



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Δημιουργία ιστότοπου με την χρήση και παραμετροποίηση εργαλείων διαχείρισης περιεχομένου
Όνοματεπώνυμο Φοιτητή	Νικόλαος Ζαχαριάδης
Πατρώνυμο	Βασίλειος
Αριθμός Μητρώου	ΜΠΠΛ/06022
Επιβλέπων	Δημήτριος Βέργαδος, Λέκτορας

Ημερομηνία Παράδοσης **ΙΟΥΛΙΟΣ 2011**



Τριμελής Εξεταστική Επιτροπή

Χ. Δουληγέρης
Καθηγητής

Δ. Βέργαδος
Λέκτορας

Π. Κοτζανικολάου
Λέκτορας

Ευχαριστίες

Με την ολοκλήρωση της μεταπτυχιακής μου διατριβής, η οποία υλοποιήθηκε στο Πανεπιστήμιο Πειραιώς, θα ήθελα να ευχαριστήσω τους ανθρώπους οι οποίοι συνέβαλαν στην περάτωση της με τις γνώσεις τους και τις συμβουλές τους.

Συγκεκριμένα οφείλω να ευχαριστήσω τον επιβλέποντα μου Λέκτορα Δημήτριο Βέργαδο για την εμπιστοσύνη που μου έδειξε αναλαμβάνοντας την εποπτεία της διατριβής μου, καθώς επίσης και γιατί ήταν διαλλακτικός στην επιλογή ενός ενδιαφέροντος και απολύτως σύγχρονου θέματος. Επιπλέον τον ευχαριστώ για τις πολύτιμες συμβουλές και κατευθύνσεις που μου έδωσε, και γενικότερα την καθοδήγηση που μου παρείχε.

Ακόμη θα ήθελα να πω ένα μεγάλο ευχαριστώ σε όλους του καθηγητές που είχα στο μεταπτυχιακό μου πρόγραμμα, διότι με βοήθησαν να αποκομίσω όσο το δυνατόν περισσότερα πράγματα σχετικά με την επιστήμη μου αλλά και γενικότερα για τη ζωή μου.

Τέλος ευχαριστώ όλους εκείνους που ήταν δίπλα μου σε όλη αυτή την προσπάθεια παρέχοντας απεριόριστη ψυχολογική υποστήριξη και κατανόηση.

Περίληψη

Ο σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η εφαρμογή και υλοποίηση μιας ιστοσελίδας με την χρήