητες διαμονής (Ξενοδοχεία , ενοικιαζόμενα δωμάτια, κτλ.) που υπάρχουν στην περιοχή που προβάλλεται μέσω του portal .

Χάρτης | Στη σελίδα αυτή παρουσιάζονται μέσω δυναμικού χάρτη πληροφορίες για την περιοχή ενδιαφέροντος του portal. Οι πληροφορίες είναι οργανωμένες κατά κατηγορίες σημείων ενδιαφέροντος (πχ, Αρχαιολογικές περιοχές, Μουσεία, Ξενοδοχεία κτλ.) και παρέχεται η δυνατότητα στους εγγεγραμμένους χρήστες να προσθέσουν τα δικά τους σημεία ενδιαφέροντος.

Καιρός | Μέσω της σελίδας αυτής, παρουσιάζονται πληροφορίες για τις καιρικές συνθήκες που επικρατούν στην περιοχή ενδιαφέροντος του Portal.

Κράτηση Ξενοδοχείου | Μέσω της σελίδας αυτής παρέχεται η δυνατότητα στους χρήστες του portal να βρίσκουν ξενοδοχεία καθώς και να πραγματοποιούν κρατήσεις δωματίων στα ξενοδοχεία αυτά.



Εικόνα 6: Η αρχική σελίδα του portal.

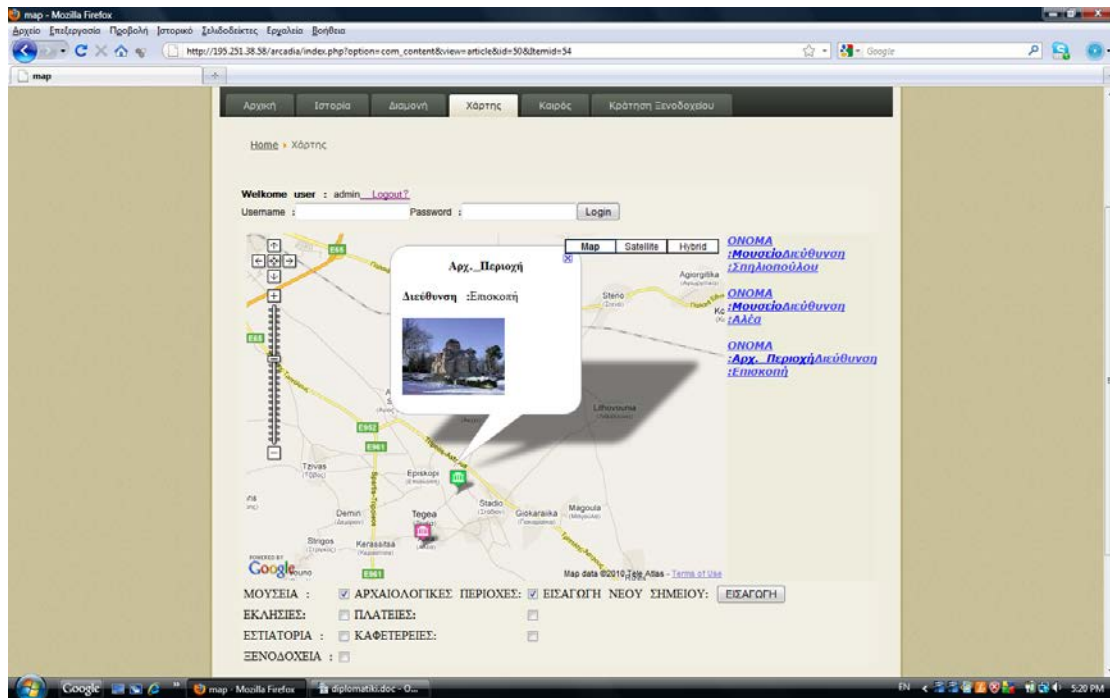
4.1.1 Σελίδα του δυναμικού χάρτη.

Στη σελίδα αυτή παρουσιάζονται και απεικονίζονται σε χάρτη διάφορα σημεία ενδιαφέροντος, τα οποία είναι οργανωμένα σε διάφορες κατηγορίες.

Η κατασκευή του χάρτη βασίζεται στο χαρτογραφικό υπόβαθρο που προσφέρει η υπηρεσία Googlemap.

Η Google, προσφέρει μια υπηρεσία απεικόνισης παγκοσμίων χαρτών στο web, που προσφέρεται δωρεάν. Οι δυνατότητες απεικόνισης - εκτός της κλασικής προβολής χάρτη - που μας προσφέρει η υπηρεσία αυτή είναι:

1. **Satellite** : δορυφορικές φωτογραφίες ή αεροφωτογραφίες υψηλής ανάλυσης η οποίες διαφέρουν από περιοχή σε περιοχή.
2. **Hybird** : Πρόκειται για μια υβριδική μορφή απεικόνισης όπου στο satellite υπόβαθρο γίνεται απεικόνιση και χαρτογραφικών πληροφοριών (πχ. Ονομασίες Πόλεων, Οδών κτλ.)
3. **Terrain** : Πρόκειται για μια μορφή απεικόνισης όπου αναπαρίσταται το ανάγλυφο της γης.



Εικόνα 7: Η σελίδα του χάρτη.

Η Google παρέχει επίσης προς τους προγραμματιστές ένα API (Application Programming Interface) ώστε να μπορούν να ενσωματώσουν και να παρουσιάσουν χάρτες στις δικές τους ιστοσελίδες.

Πρόκειται για ένα Javascript API (πράγμα που επιτρέπει την ενσωμάτωσή του σε απλές ιστοσελίδες δίχως ιδιαίτερες απαιτήσεις από τη μεριά του server) που αποτελείται από ένα σύνολο Κλάσεων (Classes) με Ιδιότητες (Properties), Συμβάντα (Events), Μεθόδους (Methods) δομημένα με τρόπο που να κάνουν εφικτή τη πρόσβαση στα δεδομένα, και να επιτρέπουν την ανάκτηση αυτών, καθώς και τον εμπλουτισμό τους πριν από την απεικόνιση.

Για την χρήση των χαρτών απαιτείται η χρήση ενός κλειδιού (Api – key) τον οποίο διατίθεται δωρεάν με μοναδική προϋπόθεση την ύπαρξη ενός λογαριασμού στη Google. Το κάθε κλειδί που διατίθεται είναι έγκυρο μόνο για ένα directory ή domain.

Τα σημεία ενδιαφέροντος που θέλουμε να προβάλουμε είναι οργανωμένα σε βάση δεδομένων, ώστε να είναι εφικτή η ξεχωριστή αναζήτηση τους από κάθε χρήστη. Σαν βάση δεδομένων χρησιμοποιείται η Mysql.

Η κατασκευή του πίνακα της βάσης παρουσιάζεται πιο κάτω και ιδιαίτερη προσοχή πρέπει να δοθεί στα πεδία 'lat' και 'lng' που αφορούν τις γεωγραφικές συντεταγμένες του σημείου ενδιαφέροντος. Οι υπόλοιπες πληροφορίες αφορούν στοιχεία του σημείου ενδιαφέροντος που θέλουμε να προβληθούν στο χάρτη με τη μορφή info window και περιλαμβάνουν πληροφορίες όπως όνομα, τηλέφωνο, φωτογραφία κτλ.


```

CREATE TABLE 'markers' (
  'id' int(6) NOT NULL auto_increment,
  'category' varchar(40) collate utf8_unicode_ci NOT NULL,
  'name' varchar(60) collate utf8_unicode_ci default NULL,
  'address' varchar(60) collate utf8_unicode_ci default NULL,
  'telephone' varchar(60) collate utf8_unicode_ci default NULL,
  'email' varchar(60) collate utf8_unicode_ci default NULL,
  'photo' varchar(100) collate utf8_unicode_ci default NULL,
  'description' varchar(60) collate utf8_unicode_ci default NULL,
  'user' varchar(60) collate utf8_unicode_ci default NULL,
  'lat' varchar(60) collate utf8_unicode_ci default NULL,
  'lng' varchar(60) collate utf8_unicode_ci default NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=93 DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci

```

Τα δεδομένα αυτά προβάλλονται στο χάρτη καλώντας ένα xml αρχείο της παρακάτω μορφής:

```

<markers>
<marker          category="restaurant"          name="Grand_Sale"
address="Πλ._Αρεος"  user="admin"  photo="grandchalet1-150.jpg"
telephone="2710234661"  email=""  description=""  lat="37.51366"
lng="22.37627"/>
<marker          category="restaurant"          name="Ταβέρνα"
address="Καλαβρύτων"  user="admin"  photo=""  telephone="testςεςεςς"
email="555"  description=""  lat="37.51539"  lng="22.37492"/>
<marker          category="eklisies"          name="Αγ_Γεώργιος"
address="Άλσος_Αγ_Γεωργίου"  user="admin"  photo=""  telephone=""
email=""  description=""  lat="37.53073"  lng="22.37533"/>
</markers>

```

Για την κατασκευή αυτού του αρχείου γίνεται η χρήση κώδικα της php, που περιέχεται στο αρχείο `phpsqlajax_genxml31.php` που εμφανίζεται παρακάτω:

```

<?php
session_start();
if(!session_is_registered(myusername)){
header("location: index.php");
}

require("phpsqlajax_dbinfo.php");

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

$connection=mysql_connect (localhost, $username, $password);
if (!$connection) {
die('Not connected : ' . mysql_error());
}

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

$query = "SELECT * FROM markers WHERE
user='$_SESSION[myusername]' or user='admin'";
$result = mysql_query($query);
if (!$result) {
die('Invalid query: ' . mysql_error());
}
header("Content-type: text/xml");
// Start XML file
echo '<markers>';

while ($row = @mysql_fetch_assoc($result)){

echo '<marker ' ;
echo 'category="' . parseToXML ($row['category']). '" ' ;
echo 'name="' . parseToXML($row['name']) . '" ' ;
echo 'address="' . parseToXML($row['address']) . '" ' ;
echo 'user="' . parseToXML($row['user']) . '" ' ;
echo 'photo="' . parseToXML($row['photo']) . '" ' ;
echo 'telephone="' . parseToXML($row['telephone']) . '" ' ;
echo 'email="' . parseToXML($row['email']) . '" ' ;
}
}

```

```

echo 'description=' . parseToXML($row['description']) . ' ' ;
echo 'lat=' . $row['lat'] . ' ' ;
echo 'lng=' . $row['lng'] . ' ' ;
echo '>';
}
// End XML file
echo '</markers>';

?>

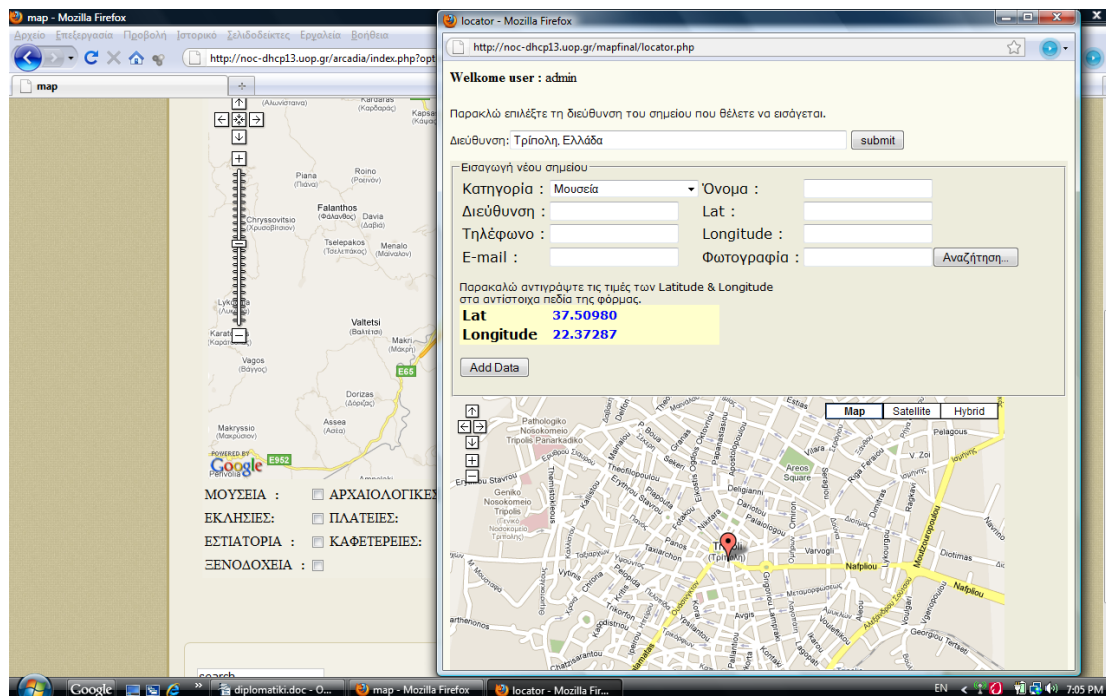
```

4.1.2 Εισαγωγή δυναμικού περιεχομένου στο χάρτη.

Η εισαγωγή περιεχομένου στο χάρτη επιτρέπεται για τους εγγεγραμμένους χρήστες της εφαρμογής. Η εγγραφή νέου χρήστη γίνεται ακολουθώντας το κατάλληλο υπέρ-σύνδεσμο που υπάρχει στην ιστοσελίδα του χάρτη, και συμπληρώνοντας τα στοιχεία του χρήστη στα κατάλληλα στοιχεία της φόρμας.

Για την προσθήκη ενός νέου σημείου ενδιαφέροντος, ο εγγεγραμμένος χρήστης πρέπει να επιλέξει το "button" με την ένδειξη εισαγωγή.

Η επιλογή αυτή θα εμφανίσει μια αναδυόμενη σελίδα με το όνομα locator.



Εικόνα 8: Η σελίδα εισαγωγής νέου σημείου ενδιαφέροντος.

Η προσθήκη ενός νέου σημείου ενδιαφέροντος, γίνεται με εύκολο τρόπο. Αρχικά πρέπει να βρεθούν οι γεωγραφικές συντεταγμένες του σημείου, που μπορεί να γίνει με δύο τρόπους, είτε αναζητώντας το σημείο ενδιαφέροντος με βάση την διεύθυνση του, ή μετακινώντας το χάρτη μέχρι το σημείο ενδιαφέροντος να συμπέσει με τον marker που υπάρχει στο κέντρο του χάρτη. Και στις δύο περιπτώσεις οι συντεταγμένες lat και Longitude εμφανίζονται στο κατάλληλο πλαίσιο. Έπειτα αφού επιλέξουμε μία από τις προκαθορισμένες κατηγορίες σημείων ενδιαφέροντος, αντιγράφουμε τις τιμές lat και Longitude στα κατάλληλα πεδία, και συμπληρώνουμε τα υπόλοιπα πεδία με περιγραφικές πληροφορίες για το σημείο, όπως όνομα, διεύθυνση, τηλέφωνο, e-mail. Οι προκαθορισμένες κατηγορίες σημείων ενδιαφέροντος είναι Μουσεία, Αρχαιολογικές περιοχές, Εκκλησίες, Πλατείες, Εστιατόρια, Καφετέριες, Ξενοδοχεία. Από το πεδίο φωτογραφία, είναι δυνατόν να επισυνάψουμε φωτογραφία του σημείου, η οποία θα εμφανίζεται στο info window, κατά την προβολή του σημείου.

Τέλος η καταχώρηση ολοκληρώνεται κάνοντας κλικ στο button "Add Data".

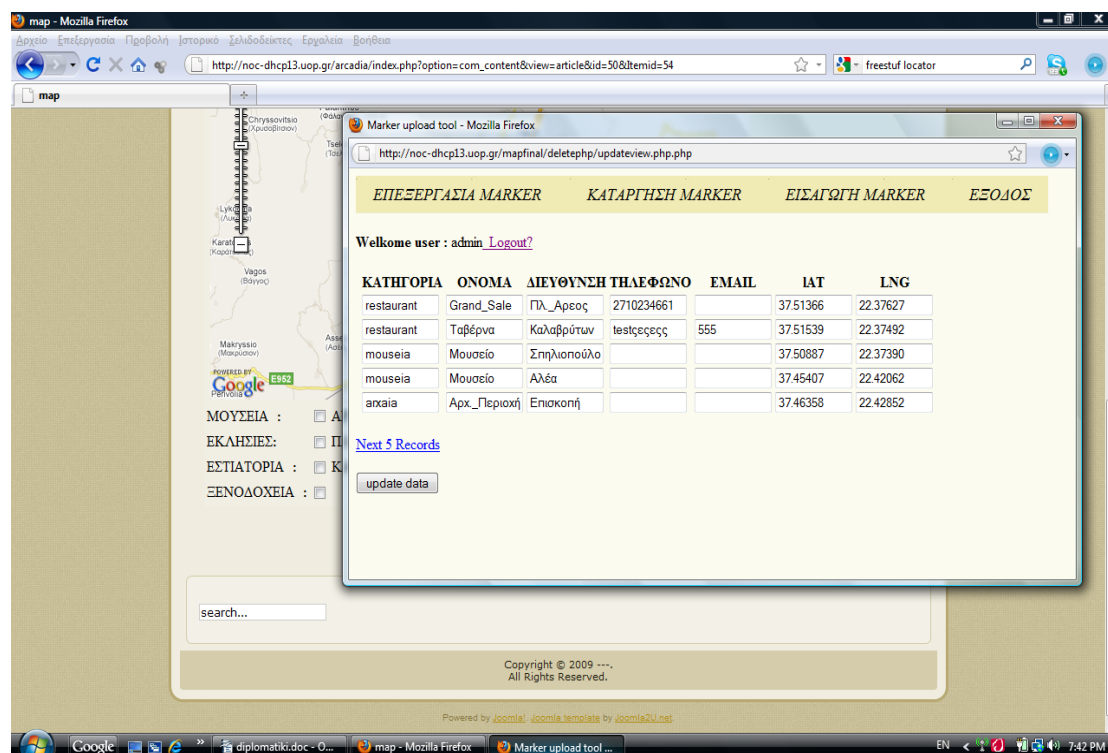
The screenshot shows a web browser window with a map application. A modal window titled "Marker upload tool" is open, displaying a table of existing points of interest. The table has columns for category, name, address, phone, email, latitude, and longitude. The data in the table is as follows:

KATHΓΟΡΙΑ	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ	ΤΗΛΕΦΩΝΟ	EMAIL	IAT	LNG
<input type="checkbox"/> restaurant	Grand Sale	Πλ_Αρεος	2710234661		37.51366	22.37627
<input type="checkbox"/> restaurant	Ταβέρνα	Καλαβρύτων	testcececs	555	37.51539	22.37492
<input type="checkbox"/> museia	Μουσείο	Σπηλιοπούλου			37.50887	22.37390
<input type="checkbox"/> museia	Μουσείο	Αλέα			37.45407	22.42062
<input type="checkbox"/> arxaia	Αρχ_Περιοχή	Επισκοπή			37.46358	22.42852

Below the table, there is a "Delete !!" button and a footer that says "created by Giorgos 20/8/2010".

Εικόνα 9: Διαγραφή σημείων ενδιαφέροντος.

Μετά την επιτυχή εισαγωγή σημείου οδηγούμαστε σε ιστοσελίδα όπου ο χρήστης ακολουθώντας κατάλληλους υπέρ-συνδέσμων έχει τη δυνατότητα



Εικόνα 10: Ενημέρωση σημείων ενδιαφέροντος.

διαγραφής, ή ενημέρωσης των σημείων που έχει εισάγει.

4.2 Σύστημα κράτησης δωματίων

Από τη σελίδα "Κράτηση Ξενοδοχείου" ο χρήστης της εφαρμογής έχει τη δυνατότητα να αναζητήσει και να πραγματοποιήσει κράτηση σε δωμάτια από ένα πλήθος από ξενοδοχεία της περιοχής ενδιαφέροντος μας.

4.2.1 Σχεδιασμός της Βάσης δεδομένων

Τα δεδομένα των ξενοδοχείων οργανώνονται σε μια mysql βάση δεδομένων που αποτελείται από τρεις πίνακες. Ο Πίνακας hotel αποθηκεύει δεδομένα σχετικά με το ξενοδοχείο (πόλη, όνομα, τηλέφωνο).

```
mysql> describe hotel;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| hotelno    | int(4)        | NO   | PRI | NULL    | auto_increment |
| hotelname  | varchar(40)   | NO   |     |         |                |
| city       | varchar(40)   | NO   |     |         |                |
| phone      | varchar(15)   | NO   |     |         |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Ο πίνακας room περιέχει πληροφορίες για τα δωμάτια των ξενοδοχείων τα πιθανά είδη δωματίων πχ. Μονόκλινο, Δίκλινο, Σουίτα κτλ., καθώς και για την τιμή τους.

```
mysql> describe room;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| roomid     | int(4)        | NO   | PRI | NULL    | auto_increment |
| roomno     | varchar(10)   | NO   |     |         |                |
| type       | varchar(40)   | NO   |     |         |                |
| price      | decimal(8,0)  | NO   |     |         |                |
| hotelno    | int(4)        | NO   |     |         |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Τέλος στο πίνακα booking περιέχονται πληροφορίες για τα δωμάτια που βρίσκονται σε κράτηση. Κύρια πεδία σε αυτό το πίνακα αποτελούν τα datefrom, και dateto, όπου στην ουσία καταγράφονται οι ημερομηνίες Άφιξης, και Αποχώρησης, αντίστοιχα. Τα υπόλοιπα πεδία καταγράφουν πληροφορίες σχετικά με τα στοιχεία του χρήστη που πραγματοποιεί τη κράτηση, όπως όνομα, επώνυμο, τηλέφωνο κτλ.

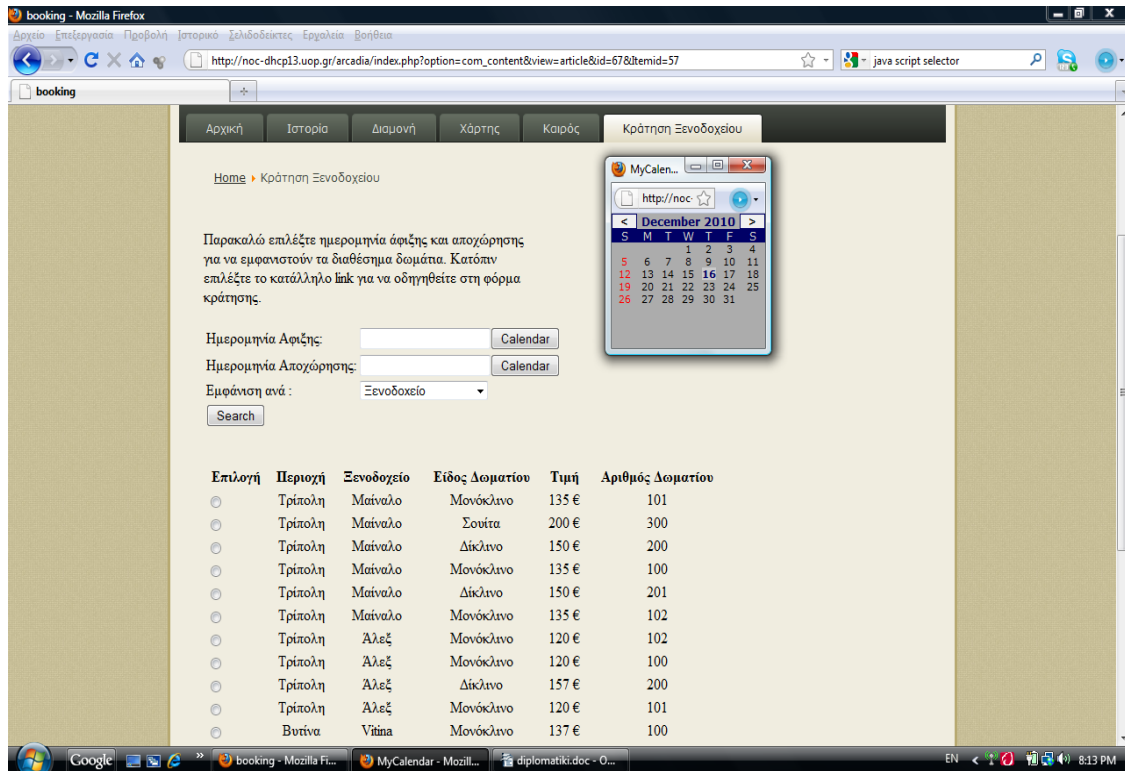
```
mysql> describe booking;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| bookid     | int(4)    | NO   | PRI | NULL    | auto_increment |
| datefrom   | date      | NO   |     |         |              |
| dateto     | date      | NO   |     |         |              |
| roomid     | int(4)    | NO   |     |         |              |
| name       | varchar(40) | NO   |     |         |              |
| lastname   | varchar(60) | NO   |     |         |              |
| telephone  | varchar(15) | NO   |     |         |              |
| address    | varchar(60) | NO   |     |         |              |
| email      | varchar(50) | NO   |     |         |              |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

4.2.2 Σελίδα κράτησης δωματίων.

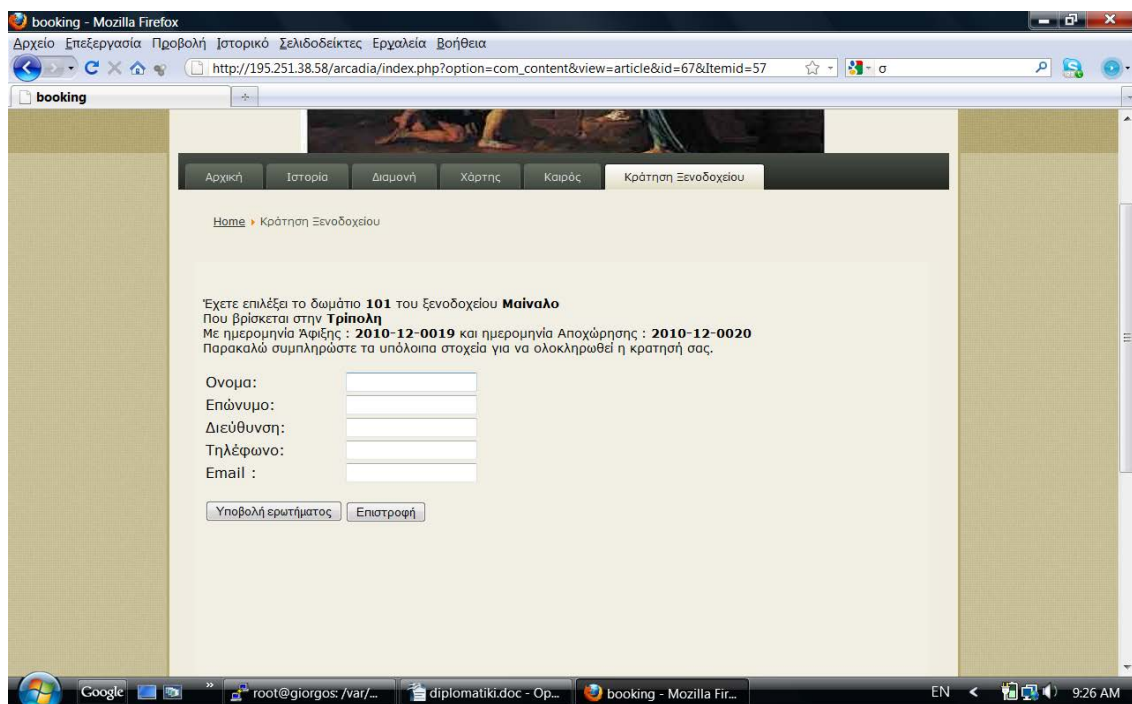
Στη σελίδα με τίτλο “Κράτηση Ξενοδοχείου” προσφέρει ένα μηχανισμό εύρεσης ελεύθερων δωματίων για την χρονική περίοδο που επιθυμεί ο χρήστης. Η αναζήτηση μπορεί να γίνει εισάγοντας τις κατάλληλες τιμές στα πεδία “Ημερομηνία Άφιξης” και “Ημερομηνία Αποχώρησης”. Η εισαγωγή μπορεί να διευκολυνθεί επιλέγοντας τα “button” με τίτλο “calendar”, όπου μας εμφανίζει ένα αναδυόμενο παράθυρο που περιέχει ένα ημερολόγιο. Αφού εισαχθούν οι ημερομηνίες που μας ενδιαφέρουν ο ρηρ κώδικας της σελίδας εκτελεί το παρακάτω sql ερώτημα :

```
$sql = mysql_query("select hotel.hotelname, hotel.city, room.roomid,
room.roomno, room.price, room.type from hotel, room where
room.roomid not in (select roomid from booking, hotel where (
(datefrom <= '$Sdate' AND dateto >= '$edate') or (datefrom
<= '$edate' AND dateto >= '$edate') or (datefrom <= '$Sdate' AND
dateto >= '$Sdate') or (datefrom >= '$Sdate' AND dateto <= '$edate'))
and room.hotelno=hotel.hotelno order by $cat") or die(mysql_error());
```

Τα αποτελέσματα μπορούν να εμφανιστούν ανά, Ξενοδοχείο , τύπο δωματίου, ή τιμή δωματίου. Στη συνέχεια αφού επιλέξουμε το δωμάτιο που μας ενδιαφέρει, και επιλέγοντας το “button” με τίτλο booking οδηγούμαστε στη σελίδα κράτησης δωματίου, όπου καλούμαστε να εισάγουμε τα στοιχεία που είναι απαραίτητα για την ολοκλήρωση της κράτησης.



Εικόνα 11: Σελίδα αναζήτησης ελεύθερου δωματίου



Εικόνα 12: Η σελίδα κράτησης δωματίου

Κεφάλαιο 5 : Διαχείριση περιεχομένου του portal

Η ανάγκη για συνεχή και εύκολη online τροποποίηση του περιεχομένου του portal μας επιβάλλει τη χρήση ενός συστήματος διαχείρισης περιεχομένου CMS (Content Management System). Η λειτουργικότητα ενός συστήματος διαχείρισης περιεχομένου, μπορεί να αναλυθεί στις εξής βασικές κατηγορίες:

- δημιουργία περιεχομένου
- διαχείριση περιεχομένου
- δημοσίευση
- παρουσίαση

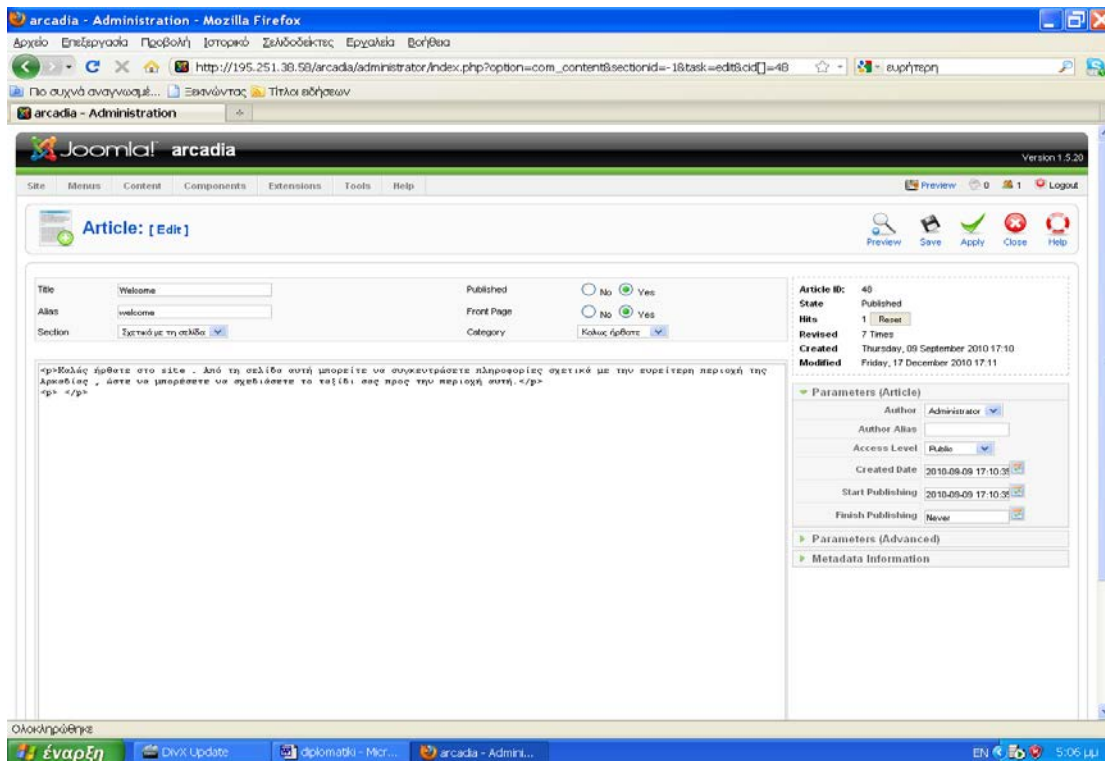
Υπάρχουν αρκετά συστήματα διαχείρισης περιεχομένου, με τα πιο δημοφιλή να είναι το Joomla!, Drupal , Wordpress. Για τις ανάγκες ανάπτυξης του portal έχει επιλεγεί το Joomla! το οποίο είναι ανοικτού κώδικα, και διαθέτει ένα αρκετά φιλικό περιβάλλον για το χρήστη.

Για τις ανάγκες διαχείρισης του περιεχομένου του χάρτη, καθώς και του συστήματος κράτησης δωματίων, έχει γίνει ανάπτυξη με χρήση κώδικα php ξεχωριστού εργαλείου.

5.1 CMS JOOMLA!

Το CMS Joomla! είναι ένα ιδιαίτερα δημοφιλές ελεύθερο λογισμικό (διατίθεται υπό την GNU GPL) Διαχείρισης Περιεχομένου. Γραμμένο σε **php**, χρησιμοποιεί την **MySQL** ως βάση δεδομένων, έχει φτάσει στην έκδοση 1.5.x και αποτελεί την εξέλιξη ενός άλλου CMS, του Mambo. Περιλαμβάνει αρκετά χαρακτηριστικά όπως caching των σελίδων, διαχείριση νέων, RSS Feeds, blogs, αναζητήσεις, υποστήριξη πολυγλωσσικών ιστοσελίδων κτλ. Η υποστήριξη του γίνεται από μια μεγάλη κοινότητα χρηστών, και η αρθρωτή του ανάπτυξη, επιτρέπει την εύκολη ενσωμάτωση διαφόρων επεκτάσεων, οι οποίες του δίνουν νέες δυνατότητες.

Από την πλευρά του χρήστη για τη δημιουργία ή τροποποίηση του περιεχομένου απαιτείται μόνο η γνώση χρήσης κειμενογράφου.



Εικόνα 13: Εισαγωγή τροποποίηση άρθρου από Joomla!.

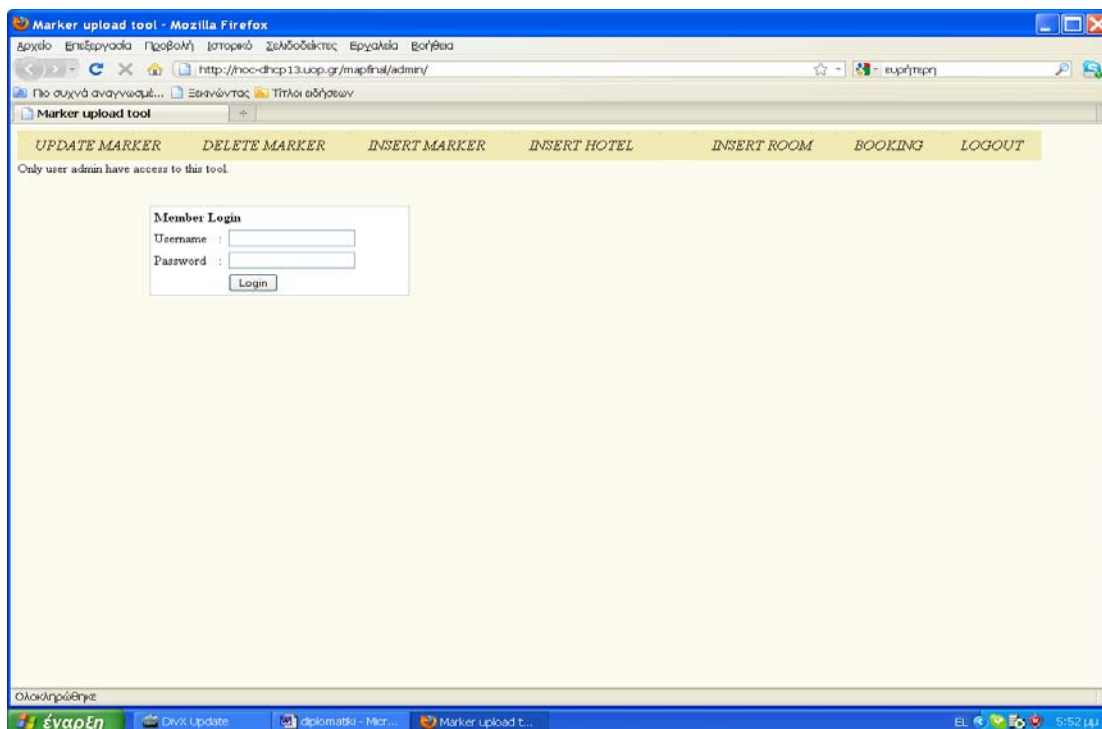
Από την πλευρά του διαχειριστή για τη συντήρηση του συστήματος, χρήσιμη είναι η διαρκής ενημέρωση για νέες εκδόσεις. Στην περίπτωση που υπάρχει κάποιο κενό ασφαλείας και η κοινότητα διαθέτει κάποια ενημέρωση (patch), σκόπιμο θα ήταν να αναβαθμιστεί το λογισμικό στη νέα έκδοση. Η αναβαθμίσεις και η εγκατάσταση νέων επεκτάσεων γίνεται με τη βοήθεια του Extension Manager.

Ο διαχειριστής του συστήματος θα πρέπει να φροντίζει, και για τη συχνή λήψη αντιγράφων ασφαλείας (Backups). Η λήψη backup γίνεται με δύο τρόπους. Ο πρώτος τρόπος υλοποιείται με την αντιγραφή των αρχείων του Joomla! σε κάποιον άλλο φάκελο ή σε κάποια εξωτερική συσκευή αποθήκευσης, και την λήψη ενός στιγμιότυπου από τη βάση δεδομένων του Joomla!. Ένας δεύτερος τρόπος είναι με τη χρήση ενός από τα backup extensions που βρίσκονται στην ιστοσελίδα του Joomla!.

5.2 Διαχειριστική σελίδα

Με την συγκεκριμένη σελίδα ο διαχειριστής του portal έχει τη δυνατότητα να διαχειριστή το περιεχόμενο του χάρτη, να προσθέσει νέα δωμάτια ξενοδοχείων προς κράτηση καθώς και να ενημερωθεί για τις ημερομηνίες που βρίσκονται αυτά σε κράτηση.

Η σελίδα βρίσκεται στη διεύθυνση: <http://nocdhcp13.uop.gr/mapfinal/admin/>



Εικόνα 14: Η φόρμα εισαγωγής της διαχειριστικής σελίδας.

Αρχικά ο χρήστης καλείται να συμπληρώσει τα στοιχεία του όπως φαίνεται και στην εικόνα . Μετά την επιτυχή εισαγωγή ο διαχειριστής μπορεί να επιλέξει το είδος της εργασίας που θέλει να κάνει επιλέγοντας τους κατάλληλους υπέρ-συνδέσμους. Η δομή της σελίδας διαχείρισης είναι η ακόλουθη:

UPDATE MARKER | Μέσω της σελίδας αυτής, μπορούν να τροποποιηθούν τα στοιχεία των σημείων ενδιαφέροντος του χάρτη.

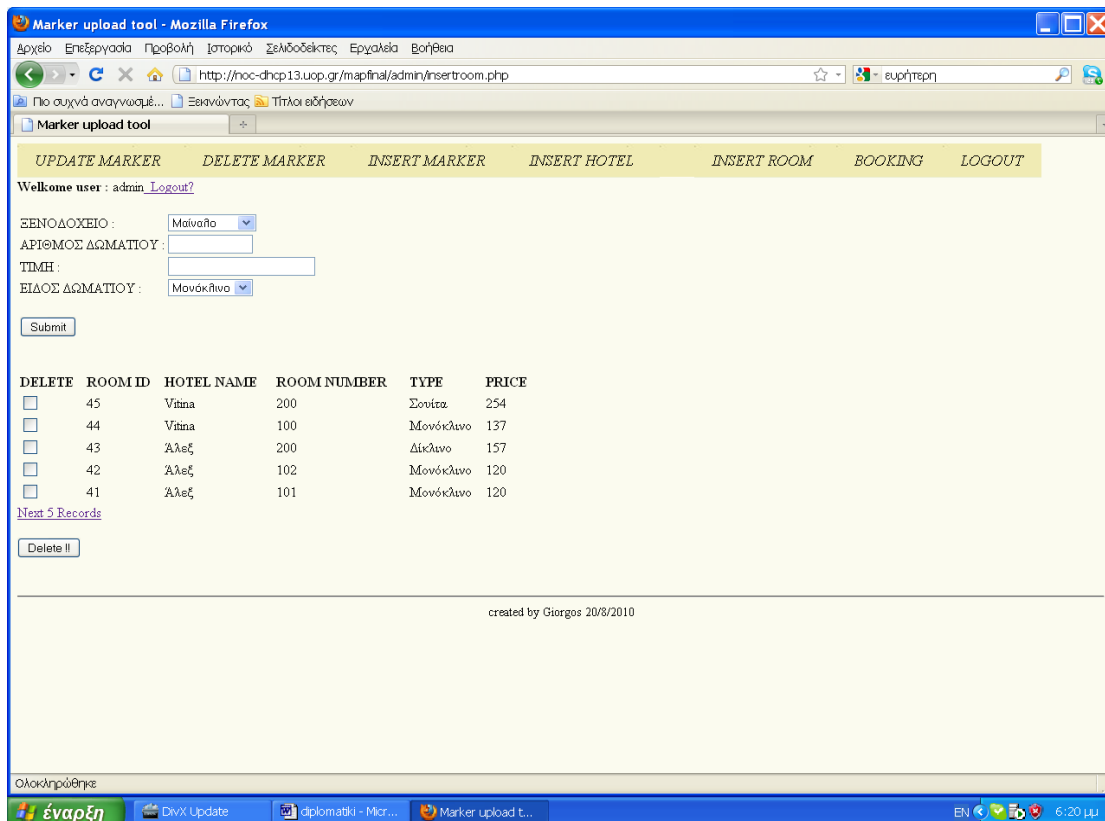
DELETE MARKER | Επιτρέπει την επιλογή πολλών σημείων ενδιαφέροντος και τη διαγραφή τους.

INSERT MARKER | Παρέχει ένα τα κατάλληλα εργαλεία για την εύρεση και εισαγωγή νέων σημείων ενδιαφέροντος στο χάρτη.

INSERT HOTEL | Επιτρέπει την εισαγωγή, ή διαγραφή ενός ξενοδοχείου.

INSERT ROOM | Παρέχει τη δυνατότητα της εισαγωγής ενός νέου δωματίου προς ενοικίαση καθώς και τη πιθανή διαγραφή του.

BOOKING | Από τη σελίδα αυτή ο διαχειριστής μπορεί να ενημερώνεται για τα δωμάτια που έχουν κρατηθεί.



Εικόνα 15: Η σελίδα διάθεσης δωματίων.

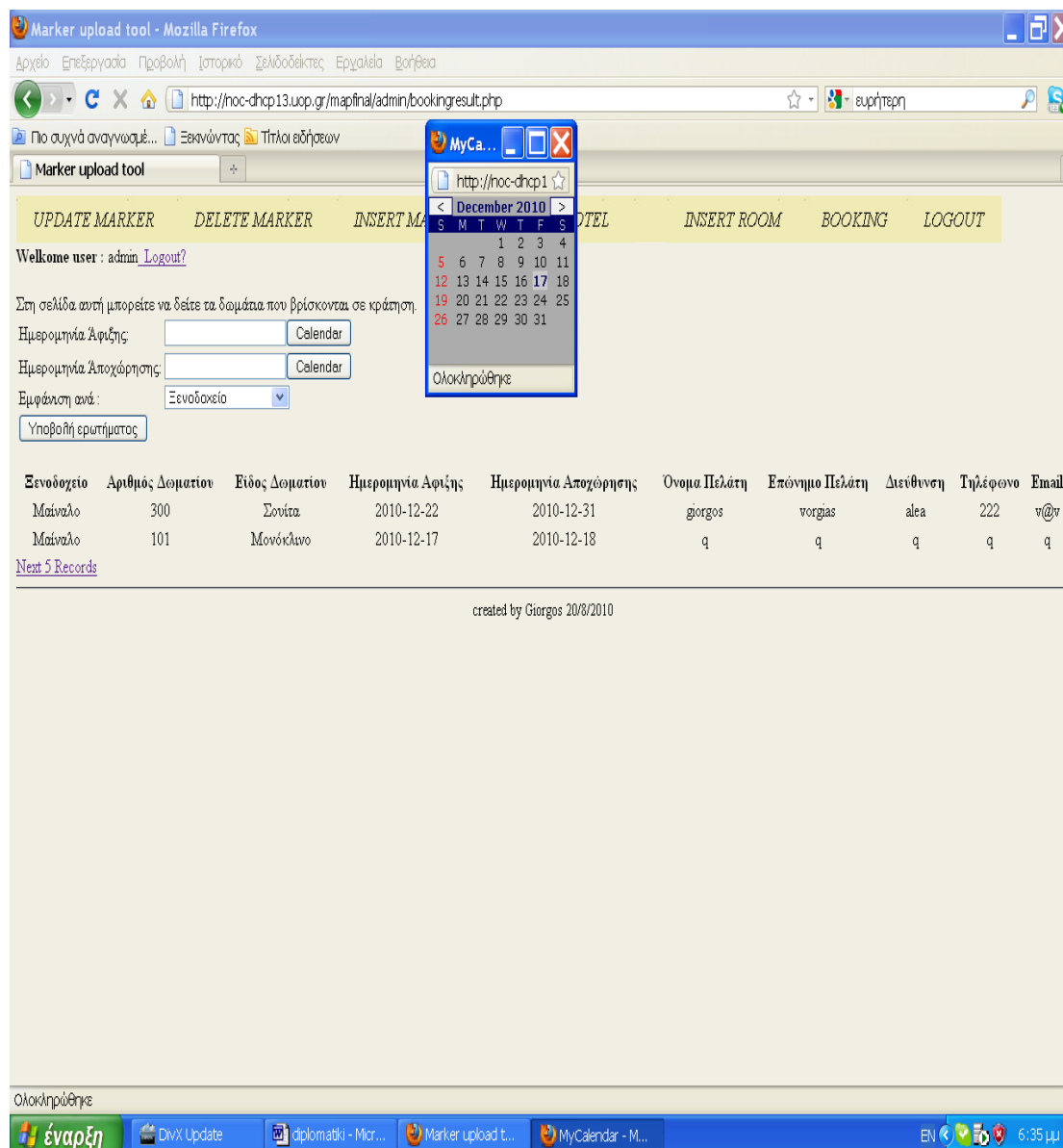
Ο τρόπος εισαγωγής και διαχείρισης περιεχομένου του χάρτη, είναι ίδιος με αυτόν που χρησιμοποιούν οι απλοί χρήστες, με την διαφορά ότι ο διαχειριστής έχει πλήρη δικαιώματα στη τροποποίηση του περιεχόμενου ενώ οι απλοί χρήστες μπορούν να τροποποιήσουν μόνο το περιεχόμενο που έχουν οι ίδιοι έχουν δημιουργήσει.

Από τη σελίδα BOOKING ο διαχειριστής ενημερώνεται για τα δωμάτια τα οποία βρίσκονται σε κράτηση . Αυτό γίνεται επιλέγοντας τα πεδία που αφορούν την ημερομηνία Άφιξης και την ημερομηνία Αποχώρησης, και εκτελώντας την υποβολή ερωτήματος. Τότε ο ρηρ κώδικας της σελίδας εκτελεί το παρακάτω sql ερώτημα προς την βάση και παρουσιάζει τα αποτελέσματα :

```
$sql = mysql_query("select hotelname, roomno, type, datefrom, dateto, name, lastname, address, telephone, email, bookid from hotel, room, booking where bookid in (select bookid from booking where ( datefrom <= '$Sdate' AND dateto >= '$edate') or (datefrom <= '$edate' AND dateto >= '$edate') or (datefrom <= '$Sdate' AND dateto >= '$Sdate') or (datefrom >= '$Sdate' AND dateto <= '$edate')) and booking.roomid=room.roomid and hotel.hotelno=room.hotelno order by '$cat' limit $offset, $rec_limit") or die(mysql_error());
```

Το αποτέλεσμα του ερωτήματος, είναι να μας εμφανίσει όλα τα δωμάτια, που έστω και μία ημέρα, από την περίοδο που βρίσκονται σε κράτηση, περιέχεται ανάμεσα στις ημερομηνίες που εισήχθησαν προηγουμένως.

Η ταξινόμηση των αποτελεσμάτων μπορεί να γίνει ανά ξενοδοχείο, τύπο δωματίου ή τιμή δωματίου.

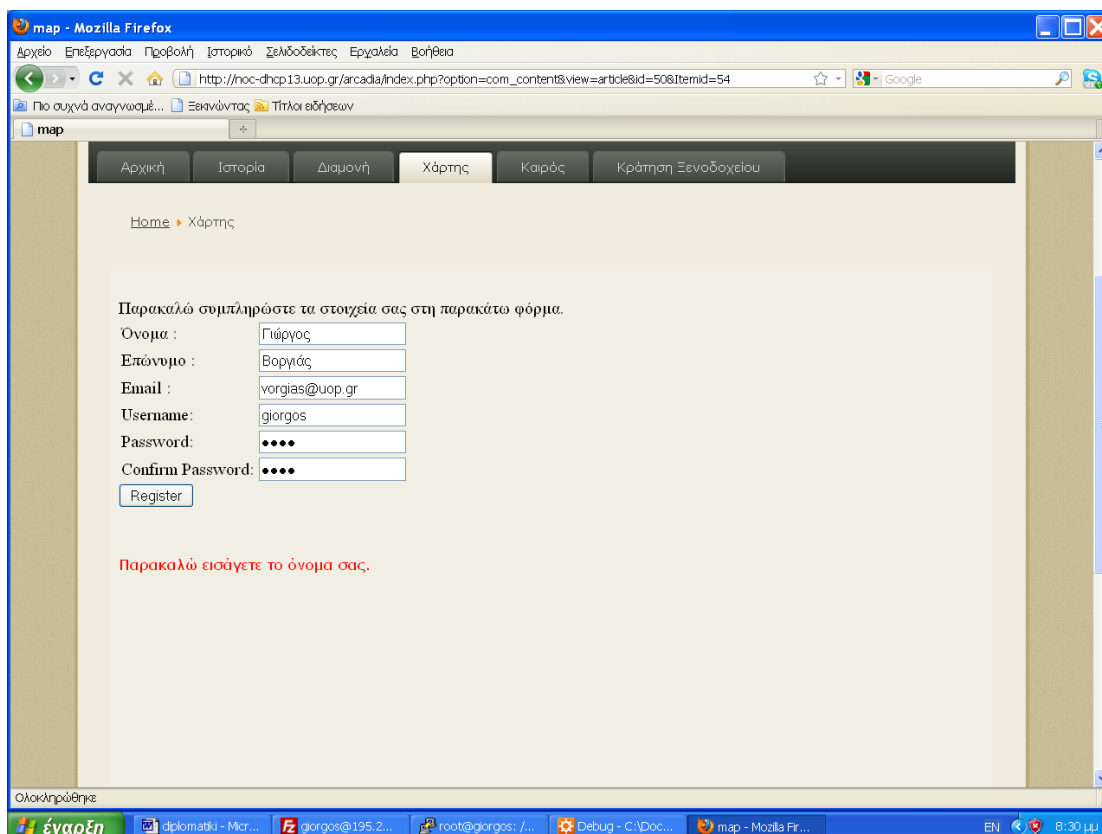


Εικόνα 16: Η ενημερωτική σελίδα κράτησης δωματίων.

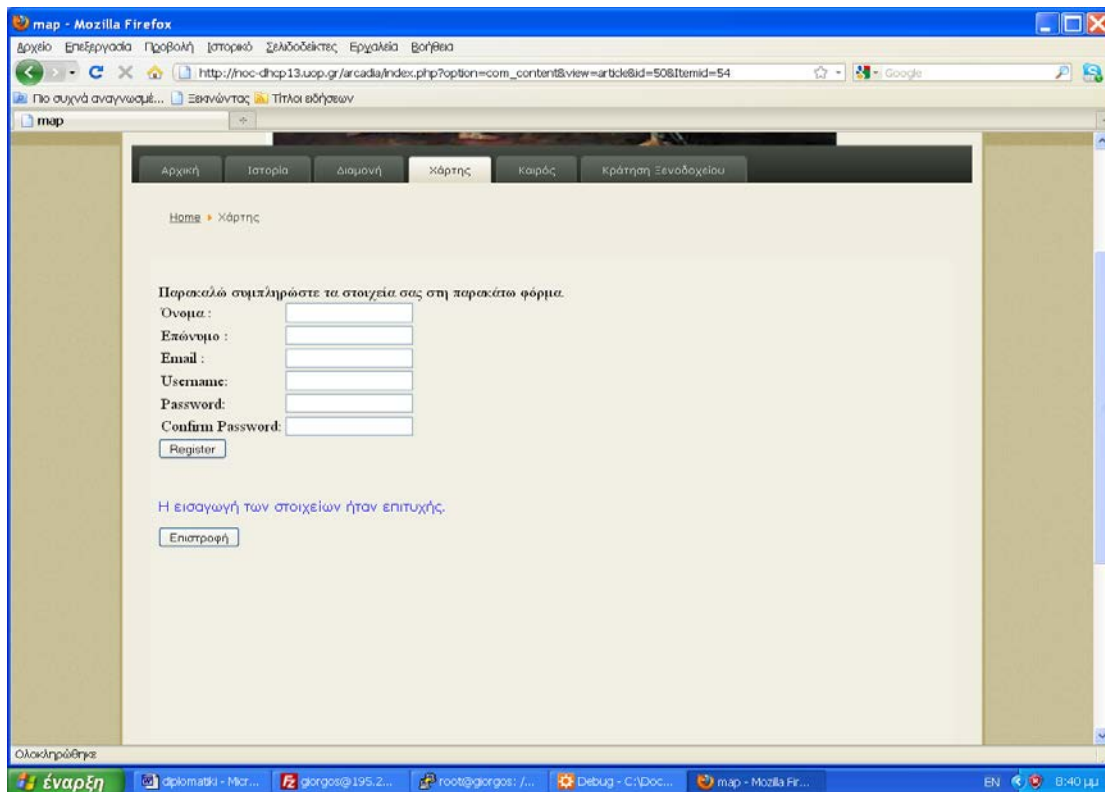
Κεφάλαιο 6 : Σενάρια Χρήσης

Στο κεφάλαιο αυτό θα δούμε τη λειτουργία του portal από τη πλευρά του χρήστη.

Για να έχει ο χρήστης πρόσβαση σε όλες τις υπηρεσίες του portal θα πρέπει να διαθέτει ένα λογαριασμό στο portal. Η δημιουργία ενός νέου λογαριασμού χρήστη, γίνεται από τη σελίδα με την ετικέτα χάρτης, και ακολουθώντας τον σύνδεσμο με την ονομασία Register. Στη φόρμα που εμφανίζεται, εισάγουμε τα προσωπικά μας στοιχεία, καθώς και όνομα χρήστη και password. Για τις ανάγκες της εργασίας θέτουμε έναν δοκιμαστικό χρήστη με username giorgos και password 1014, όπως βλέπουμε και στην εικόνα. Μετά τη επιτυχή καταχώρηση των στοιχείων μας, είμαστε σε θέση να συνδεθούμε, και να κάνουμε χρήση των προσφερόμενων υπηρεσιών με το username και password που μόλις δημιουργήσαμε.

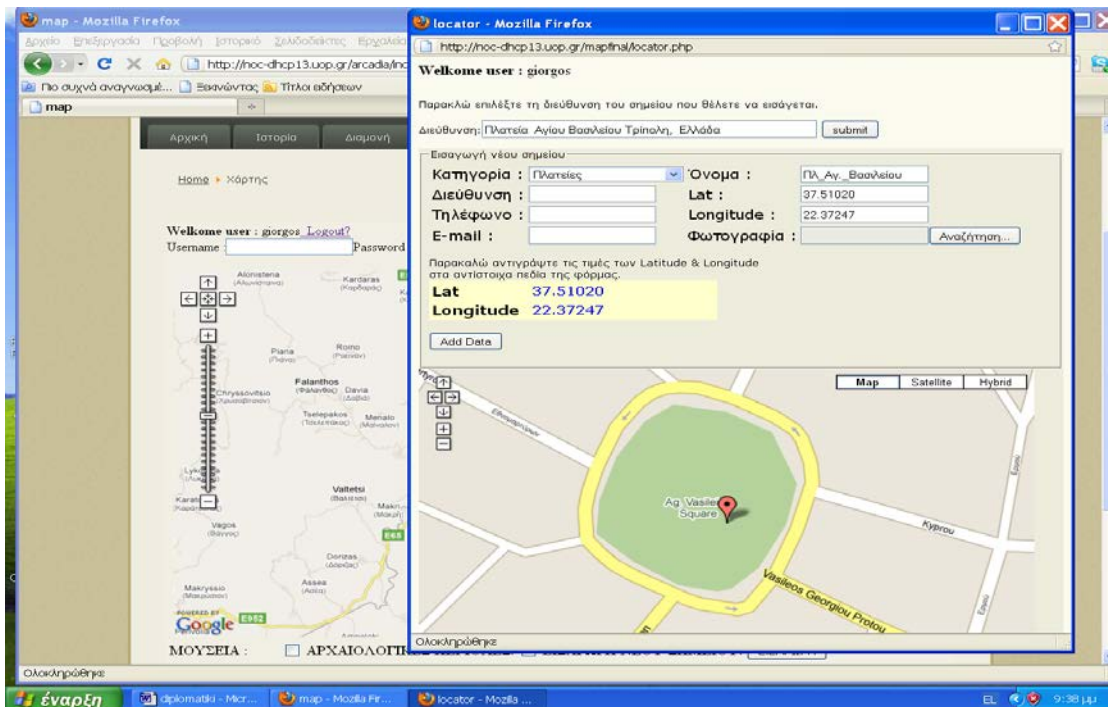


Εικόνα 17: Δημιουργία λογαριασμού χρήστη.



Εικόνα 18: Επιτυχής εισαγωγή στοιχείων.

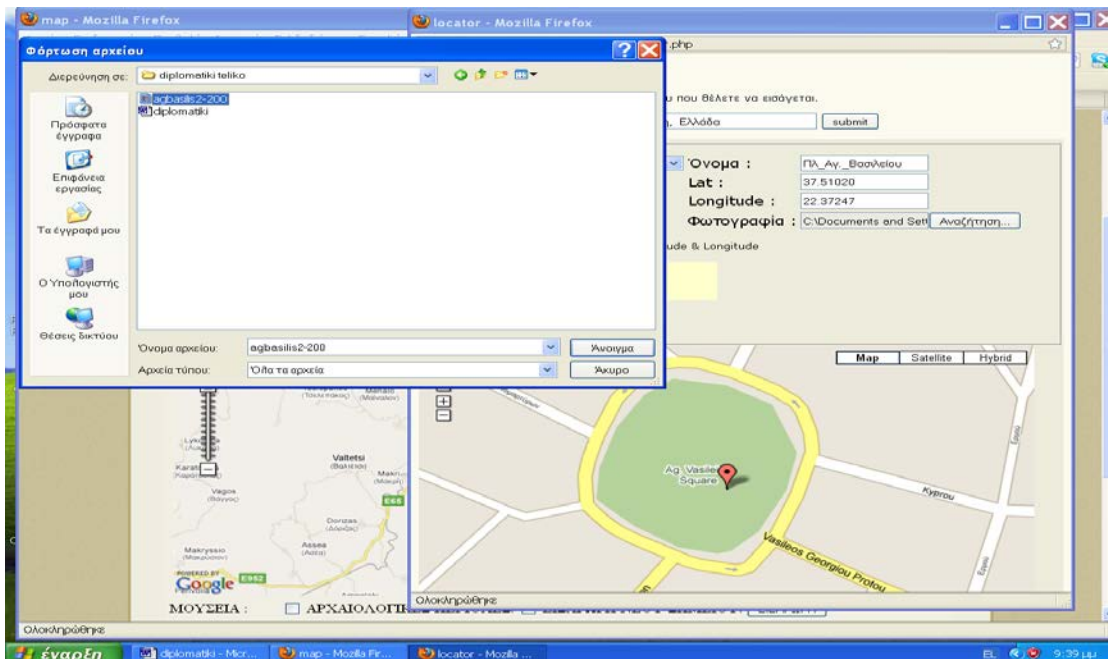
Αφού μεταβούμε πάλι στη σελίδα του χάρτη εισάγουμε το username και password που μόλις δημιουργήσαμε και πατάμε το πλήκτρο login, εάν η σύνδεση μας είναι επιτυχής λαμβάνουμε μήνυμα καλωσορίσματος από το σύστημα. Πλέον, είμαστε σε θέση να εισάγουμε τα προσωπικά μας σημεία ενδιαφέροντος στον χάρτη. Για τις ανάγκες της εργασίας θα εισάγουμε ως σημείο ενδιαφέροντος την πλατεία Αγ. Βασιλείου που βρίσκεται στην Τρίπολη. Πατώντας το Button με το όνομα «Εισαγωγή» εμφανίζεται η σελίδα εισαγωγής σημείων ενδιαφέροντος. Στη σελίδα αυτή πληκτρολογούμε τη διεύθυνση του σημείου ενδιαφέροντος που θέλουμε να εισάγουμε στο πεδίο με το όνομα «Διεύθυνση» ή οδηγούμαστε στο σημείο αυτό κάνοντας χρήση του ενσωματωμένου χάρτη που βρίσκεται στο τέλος της σελίδας. Στο παράδειγμά μας πληκτρολογώντας «Πλατεία Αγίου Βασιλείου, Τρίπολη, Ελλάδα» ο marker του χάρτη κεντράρεται στη συγκεκριμένη διεύθυνση και στο κίτρινο πλαίσιο εμφανίζονται οι γεωγραφικές συντεταγμένες του σημείου αυτού. Για να ολοκληρώσουμε την εισαγωγή επιλέγουμε μία από τις προκαθορισμένες επιλογές για το πεδίο "Κατηγορία". Για το παράδειγμά μας κάνουμε χρήση της επιλογής "Πλατείες". Στη συνέχεια, εισάγουμε ένα όνομα στο κατάλληλο πεδίο και αντιγράφουμε τις γεωγραφικές συντεταγμένες που βρήκαμε στα πεδία Latitude και Longitude (τα υπόλοιπα πεδία όπως Διεύθυνση, Τηλέφωνο, Email δεν τα χρησιμοποιούμε στο παράδειγμά μας).



Εικόνα 19: Αναζήτηση διεύθυνσης.

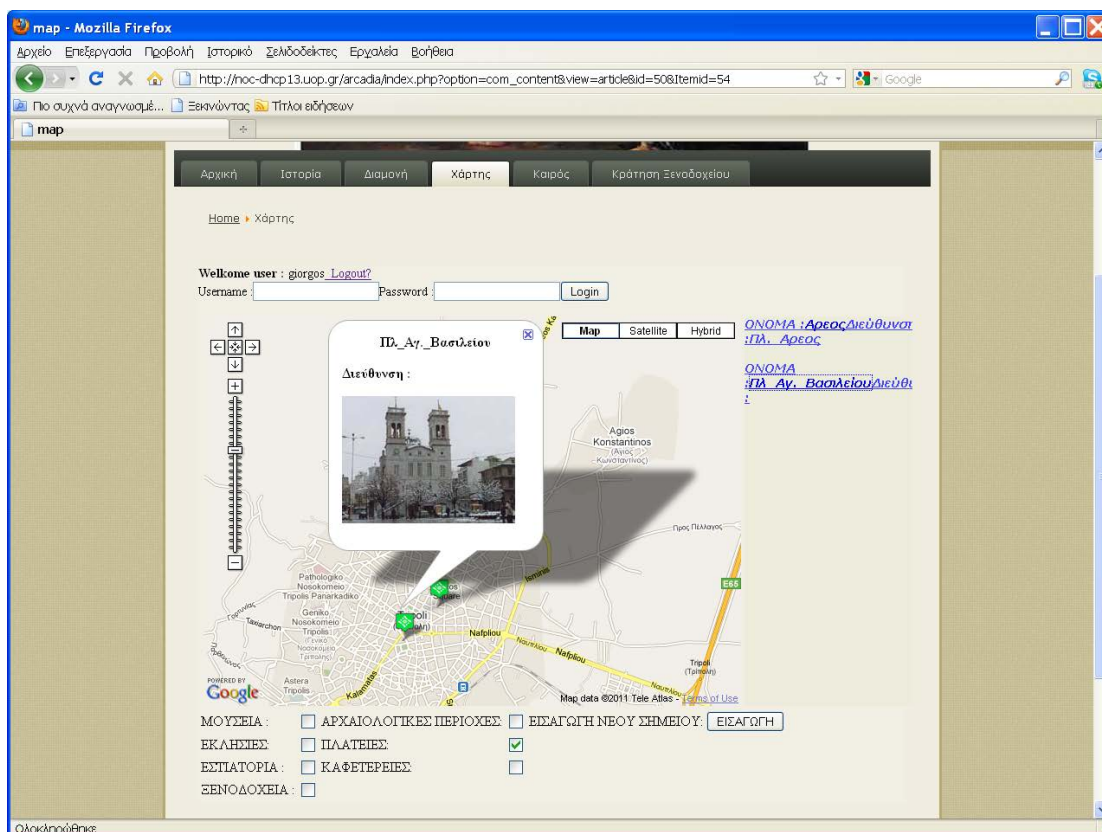
Για να προσθέσουμε μία φωτογραφία πατάμε το button «Αναζήτηση» που βρίσκεται δίπλα από το αντίστοιχο πεδίο, τότε εμφανίζεται σε παράθυρο ο φάκελος του υπολογιστή μας, που βρίσκονται αποθηκευμένες οι φωτογραφίες μας. Στο παράδειγμά μας επιλέγουμε τη φωτογραφία «agbasillis2-200», που παρουσιάζει μια άποψη της Πλατείας.

Η διαδικασία ολοκληρώνεται πατώντας το button «Add Data».



Εικόνα 20: Εισαγωγή φωτογραφίας.

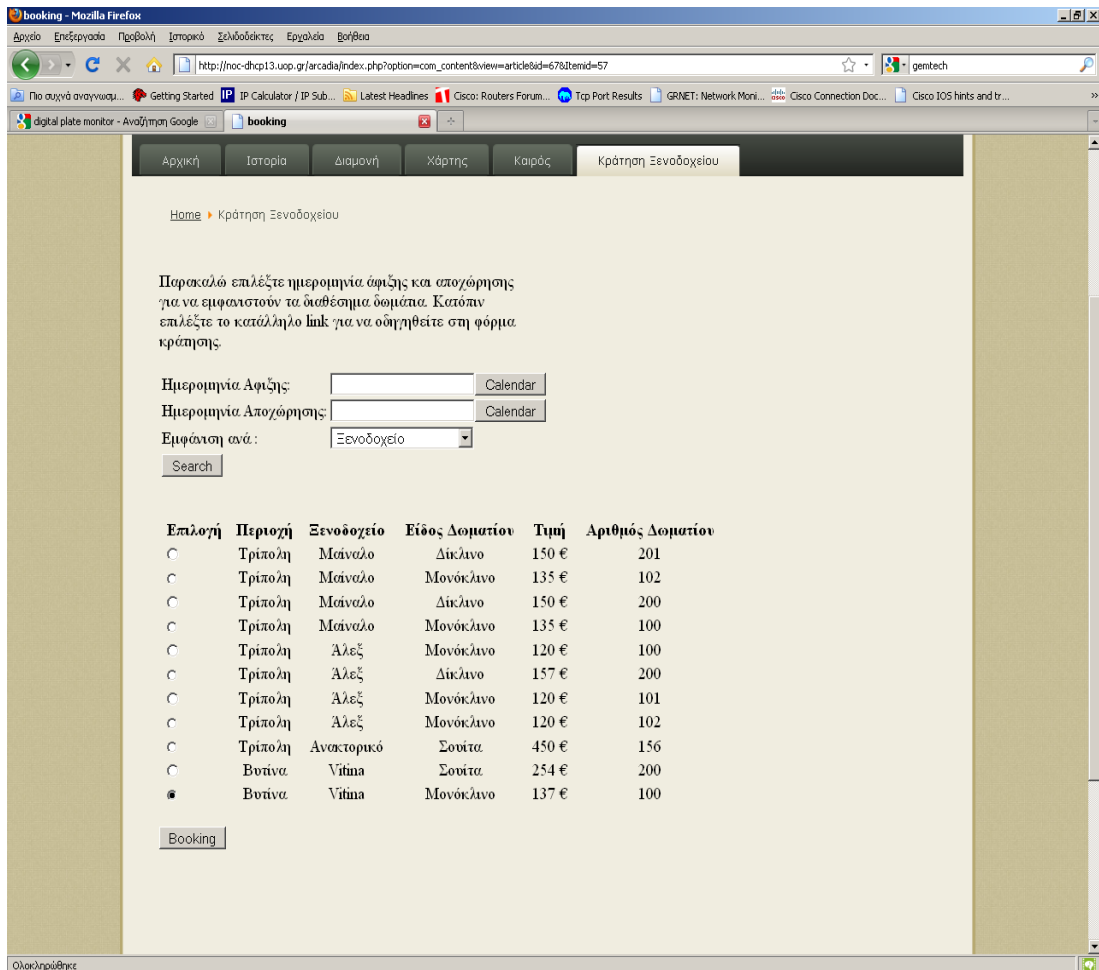
Μετά την ολοκλήρωση της διαδικασίας αυτής μπορούμε να αναζητήσουμε το σημείο που μόλις εισάγαμε και να το εμφανίσουμε στη σελίδα του χάρτη. Αυτό επιτυγχάνεται ως εξής: στη σελίδα του χάρτη επιλέγουμε την κατηγορία “ΠΛΑΤΕΙΕΣ”, αυτό έχει σαν αποτέλεσμα να εμφανιστεί στη δεξιά πλευρά της σελίδας μια λίστα με όλα τα σημεία ενδιαφέροντος της κατηγορίας αυτής. Ταυτόχρονα, τα σημεία αυτά εμφανίζονται και στο χάρτη με τη μορφή συγκεκριμένου εικονιδίου. Επιλέγοντας από τη λίστα την πλατεία με όνομα “Πλατεία Αγ. Βασιλείου”, από τον marker που αντιστοιχεί στο σημείο αυτό αναδύεται ένα παράθυρο πληροφοριών που περιέχει το όνομα, τη διεύθυνση και τη φωτογραφία του σημείου αυτού.



Εικόνα 21:Εμφάνιση σημείου ενδιαφέροντος στο χάρτη.

Για να πραγματοποιηθεί μια κράτηση ενός δωματίου ξενοδοχείου, όπως είδαμε και στο κεφάλαιο 4, θα πρέπει να μεταβούμε στη σελίδα «Κράτηση Ξενοδοχείου».

Στο παραδειγμά μας θα πραγματοποιήσουμε μια αναζήτηση για ένα μονόκλινο δωμάτιο στη περιοχή της Βυτίνας κατά το χρονικό διάστημα μεταξύ 23-12-2010 και 26-12-2010. Επιλέγοντας τις ανάλογες ημερομηνίες Άφιξης και Αναχώρησης εμφανίζεται η λίστα με τα διαθέσιμα δωμάτια ανά ξενοδοχείο που βρίσκονται στο portal μας (Εικόνα 22).



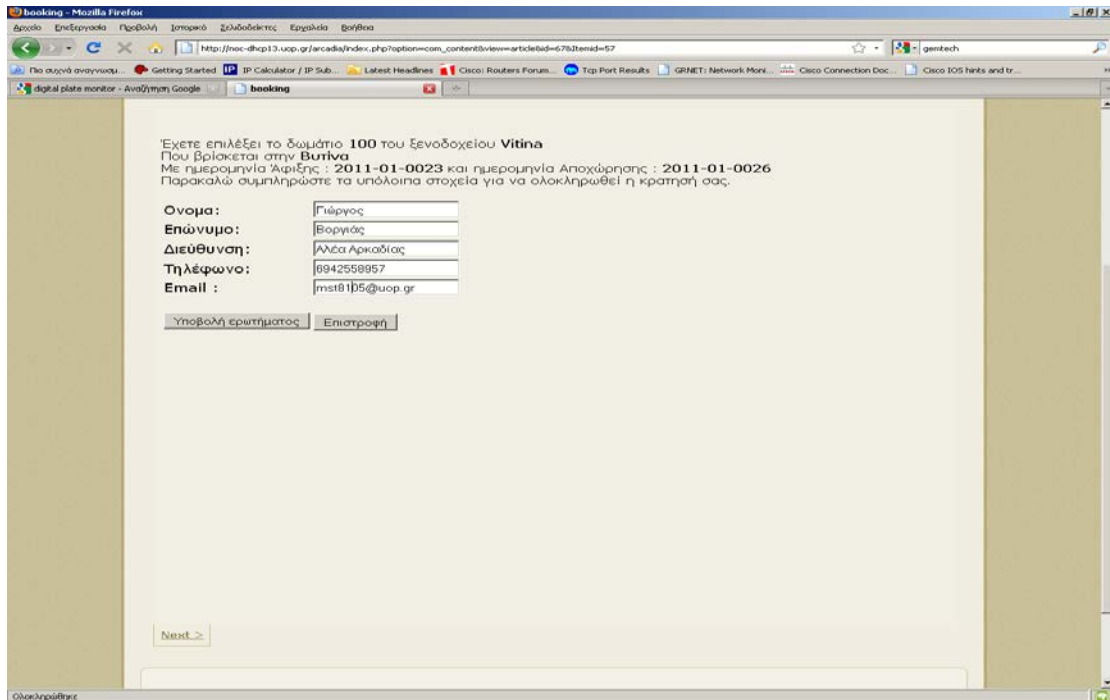
Εικόνα 22: Εμφάνιση ελεύθερων δωματίων.

Επιλέγουμε το δωμάτιο που μας ενδιαφέρει, στο παράδειγμά μας αυτό είναι το δωμάτιο με αριθμό 100, του Ξενοδοχείου Vithina και επιλέγοντας Booking οδηγούμαστε στη σελίδα κράτησης (εικόνα 23). Στη σελίδα αυτή βλέπουμε αναλυτικά τα στοιχεία του δωματίου που επιλέξαμε και καλούμαστε να συμπληρώσουμε τα προσωπικά μας στοιχεία που είναι απαραίτητα για να ολοκληρωθεί η κράτησή μας.

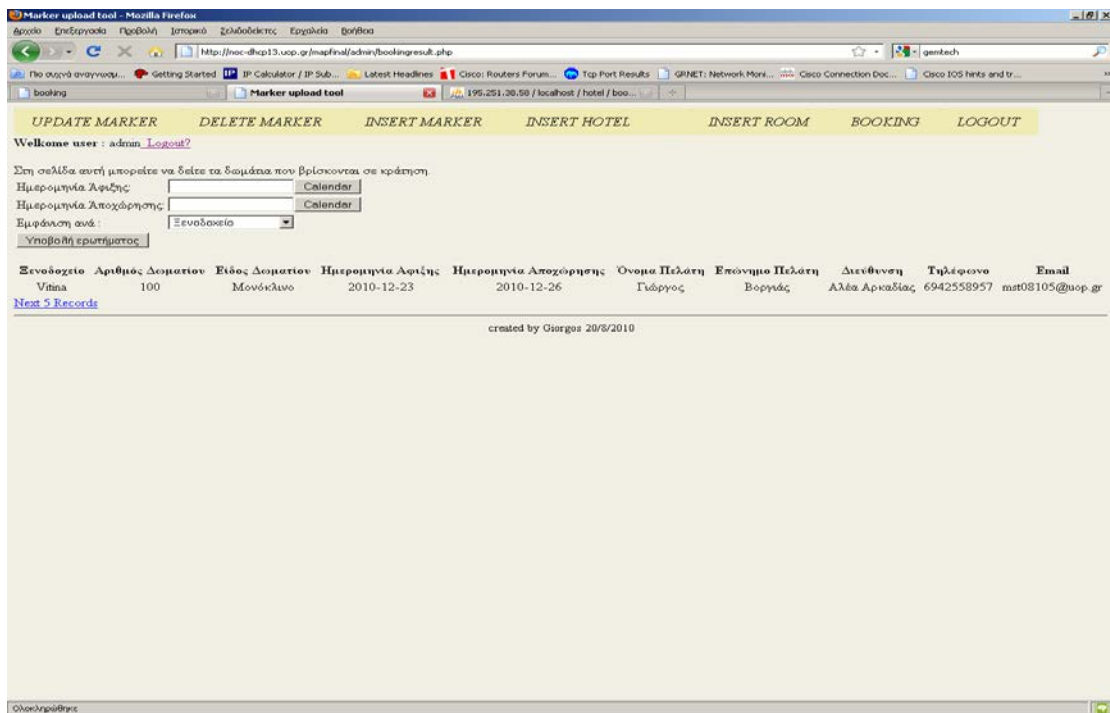
Όταν ολοκληρωθεί η διαδικασία λαμβάνουμε ένα μήνυμα επιτυχούς καταχώρησης. Το portal έχει πλέον ενημερωθεί για την κράτηση του συγκεκριμένου δωματίου και το έχει αφαιρέσει από τη λίστα των ελεύθερων δωματίων για το συγκεκριμένο χρονικό διάστημα.

Τα δεδομένα της κράτησής μας είναι διαθέσιμα μόνο για τον διαχειριστή του portal. Η ενημέρωση του διαχειριστή γίνεται από τη σελίδα διαχείρισης που βρίσκεται στη διεύθυνση <http://noc-dhcp13.uop.gr/mapfinal/admin/>.

Εφόσον συνδεθούμε στη διαχειριστική σελίδα με τους κατάλληλους κωδικούς, μπορούμε να ενημερωθούμε για τα δωμάτια που βρίσκονται σε κράτηση. Η ενημέρωση γίνεται από τη σελίδα "BOOKING" όπου, κάνοντας αναζήτηση για τη χρονική περίοδο που επιθυμούμε, μπορούμε να ενημερωθούμε για τα δωμάτια που βρίσκονται σε κράτηση (εικόνα 24).



Εικόνα 23 : Η σελίδα κράτησης δωματίου



Εικόνα 24: Τα στοιχεία κράτησης στη σελίδα του διαχειριστή.

Κεφάλαιο 7 : Επίλογος – Συμπεράσματα

Η εργασία για την κατασκευή ενός portal δεν τελειώνει ποτέ. Η βιωσιμότητά του εξαρτάται από τη συχνή και ποιοτική ενημέρωση του περιεχομένου του. Ταυτόχρονα, απαιτούνται συνεχείς προσαρμογές και επεκτάσεις. Συμπερασματικά, θα μπορούσε να ειπωθεί ότι η εφαρμογή των επιτευγμάτων και των ιδεών του Web 2.0, και ιδιαίτερα των mashup εφαρμογών, δίνουν ώθηση στην εξελικτική πορεία του Web.

Σαν επακόλουθο των παραπάνω, οι μελλοντικές προσθήκες που θα μπορούσαν να γίνουν έχοντας σαν βάση την υπάρχουσα εργασία είναι:

- Στο κομμάτι της υλοποίησης θα μπορούσε εναλλακτικά να γίνει χρήση της βιβλιοθήκης openlayer και να παραμείνει ο χάρτης της Google ως χαρτογραφικό υπόβαθρο.
- Στο τομέα παρουσίασης και κράτησης ξενοδοχείων θα μπορούσε να ενσωματωθεί ένα σύστημα βαθμολόγησής τους από τους χρήστες.
- Ενσωμάτωση ενός Forum χρηστών, ώστε να παρέχεται η δυνατότητα επικοινωνίας και ανταλλαγής απόψεων στους χρήστες .

Αναφορές – Βιβλιογραφία

- How to do everything with PHP & Mysql Vikram Vaswani ISBN: 0-07-225795-4.
- PHP Cookbook | OReilly ISBN: 1- 56592-681-1.
- JavaScript by example | Ellie Quigley ISBN: 0-13-140162-9.
- Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional | (by Michael Purvis, Jeffrey Sambells, and Cameron Turner) ISBN-10 (pbk): 1-59059-707-9.
- How to Do Everything with Web 2.0 Mashups | Jesse Feiler.
- Beginning PHP, Apache, MySQL Web Development | Published by Wiley Publishing, Inc. ISBN: 0-7645-5744-0.
- Database Applications with PHP and MySQL | Published by M&T Books ISBN: 0-7645-3537-4.
- CSS, DHTML and JavaScript | Prentice Hall.
- PHP *Fast & Easy Web Development*, 2nd Edition | by Premier Press ISBN: 1-931841-87-X.
- JavaScript Bible, 5th Edition | Published by Wiley Publishing, Inc ISBN: 0-7645-5743-2.
- Web Design - Creating Cool Web Sites With HTML, XHTML, and CSS | John Wiley & Sons ISBN: 0-7645-5738-6.

- What Is Web 2.0 | <http://oreilly.com/web2/archive/what-is-web-20.html>

- API Dashboard | <http://www.programmableweb.com/apis>
- Ajax: A New Approach to Web Applications | <http://adaptivepath.com/ideas/essays/archives/000385.php>
- Using PHP, MySQL with Google Maps | <http://code.google.com/intl/el-GR/apis/maps/articles/phpsqlajax.html>

- Google Maps API Tutorial | <http://econym.org.uk/gmap/>

- Ajax Help and Tutorials Ajax Tutorial | <http://www.ajaxtutorial.net/>

- MySQL Documentation | <http://dev.mysql.com/doc/>

- PhpMyadmin | <http://www.phpmyadmin.net/>

- Aptana Studio | <http://aptana.com/>

- Joomla! | <http://www.joomla.org/>

Παράρτημα | Πηγαίος Κώδικας

phpsqlajax_dbinfo.php

```
<?php
$username="root";
$password="";
$databse="map";
?>
```

phpsqlajax_genxml31.php

```
<?php
session_start();
if(!session_is_registered(myusername)){
header("location:index.php");

//leitoyrgei mono me anakateythinsi kai oxi ma minima
}

//echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<a
href=Logout.php>&nbsp;&nbsp; Logout?</a><br><br>";

require("phpsqlajax_dbinfo.php");

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

// Opens a connection to a MySQL server
$connection=mysql_connect (localhost, $username, $password);
if (!$connection) {
die('Not connected : ' . mysql_error());
}
```

```

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

// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE
user='$_SESSION[myusername]' or user='admin'";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

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

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

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

    echo 'photo=' . parseToXML($row['photo']) . ' ';
    echo 'telephone=' . parseToXML($row['telephone']) . ' ';
    echo 'email=' . parseToXML($row['email']) . ' ';
    echo 'description=' . parseToXML($row['description']) . ' ';

    echo 'lat=' . $row['lat'] . ' ';

    echo 'lng=' . $row['lng'] . ' ';

    echo '>';
}

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

?>

```

index.php

```
<?php
session_start();

?>
<head>

    <title>Google Maps</title>
    <meta http-equiv="Content-Type" charset="utf-8" >

    <script
src="http://maps.google.com/maps?file=api&v=2&hl=en&oe=utf-
8&key=ABQIAAAACKdP_sh-kO6Gum4UG6C_ihTxIbXNLk-
qv272wwLNZrr5dxWxRQmbzowlqHrbnDOLM7WZ_TW6A8KA"
type="text/javascript"></script>

</head>
<body onunload="GUnload()">

<div id="condainer" style="background-color: #F1EFE4; font-
family: Vrinda; font-size: 14px; ">
    <!-- you can use tables or divs for the overall layout -->
    <?php
    if(!session_is_registered(myusername)){

echo " Θα πρέπει να γραφτείτε για να δείτε το προσωπικό σας χάρτη ή
κάντε login με τα admin & 1014gv .<a
href=registration.php>&nbsp;&nbsp; Register?</a>";
//leitoyrgei mono me anakateythinsi kai oxi ma minima
}
else if(session_is_registered(myusername)){
echo "<b>Welkome user : </b>".$_SESSION['myusername'].<a
href=Logout.php>&nbsp;&nbsp; Logout?</a><br>";
}
?>
<form id="form2" action="checklogin1.php" method="post">Username
:<input name="myusername" type="text" id="myusername">Password
:<input name="mypassword" type="text" id="mypassword"><input
type="submit" name="Submit" value="Login"></form>
    <table border=0>
        <tr>
            <td>
                </td>
            <td>
                <div id="map" style="width: 620px; height: 450px"><div
id="selection"></div></div>
            </td>
        </tr>
    </table>
</div>
```



```

        <td width = 450 valign="top" style="text-decoration: underline;
color: #4444ff;">
        <div id="side_bar" style="overflow:auto; height:450px; font-
family:Verdana; font-size:14px; font-weight:bold; font-style:italic;
background-color:#EDECDC; text-decoration:none;"> </div>
        </td>
</tr>

</table>
<form action="#">
    <table>
<tr><td><label>ΜΟΥΣΕΙΑ :</label></td>
<td>        <label><input          type="checkbox"          id="mouseibox"
onclick="boxclick(this,'mouseia')" /></label></td>
<td><label>ΑΡΧΑΙΟΛΟΓΙΚΕΣ ΠΕΡΙΟΧΕΣ:</label></td>
<td><label><input          type="checkbox"          id="arxaibox"
onclick="boxclick(this,'arxaia')" /></label></td>

    <td><label>ΕΙΣΑΓΩΓΗ  ΝΕΟΥ  ΣΗΜΕΙΟΥ:  <a href:"locator.php"
onclick="window.open('locator.php', 'myWin', 'toolbar=no,
directories=no, location=no, status=yes, menubar=no, resizable=no,
scrollbars=no, width=640, height=780'); return
false"><button>ΕΙΣΑΓΩΓΗ</button></a>
</tr>
<tr><td><label>ΕΚΛΗΣΙΕΣ: </label></td>

<td><label><input          type="checkbox"          id="eklisiesbox"
onclick="boxclick(this,'eklisies')" /></label></td>
<td><label>ΠΛΑΤΕΙΕΣ:</label></td>
<td>        <label><input          type="checkbox"          id="plateiesbox"
onclick="boxclick(this,'plateies')" /></label></td></tr>
<tr><td><label>ΕΣΤΙΑΤΟΡΙΑ :</label></td>
<td>        <label><input          type="checkbox"          id="restaurantbox"
onclick="boxclick(this,'restaurant')" /></label></td>
<td><label>ΚΑΦΕΤΕΡΕΙΕΣ: </label></td>
<td>        <label><input          type="checkbox"          id="coffebox"
onclick="boxclick(this,'coffe')" /></label></td></tr>
<tr><td><label>ΞΕΝΟΔΟΧΕΙΑ :</label></td>
<td><label>        <input          type="checkbox"          id="hotelbox"
onclick="boxclick(this,'hotel')" /></label> </td> </tr>
</table>
</form>

</div>

```

<noscript>JavaScript must be enabled in order for you to use Google Maps.</noscript>

However, it seems JavaScript is either disabled or not supported by your browser.

To view Google Maps, enable JavaScript by changing your browser options, and then

try again.

</noscript>

```
<script type="text/javascript">
//</pre></div><div data-bbox="171 292 459 324" data-label="Text"><pre>if (GBrowserIsCompatible()) {
//select menou-----</pre></div><div data-bbox="184 340 401 373" data-label="Text"><pre>    var gmarkers = [];
    var gicons = [];</pre></div><div data-bbox="184 435 640 500" data-label="Text"><pre>    var baselcon = new GIcon(G_DEFAULT_ICON);
    baselcon.iconAnchor = new GPoint(16,16);
    baselcon.iconSize = new GSize(25,25);
    baselcon.infoWindowAnchor = new GPoint(5,1);</pre></div><div data-bbox="144 530 853 769" data-label="Text"><pre>// eikonidia
    gicons["arxaia"] = new GIcon(baselcon,'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/ancienttemple.png');
    gicons["restaurant"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/restaurant.png');
    gicons["coffe"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/fastfood.png');
    gicons["hotel"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/hotel.png');
    gicons["mouseia"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/museum-historical.png');
    gicons["eklisies"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/church2.png');
    gicons["plateies"] = new GIcon(baselcon, 'http://noc-dhcp13.uop.gr/mapfinal/icons/icons/citysquare.png');</pre></div><div data-bbox="144 783 853 880" data-label="Text"><pre>    // A function to create the marker and set up the event window
    function createMarker(point,name,address,user,html,category) {
        var marker = new GMarker(point,gicons[category]);
        // === Store the category and name info as a marker properties
        ===
        marker.mycategory = category;</pre></div><div data-bbox="471 937 523 955" data-label="Page-Footer"><p>- 50 -</p></div>
```

```

marker.myname = name;
    marker.myaddress = address;
    marker.myuser = user;

GEvent.addListener(marker, "click", function() {
    marker.openInfoWindowHtml(html);

});

gmarkers.push(marker);

return marker;
}

// == shows all markers of a particular category, and ensures the
checkbox is checked ==
function show(category) {
    for (var i=0; i<gmarkers.length; i++) {
        if (gmarkers[i].mycategory == category) {
            gmarkers[i].show();

        }
    }
}
// == check the checkbox ==

document.getElementById(category+"box").checked = true;

}

// == hides all markers of a particular category, and ensures the
checkbox is cleared ==
function hide(category) {
    for (var i=0; i<gmarkers.length; i++) {
        if (gmarkers[i].mycategory == category ) {
            gmarkers[i].hide();

        }
    }
}
// == clear the checkbox ==
document.getElementById(category+"box").checked = false;
// == close the info window, in case its open on a marker that we
just hid
map.closeInfoWindow();
}

// == a checkbox has been clicked ==

```

```

function boxclick(box,category) {
    if (box.checked) {
        show(category);
    } else {
        hide(category);
    }
    // == rebuild the side bar
    makeSidebar();
}

function myclick(i) {
    GEvent.trigger(gmarkers[i], "click");
}

// == rebuilds the sidebar to match the markers currently displayed
==
function makeSidebar() {
    var html = "";
    for (var i=0; i<gmarkers.length; i++) {
        if (!gmarkers[i].isHidden()) {
            html += 'ONOMA :<a href="javascript:myclick(' + i + ')" >' +
gmarkers[i].myname + '</a><b>Διεύθυνση :</b>'+
gmarkers[i].myuser + '<br><br>';
        }
    }
    document.getElementById("side_bar").innerHTML = html;
}

// create the map
var map = new GMap2(document.getElementById("map"));
map.addControl(new GLargeMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.50980, 22.37287), 11);

// Read the data
GDownloadUrl("phpsqlajax_genxml31.php", function(doc) {
    var xmlDoc = GXml.parse(doc);
    var markers =
xmlDoc.documentElement.getElementsByTagName("marker");

    for (var i = 0; i < markers.length; i++) {
        // obtain the attributes of each marker
        var lat = parseFloat(markers[i].getAttribute("lat"));
        var lng = parseFloat(markers[i].getAttribute("lng"));
        var point = new GLatLng(lat, lng);
        var address = markers[i].getAttribute("address");
    }
}

```

```

var name = markers[i].getAttribute("name");
var telephone = markers[i].getAttribute("telephone");
var photo = markers[i].getAttribute("photo");
var user = markers[i].getAttribute("user");
var html = "<b><center>" + name + "</center></b><p><b>Διεύθυνση
: </b>" + address + "<p><img src=http://noc-
dhcp13.uop.gr/mapfinal/uploads/" + photo + ">";
var category = markers[i].getAttribute("category");
// create the marker
var marker =
createMarker(point, name, photo, address, html, category, telephone);
map.addOverlay(marker);
}

hide("mouseia");
hide("plateies");
hide("arxaia");
hide("restaurant");
hide("eklisies");
hide("coffe");
hide("hotel");

// == create the initial sidebar ==
makeSidebar();
});
}

else {
alert("Sorry, the Google Maps API is not compatible with this
browser");
}

</script>
</body>

```

checklogin1.php

```
<?php
ob_start();
$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="map"; // Database name
$table_name="members"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot
connect");
mysql_select_db("$db_name")or die("cannot select DB");

// Define $myusername and $mypassword
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

//
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);

$sql="SELECT * FROM $table_name WHERE username='$myusername' and
password='$mypassword'";
$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $myusername and $mypassword, table row must be
1 row

if($count==1){
// Register $myusername, $mypassword and redirect to file "index.php"
session_register("myusername");
session_register("mypassword");

header("location: index.php");
}
else {

header("location: index.php");
}

ob_end_flush();
?>
```

upload.php

```
<?
session_start();
if(!session_is_registered(myusername)){
header("location:index.php");

//leitoyrgei mono me anakateythinsi kai oxi ma minima
}

echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<a
href=Logout.php>&nbsp;&nbsp; Logout?</a><br><br>";
?><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">
<!--
P{
font-size : 14px;
font-family : Verdana;
}
body {background-color: #FBFBEB;}
.form1 {
font-size : 11px;
font-family : Verdana;
background-color: #EDECDC
}
.container{
font-size: 11px;
font-family: Verdana;
background-color: #FBFBEB
}
}
</style>
<script type="text/javascript">
//redirect function gia close window
function redirect(linkid)
{
opener.location.href=linkid
window.close()
}

</script>
</head>
<body>

<?php
/** Arxeio gia enimerosi tis vaseis gia google map
include 'deletephp/header.php';
$con = mysql_connect("localhost","root","");
```

```

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

mysql_select_db("map", $con);

//
//

$allowed_filetypes = array('.jpg','.gif','.bmp','.png','.psd','.jpeg',''); //
These will be the types of file that will pass the validation.
$max_filesize = 10485760; // Maximum filesize in BYTES.
$upload_path = './uploads/'; // The place the files will be uploaded to
(currently a 'files' directory).
$filename = $_FILES['userfile']['name']; // Get the name of the file
(including file extension).
$ext = substr($filename, strpos($filename,'.'), strlen($filename)-1); //
Get the extension from the filename.
// Check if the filetype is allowed, if not DIE and inform the user.
//
if(!in_array($ext,$allowed_filetypes))
die('The file you attempted to upload is not allowed .Close this window');

//
// Now check the filesize, if it is too large then DIE and inform the user.
if(filesize($_FILES['userfile']['tmp_name']) > $max_filesize)
die('The file you attempted to upload is too large.');
```

```

//
// Check if we can upload to the specified path, if not DIE and inform the
user.
if(!is_writable($upload_path))
die('You cannot upload to the specified directory, an error is present.');
```

```

// Upload the file to your specified path.
if(move_uploaded_file($_FILES['userfile']['tmp_name'],$upload_path
.$filename))
//Gia periptosi poy den exo foto
if(filesize($_FILES['userfile']['tmp_name']) == 0)
$filename1="no photo";
else
$filename1=$_FILES['userfile']['name'];
//echo 'Your file upload was successful.'; // It worked.
$sql="INSERT INTO markers (category, address, email, telephone, name,
photo, lat, lng, user) VALUES ('$_POST[category]', '$_POST[address]',
'$_POST[email]', '$_POST[telephone]', '$_POST[name]', '$filename1',
'$_POST[lat1]', '$_POST[lng1]', '$_SESSION[myusername] )";

if (!mysql_query($sql,$con))
{

```



```
die('Error: ' . mysql_error());
}
?><br><br><P><?php
echo "Your file upload was successful.";
?>
<br><br><?php
mysql_close($con);
?>
```

locator.php

```
<?php
session_start();
//javascript close popup
if(!session_is_registered(myusername)){?>
//javascript mesa se php script
<SCRIPT LANGUAGE="JavaScript">

window.close();

</SCRIPT>
<?>
echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<br>";
?>
<head>
    <meta http-equiv="Content-Type" charset=utf-8" ">

<title>locator</title>
<script
src="http://maps.google.com/maps?file=api&v=2&hl=en&oe=utf-
8&key=ABQIAAAACKdP_sh-kO6Gum4UG6C_ihTxIbXNLk-
qv272wwLNZrr5dxWaxRQmbzowlqHrbnDOLM7WZ_TW6A8KA"
type="text/javascript"></script>

    <script src="jquery.js" type="text/javascript"></script>
    <script type="text/javascript" src="jquery-
validate/jquery.validate.min.js"></script>

    <script type="text/javascript">

function load() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map"));
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
        var center = new GLatLng(    37.50980, 22.37287);
        map.setCenter(center, 15);
        geocoder = new GClientGeocoder();
        var marker = new GMarker(center, { draggable: true});
        map.addOverlay(marker);
        document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

        GEvent.addListener(marker, "dragend", function() {
            var point = marker.getPoint();
            map.panTo(point);
```

```

document.getElementById("lat").innerHTML = point.lat().toFixed(5);
document.getElementById("lng").innerHTML = point.lng().toFixed(5);

});

    GEvent.addListener(map, "moveend", function() {
        map.clearOverlays();
    var center = map.getCenter();
        var marker = new GMarker(center, {draggable: true});
        map.addOverlay(marker);
        document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

        GEvent.addListener(marker, "dragend", function() {
    var point = marker.getPoint();
        map.panTo(point);
        document.getElementById("lat").innerHTML = point.lat().toFixed(5);
        document.getElementById("lng").innerHTML =
point.lng().toFixed(5);

        });

    });

}
}

function showAddress(address) {
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GSmallMapControl());
    map.addControl(new GMapTypeControl());
    if (geocoder) {
        geocoder.getLatLng(
            address,
            function(point) {
                if (!point) {
                    alert(address + " not found");
                } else {
                    document.getElementById("lat").innerHTML =
point.lat().toFixed(5);
                    document.getElementById("lng").innerHTML =
point.lng().toFixed(5);
                    map.clearOverlays()
                    map.setCenter(point, 14);
                    var marker = new GMarker(point, {draggable: true});
                    map.addOverlay(marker);

```

```

        GEvent.addListener(marker, "dragend", function() {
var pt = marker.getPoint();
    map.panTo(pt);
    document.getElementById("lat").innerHTML = pt.lat().toFixed(5);
    document.getElementById("lng").innerHTML =
pt.lng().toFixed(5);
    });

        GEvent.addListener(map, "moveend", function() {
    map.clearOverlays();
var center = map.getCenter();
    var marker = new GMarker(center, {draggable: true});
    map.addOverlay(marker);
    document.getElementById("lat").innerHTML =
center.lat().toFixed(5);
    document.getElementById("lng").innerHTML =
center.lng().toFixed(5);

        GEvent.addListener(marker, "dragend", function() {
var pt = marker.getPoint();
    map.panTo(pt);
    document.getElementById("lat").innerHTML = pt.lat().toFixed(5);
    document.getElementById("lng").innerHTML =
pt.lng().toFixed(5);
    });
    });
    }
    }
    );
    }
    }
</script>
<style type="text/css">
<!--
P{
font-size : 12px;
font-family : Verdana;
}
body {background-color: #FBFBEB; }
.form1 {
font-size : 11px;
font-family : Verdana;
background-color: #EDECDC
}
.container{
font-size: 11px;

```

```

        font-family: Verdana;
        background-color: #FBFBFB
    }
    .success{
        font-size: 14px;
        font-family: Verdana;
        font-style: italic;
        background-color: #EDECDC
    }

    .error {
        font: normal 9px arial;
        padding: 3px;
        margin: 3px;
        background-color: #ffc;
        border: 1px solid #c00;
    }
-->
</style>

</head>
    <body onload="load()">
<div class="container" id="container">
    <br>
<form action="" onsubmit="showAddress(this.address.value); return
false">
    <p>Παρακλώ επιλέξτε τη διεύθυνση του σημείου που θέλετε να
εισάγεται.
    <br>
    <p>Διεύθυνση: <input type="text" size="60" name="address"
value="Τρίπολη, Ελλάδα"> <input type="submit" value="submit"
class="subm" />
    </p>
</form>
    <div class="form1">

        <fieldset>
            <legend style="font-size : 12px;font-family :
Verdana;">Εισαγωγή νέου σημείου</legend>
            <table>
                <form id="submit" action="./upload.php" method="post"
enctype="multipart/form-data">
                    <tr><td><label for="category">Κατηγορία :</label></td>
                    <td><select id="category" name="category">
                        <option value="mouseia">Μουσεία</option>

```

```

        <option value="arxaia">Αρχαιολογικές περιοχές</option>
        <option value="eklisies">Εκκλησίες</option>
        <option value="plateies">Πλατείες</option>
        <option value="restaurant">Εσπιατόρια</option>
        <option value="coffe">Καφετέριες</option>
        <option value="hotel">Ξενοδοχεία</option>
    </select></td>
    <td> <label for="name">'Όνομα :</label></td>
    <td><input id="name" class="required" name="name" size="20"
type="text"></td></tr>
    <tr><td> <label for="address">Διεύθυνση :</label> </td>
    <td><input id="address" class="text" name="address" size="20"
type="text"></td>
    <td><label for="lat1">Lat :</label></td>
    <td> <input id="lat1" class="required" name="lat1" size="20"
type="text"> </td></tr>
    <tr><td> <label for="telephone">Τηλέφωνο :</label> </td>
    <td><input id="telephone" class="text" name="telephone" size="20"
type="text"> </td>
    <td> <label for="lng1">Longitude :</label> </td>
    <td> <input id="lng1" class="required" name="lng1" size="20"
type="text"> </td></tr>

    <tr> <td> <label for="email">E-mail :</label> </td>
    <td> <input id="email" class="text" name="email" size="20"
type="text"> </td>

    <td> <label for="file">Φωτογραφία :</label> </td>
    <td> <input type="file" id="file" name="userfile" value="no-
photo"></td></tr>
</table>
<p align="left">

Παρακαλώ αντιγράψτε τις τιμές των Latitude & Longitude <br>στα
αντίστοιχα πεδία της φόρμας.
<table bgcolor="#FFFFCC" width="300">
<tr>
<td width="100"><b><label for="lat">Lat</label></b></td>
<td id="lat" style="font-size : 14px;font-family : Verdana;font-weight
: bold;color : blue;" name="lat"></td>
</tr>
<tr>
<td width="100"><b>Longitude</b></td>
<td id="lng" style="font-size : 14px;font-family : Verdana;font-weight
: bold;color : blue;" name="lng"></td>

</tr>
</table><br>

<button> Add Data </button>

```

```
</form>  
</fieldset>  
<div id="map" style="width: 640px; height: 380px;"><br/>  
</div>  
</body>
```

registration.php

```
<head>

<style type="text/css">
<!--
P{
  font-size : 14px;
  font-family : Verdana;
}
body {background-color:#F1EFE4; }
.form1 {
  font-size : 11px;
  font-family : Verdana;
background-color:#EDECDC
}
.container{
  font-size: 11px;
  font-family: Verdana;
  background-color: #F1EFE4;

</style>

</head>
<body>

<br>
Παρακαλώ συμπληρώστε τα στοιχεία σας στη παρακάτω φόρμα.
<br>

<table>

<form          name="registration_form"          method="post"
action="<?=$_SERVER['php_self']?>" >
<tr><td>'Όνομα :</td><td> <input
name="name"></td></tr>
<tr><td> Επώνυμο : </td><td> <input type="text"
name="lastname"></td></tr>
<tr><td>          Email      : </td><td> <input type="text"
name="email"></td></tr>
<tr><td>          Username: </td><td> <input type="text"
name="username"></td></tr>
<tr><td>          Password: </td><td> <input type="password"
name="password"></td></tr>
<tr><td>          Confirm Password:</td><td><input type="password"
name="password_confirmation"></td></tr>
```



```

</table>

        <input type="submit" value="Register">
    </form><br>
    <p>
<?php

    $con = mysql_connect("localhost","root","");
if (!$con)
    {
    die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("map", $con);
$password=$_POST[password];
$password_conf=$_POST[password_confirmation];
$name=$_POST[name];
$email=$_POST[email];
$username=$_POST[username];
$lastname=$_POST[lastname];

if (strlen($name)==0) {
die ('<br><p style="color:red;">Παρακαλώ εισάγετε το όνομα σας. ');
}
if (strlen($lastname)==0) {
die ('<br><p style="color:red;">Παρακαλώ εισάγετε το επώνυμό σας. ');
}
if (strlen($email)==0) {
die ('<br><p style="color:red;">Παρακαλώ εισάγετε μια διεύθυνση
ηλεκτρονικού ταχυδρομείου. ');
}
if (strlen($username)==0) {
die ('<br><p style="color:red;">Παρακαλώ εισάγετε το username . ');
}
if (!$password==$password_conf) {
die('<br><p style="color:red;">Παρακαλώ εισάγετε ξανά το password. ');
}

$sql="INSERT INTO members (name, lastname, email, username,
password)
VALUES
('$ _POST[name]',          '$ _POST[lastname]',          '$ _POST[email]',
'$ _POST[username]', '$ _POST[password]' )";

if (!mysql_query($sql,$con))
    {

```

```
die('Error: ' . mysql_error());
}
echo "<br><p style=color:blue;>Η εισαγωγή των στοιχείων ήταν
επιτυχής.<br>";

?><FORM><INPUT TYPE="BUTTON" VALUE="Επιστροφή"
ONCLICK="history.go(-2);return false;"></FORM>
<?

mysql_close($con);

?>
</body>
```

view.php

```
<?
session_start();
if(!session_is_registered(myusername)){
header("location:index.php");
//echo "Θα πρέπει να κάνετε login ";
//leitoyrgei mono me anakateythinsi kai oxi ma minima
}
?>?
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">
<!--
P{
font-size : 14px;
font-family : Verdana;
}
body {background-color:#FBFBEB;}
.form {
font-size : 11px;
font-family : Verdana;
background-color:#EDECDC
}
.container{
font-size: 11px;
font-family: Verdana;
background-color: #FBFBEB
}

</style>
<?php
include 'header.php';
$host = "localhost";
$username = "root";
$password = "";
$databse = "map";
$rec_limit = 5; //arithmos egrafon ana selida
//display user

echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<a
href=Logout.php>&nbsp;&nbsp; Logout?</a><br><br>";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($databse, $connection) or die("MySQL Error");
/**paging selidas
//metrame arithmo egrafov toy pinaka
$sql = "SELECT * FROM markers ";
$result1=mysql_query($sql);
```

```

// Mysql_num_row is counting table row
$rec_count=mysql_num_rows($result1);

if( isset($_GET{'page'}) ) )
{
    $page = $_GET{'page'} + 1;
    $offset = $rec_limit * $page ;
}
else
{
    $page = 0;
    $offset = 0;
}
$left_rec = $rec_count - ($page * $rec_limit);

//to query poy emfanizei dedomena

$sql = mysql_query("select * from markers WHERE
user='$_SESSION[myusername]' limit $offset, $rec_limit");

?>
<html><body>
<form action="delete.php" method="POST">
<?php

echo "<table border=0>";
echo
"<tr><b><th></th><th>ΚΑΤΗΓΟΡΙΑ</th><th>ΟΝΟΜΑ</th><th>ΔΙΕΥ
ΘΥΝΣΗ</th><th>ΤΗΛΕΦΩΝΟ</th><th>EMAIL</th><th>ΙΑΤ</th><th>
ΛΝΓ</th><b></tr> ";
while ($result = mysql_fetch_array($sql))
{
echo " <tr><td><input type=checkbox name=del[] id=del
value=$result[id]></td><td> $result[category]</td><td>$result[name]
</td><td>
$result[address]</td><td>$result[telephone]</td><td>$result[email]</
td><td>$result[lat]</td><td>$result[lng]</td></tr>";
}
echo "</table>";
if( $page > 0 )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a> |";
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $page == 0 ){

```

```
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $left_rec < $rec_limit )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a>";
}

?>
<br>
<br>

<input type="submit" name="justdel" value="Delete !!" id="justdel">
</form><?php
mysql_close($connection);
include 'footer.php';
?>
</body></html>
```

delete.php

```
<?php
/* delete.php */
$host = "localhost";
$username = "root";
$password = "";
$databse = "map";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($databse, $connection) or die("MySQL Error");
$id = $_POST[del];
$count = count($id); //counting how many rows (from checkbox) to
delete
if($_POST['justdel'])
{
for ($i=0; $i<$count; $i++)
{
$sql = mysql_query("delete from markers where id=$id[$i]");
}
if ($sql)
{
echo "<meta http-equiv=\\"refresh\\" content=\\"0;URL=view.php\\">";
}
}
mysql_close();
?>
```

updateview.php.php

```
<?
session_start();
if(!session_is_registered(myusername)){
header("location:index.php");
}
?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">
<!--
P{
    font-size : 14px;
    font-family : Verdana;
}
body {background-color:#FBFBEB;}
.form1 {
    font-size : 11px;
    font-family : Verdana;
background-color:#EDECDC
}
.container{
    font-size: 11px;
    font-family: Verdana;
    background-color: #FBFBEB
}
</style>
<script type="text/javascript">
    //redirect function gia close window
function redirect(linkid)
{
opener.location.href=linkid
window.close()
}
</script>

<?php
/* view.php */
include 'header.php'?>
<br>
<br>
<br>
<?php
$host = "localhost";
$username = "root";
$password = "";
$databse = "map";
```

```

$rec_limit = 5; //arithmos egrafon ana selida
//emfanisi user sti selida
echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<a href=Logout.php>&nbsp;&nbsp; Logout?</a><br><br>";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($database, $connection) or die("MySQL Error");
//metrame arithmo egrafov toy pinaka

$sql = "SELECT * FROM markers";
$result1=mysql_query($sql);

// Mysql_num_row is counting table row
$rec_count=mysql_num_rows($result1);

if( isset($_GET{'page'}) ) )
{
    $page = $_GET{'page'} + 1;
    $offset = $rec_limit * $page ;
}
else
{
    $page = 0;
    $offset = 0;
}
$left_rec = $rec_count - ($page * $rec_limit);

//to query poy emfanizei dedomena

$sql = mysql_query("select * from markers WHERE
user='".$_SESSION[myusername]' limit $offset, $rec_limit");
?>
<html><body>
<form action="update.php.php" method="POST">
<?php

echo "<table border=0>";
echo
"<tr><b><th></th><th>ΚΑΤΗΓΟΡΙΑ</th><th>ΟΝΟΜΑ</th><th>ΔΙΕΥ
ΘΥΝΣΗ</th><th>ΤΗΛΕΦΩΝΟ</th><th>EMAIL</th><th>ΙΑΤ</th><th>
LNG</th><b></tr> ";
while ($result = mysql_fetch_array($sql))
{
echo "<tr><td><input type=checkbox checked=value name=del[] id=del
value=$result[id] style=display:none; ></td><td><input type=text
name=category[] size=10 id=category
value=$result[category]></td><td><input type=text name=name[]
size=10 id=name value=$result[name]></td><td><input type=text
name=address[] size=10 id=address

```



```

value=$result[address]></td><td><input type=text name=telephone[]
size=10 id=telephone value=$result[telephone]></td><td><input
type=text name=email[] size=10 id=email
value=$result[email]></td><td><input type=text name=lat[] size=10
id=lat value=$result[lat]> </td><td><input type=text name=lng[]
size=10 id=lng value=$result[lng]></td></tr>";
}

//prepei na exo ta pedia checked gia na doylepsi gia to logo ayto exo style
none; sto input ckeckbox

echo "</table>";
echo "<br>";
if( $page > 0 )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a> |";
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $page == 0 )
{
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $left_rec < $rec_limit )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a>";
}
?>
<br>
<br>

<input type="submit" name="update" value="update data" id="update">
</form>

</form><?php
mysql_close($connection);
?>
</body></html>

```

update.php.php

```
<?
session_start();
if(!session_is_registered(myusername)){
header("location:main_login.php");
}
?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">
<!--
P{
    font-size : 14px;
    font-family : Verdana;
}
body {background-color:#FBFBEB;}
.form1 {
    font-size : 11px;
    font-family : Verdana;
background-color:#EDECDC
}
.container{
    font-size: 11px;
    font-family: Verdana;
    background-color: #FBFBEB
}
</style>
<script type="text/javascript">
    //redirect function gia close window
function redirect(linkid)
{
opener.location.href=linkid
window.close()
}
</script>

<?php
/* update.php */
include 'header.php';

$host = "localhost";
$username = "root";
$password = "";
$databse = "map";
$conection = mysql_connect($host, $username, $password);
mysql_select_db($databse, $conection) or die("MySQL Error");
```

```

$id = $_POST[del];
$name=$_POST[name];
$category=$_POST[category];
$address=$_POST[address];
$telephone=$_POST[telephone];
$email=$_POST[email];
$lat=$_POST[lat];
$lng=$_POST[lng];

$count = count($id); //counting how many rows (from checkbox) to
delete

if($_POST['update'])
{

for ($i=0; $i<$count; $i++)

{

$sql = mysql_query("update markers set name='$name[$i]',
category='$category[$i]', telephone='$telephone[$i]',
address='$address[$i]', email='$email[$i]', lat='$lat[$i]', lng='$lng[$i]'
where id='$id[$i]'");
}
if ($sql)
{
//echo " <meta http-equiv=\\"refresh\\"
content=\\"0;URL=updateview.php.php\\">";
?>
<br><br>
<?php
echo " <br><br>Η επεξεργασία των δεδομένων
ολοκληρώθηκε.<br><br>";
echo " <a
href=http://localhost/mapfinal/deletephp/updateview.php.php>Επιστροφή
</a>";
}
}

mysql_close($connection);
?>

```

export.php

```
<?
session_start();
?>
<html>
  <head>

    <title>Εύρεση - κράτηση ξενοδοχείου</title>
  </head>
  <body>

    <p>          Παρακαλώ επιλέξτε ημερομηνία άφιξης και
αποχώρησης <br>
    για να εμφανιστούν τα διαθέσιμα δωμάτια. Κατόπιν <br>
    επιλέξτε το κατάλληλο link για να οδηγηθείτε στη φόρμα
<br>
    κράτησης.
  </br></p>

  <form action="<?php echo $_SERVER['PHP_SELF']; ?>" name="form"
method="post">
    <table>
      <tr>

      </tr>
      <tr>
        <td>
          Ημερομηνία Αφιξης:
        </td>
        <td>
          <input type="text" name="Sdate" /><input type="button"
name="cmdCal" value="Calendar"
onClick='javascript: window.open("calendar.php?form=form&field=Sdate",
"", "top=50,left=400,width=175,height=140,menubar=no,toolbar=no,scro
llbars=no,resizable=no,status=no"); return false;'>
        </td>
      </tr>
      <tr>
        <td>
          Ημερομηνία Αποχώρησης:
        </td>
        <td>
          <input type="text" name="edate" /><input type="button"
name="cmdCal" value="Calendar"
onClick='javascript: window.open("calendar.php?form=form&field=edate",
"", "top=50,left=400,width=175,height=140,menubar=no,toolbar=no,scro
llbars=no,resizable=no,status=no"); return false;'>
        </td>
      </tr>
    </table>
  </form>

```

```

        <tr><td>Εμφάνιση ανά :</td> <td><select name="cat"><option
value="hotel.hotelno">Ξενοδοχείο</option>
        <option value="room.type">Κατηγορία δωματίου</option>
        <option value="room.price">Τιμή</option>

                </td>
        </tr>
<tr>
        <td colspan="2" align="left">
                <input type="submit" value="Search"/>
        </td>
</tr>
</table>
</form>
<br>

<?php
$username="root";
$password="";
$database="hotel";

mysql_connect(localhost,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");

$room=$_POST['room'];
$Sdate=$_POST['Sdate'];
$edate=$_POST['edate'];
$cat=$_POST['cat'];

//query free rooms//
?>
<form action="booking.php" method="POST">
        <?
        $sql = mysql_query("select hotel.hotelname, hotel.city, room.roomid,
room.roomno, room.price, room.type from hotel, room where
room.roomid not in (select roomid from booking, hotel where (
datefrom <= '$Sdate' AND dateto >= '$edate') or (datefrom <= '$edate' AND
dateto >= '$edate') or (datefrom <= '$Sdate' AND dateto >= '$Sdate')
or (datefrom >= '$Sdate' AND dateto <= '$edate')) and
room.hotelno=hotel.hotelno order by $cat") or die(mysql_error());

        $_SESSION['Sdate']=$Sdate;
        $_SESSION['edate']=$edate;

```

```

echo "<table border='0' width='600'>";
echo
"<tr><th>Επιλογή</th><th>Περιοχή</th><th>Ξενοδοχείο</th><th>Ει
δος Δωματίου</th><th>Τιμή</th><th>Αριθμός Δωματίου</th></tr>";
// keeps getting the next row until there are no more to get
while($result = mysql_fetch_array( $sql )) {
    // Print out the contents of each row into a table

    echo    "<tr><td><input    type=radio    name=del[]    id=del
value=$result[roomid]></td>";
    echo "<td><center> $result[city]</td></center>";
    echo "<td><center>$result[hotelname] </td></center>";
    echo "<td><center> $result[type]</td></center>";
    echo "<td><center>$result[price] €</td></center>";
    echo "<td><center>$result[roomno]</td></center></tr>";

}
echo "</table>";
?>
<br>
<input type="submit" name="justdel" value="Booking" id="justdel">
</form>
<br>

</body>
</html>

```

booking.php

```
<?php session_start();
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
    <title>Κράτηση Δωματιου</title>
    <script type="text/javascript" src="http://code.jquery.com/jquery-
latest.js">
    </script>
    <script type="text/javascript" src="includes/js/jquery.form.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $('#myForm').ajaxForm({
          target: '#showdata',
          success: function(){
            $('#showdata').fadeIn('slow');
          }
        });
      });
    </script>
    <style type="text/css">
    <!--P {
      font-size: 14px;
      font-family: Verdana;
    }

    body {
      background-color: #F1EFE4;
    }

    .form1 {
      font-size: 11px;
      font-family: Verdana;
      background-color: #EDECDC
    }

    .container {
      font-size: 11px;
      font-family: Verdana;
      background-color: #F1EFE4;
    }

    </style>
  </head>
```

```

<body>
  <?php

$host = "localhost";
$username = "root";
$password = "";
$database = "hotel";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($database, $connection) or die("MySQL Error");
$roomid = $_POST[del];
$count = count($roomid); //counting how many rows (from checkbox) to
delete

if($_POST['justdel'])
{
for ($i=0; $i<$count; $i++)
{
echo "<table>";
$sql = mysql_query("select hotel.hotelname, hotel.city, room.roomid,
room.roomno, room.price, room.type from hotel, room where
roomid=$roomid[$i] and room.hotelno=hotel.hotelno limit 1");

if($sql){
while ($result = mysql_fetch_array($sql))
{

//echo "<tr><td> $result[hotelname]</td> <td> $result[roomno]</td>
<td> $result[type]</td> </tr>";
$_SESSION['hotelname']=$result[hotelname];
$_SESSION['roomno']=$result[roomno];
$_SESSION['type']=$result[type];
$_SESSION['city']=$result[city];
$_SESSION['roomid']=$result[roomid];
}

echo "</table>";

}
}
}

?>
<p>
  <br>
  Έχετε επιλέξει το δωμάτιο <b>
    <? echo $_SESSION['roomno'];?>
  </b>
  του ξενοδοχείου <b>
    <? echo $_SESSION['hotelname'];?>

```



```

</b>
<br>
Που βρίσκεται στην <b>
    <? echo $_SESSION['city'];?>
</b>
<br>
Με ημερομηνία Άφιξης :<b>
    <? echo $_SESSION['Sdate'];?>
</b>
και ημερομηνία Αποχώρησης :<b>
    <? echo $_SESSION['edate'];?>
</b>
<br>
Παρακαλώ συμπληρώστε τα υπόλοιπα στοιχεία για να ολοκληρωθεί
η κρατησή σας.
<br>
<br>
<table>
    <form          id="myForm"          action="bookingdata.php"
method="post">
    <tr>
        <td>
            Ονομα:
        </td>
        <td>
            <input type="text" name="name" />
        </td>
    </tr>
    <tr>
        <td>
            Επώνυμο:
        </td>
        <td>
            <input type="text" name="lastname" />
        </td>
    </tr>
    <tr>
        <td>
            Διεύθυνση:
        </td>
        <td>
            <input type="text" name="address" />
        </td>
    </tr>
    <tr>
        <td>
            Τηλέφωνο:
        </td>
        <td>

```

```

        <input type="text" name="phone" />
    </td>
</tr>
<tr>
    <td>
        Email :
    </td>
    <td>
        <input type="text" name="email" />
    </td>
</tr>

<tr>

    <td>
        <input type="submit" />

    </td>
</tr>
</form>
</td>
<td><br>
    <FORM>
        <INPUT    TYPE="BUTTON"    VALUE="Επιστροφή"
ONCLICK="history.go(-1);return false;">
    </FORM>
</td>
</tr>
</table>
<div id="showdata">
    <?// prepei na einai keno gia na min exo diples
egrafes include("bookingdata.php");?>
</div>
</body>
</html>

```

bookingdata.php

```
<?php session_start();

$host = "localhost";
$username = "root";
$password = "";
$databse = "hotel";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($databse, $connection) or die("MySQL Error");

//NOTE: MAKE SURE YOU DO YOUR OWN APPROPRIATE SERVERSIDE
ERROR CHECKING HERE!!!
if(!empty($_POST) && isset($_POST))
{
//make variables safe to insert gia apofigi diploegrafon sti basi apo refresh
tis selidas booking.php

$name = mysql_real_escape_string($_POST['name']);
$lastname = mysql_real_escape_string($_POST['lastname']);
$phone = mysql_real_escape_string($_POST['phone']);
$address = mysql_real_escape_string($_POST['address']);
$email = mysql_real_escape_string($_POST['email']);
$Sdate = mysql_real_escape_string($_SESSION[Sdate]);
$edate = mysql_real_escape_string($_SESSION[edate]);
$roomid = mysql_real_escape_string($_SESSION[roomid]);

if (strlen($name)==0) {
die ('<br><p style="color:red;">Παρακαλώ εισάγεται το όνομα σας.');
```

```

values ("', '$Sdate', '$edate', '$roomid', '$name', '$lastname', '$phone',
'$address', '$email' )";

$result = mysql_query($sql);
    if(!$result)
    {
        echo "Failed to insert record to booking table.";
    }
    else {
echo   "<br><p   style=color:blue;>Η κράτηση του δωματίου
ολοκληρώθηκε.<br><br>";
?>

<FORM><INPUT TYPE="BUTTON" VALUE="Επιστροφή"
ONCLICK="history.go(-1);return false;"></FORM>
<?
}
}
?>

```

bookingresult.php

```
<?
session_start();
if(!session_is_registered(myusername)){
header("location:index.php");

//leitoyrgei mono me anakateythinsi kai oxi ma minima
}
include 'header.php';

?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
    <title>Κράτηση Δωματίου</title>
    <script type="text/javascript" src="http://code.jquery.com/jquery-
latest.js">
    </script>
    <script type="text/javascript" src="includes/js/jquery.form.js">
    </script>
    <script type="text/javascript">
      $(document).ready(function(){
        $('#myForm').ajaxForm({
          target: '#showdata',
          success: function(){
            $('#showdata').fadeIn('slow');
          }
        });
      });
    </script>
    <style type="text/css">
    <!--P {
      font-size: 14px;
      font-family: Verdana;
    }

    body {
      background-color: #F1EFE4;
    }

    .form1 {
      font-size: 11px;
      font-family: Verdana;
      background-color: #EDECDC
    }
  </style>
</html>
```

```

.container {
    font-size: 11px;
    font-family: Verdana;
    background-color: #F1EFE4;

</style>
</head>
<body>

    <html>

<head> <title>Hotel export form</title>

</head>
<html>
<body>
<br>
<? echo "<b>Welkome user : </b>".$_SESSION['myusername']. "<a
href=Logout.php>&nbsp;&nbsp;&nbsp;Logout?</a><br><br>"; ?>

Στη σελίδα αυτή μπορείτε να δείτε τα δωμάτια που βρίσκονται σε
κράτηση.<br>
<form action="bookingresult.php" name="form" method="Post" >
<table>

<tr><td>Ημερομηνία      Άφιξης:</td><td><input      type="text"
name="Sdate"           /><input      type="button"   name="cmdCal"
value="Calendar"
onClick='javascript: window.open("calendar.php?form=form&field=Sdate",
"", "top=50,left=400,width=175,height=140,menubar=no,toolbar=no,scro
llbars=no,resizable=no,status=no"); return false;'></td></tr>
<tr><td>Ημερομηνία      Αποχώρησης:</td><td><input      type="text"
name="edate"           /><input type="button" name="cmdCal" value="Calendar"
onClick='javascript: window.open("calendar.php?form=form&field=edate",
"", "top=50,left=400,width=175,height=140,menubar=no,toolbar=no,scro
llbars=no,resizable=no,status=no"); return false;'></td></tr>
<tr><td>Εμφάνιση ανά  :</td> <td><select name="cat"><option
value="hotel.hotelno">Ξενοδοχείο</option>
<option value="room.type">Κατηγορία δωματιού</option>
<option value="room.price">Τιμή</option>

<tr><td colspan="2" align="left"><input type="submit" /></td></tr>
</table>
</form>

<?php

```

```

$host = "localhost";
$username = "root";
$password = "";
$dbase = "hotel";
$connection = mysql_connect($host, $username, $password);
mysql_select_db($dbase, $connection) or die("MySQL Error");

    $Sdate=$_POST['Sdate'];
    $edate=$_POST['edate'];
    $cat=$_POST['cat'];

$rec_limit = 5; //arithmos egrafon ana selida

$sql = "SELECT * FROM booking ";
$result1=mysql_query($sql);

// Mysql_num_row is counting table row
$rec_count=mysql_num_rows($result1);

if( isset($_GET{'page'}) ) )
{
    $page = $_GET{'page'} + 1;
    $offset = $rec_limit * $page ;
}
else
{
    $page = 0;
    $offset = 0;
}
$left_rec = $rec_count - ($page * $rec_limit);

$sql = mysql_query("select hotelname, roomno, type, datefrom, dateto,
name, lastname, address, telephone, email, bookid from hotel, room,
booking where bookid in (select bookid from booking where ( (datefrom
<= '$Sdate' AND dateto >= '$edate') or (datefrom <='$edate' AND
dateto >= '$edate') or (datefrom <= '$Sdate' AND dateto >= '$Sdate')
or (datefrom >= '$Sdate' AND dateto <= '$edate')) and
booking.roomid=room.roomid and hotel.hotelno=room.hotelno order by
'$cat' limit $offset, $rec_limit") or die(mysql_error());

```

```

echo "<table border='0' width='100%'>";
echo "<tr><th>Ξενοδοχείο</th><th>Αριθμός Δωματίου</th><th>Είδος
Δωματίου</th><th>Ημερομηνία Αφιξης</th><th>Ημερομηνία
Αποχώρησης</th><th>Όνομα Πελάτη</th><th>Επώνυμο
Πελάτη</th><th>Διεύθυνση</th><th>Τηλέφωνο</th><th>Email</th>
</tr>";
// keeps getting the next row until there are no more to get
while($result = mysql_fetch_array( $sql )) {
    // Print out the contents of each row into a table

echo "<tr><td><center>$result[hotelname]</td></center>";
    echo "<td><center>$result[roomno]</td></center>";
    echo "<td><center>$result[type]</td></center>";
    echo "<td><center>$result[datefrom]</td></center>";
    echo "<td><center>$result[dateto]</td></center>";
    echo "<td><center>$result[name]</td></center>";
    echo "<td><center>$result[lastname]</td></center>";
    echo "<td><center>$result[address]</td></center>";
    echo "<td><center>$result[telephone]</td></center>";
    echo "<td><center>$result[email]</td></center></tr>";
}
echo "</table>";

    if( $page > 0 )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a> |";
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $page == 0 )
{
    echo "<a href=\"$_PHP_SELF?page=$page\">Next 5 Records</a>";
}
else if( $left_rec < $rec_limit )
{
    $last = $page - 2;
    echo "<a href=\"$_PHP_SELF?page=$last\">Last 5 Records</a>";
}
}

include 'footer.php';
?></body></html>

```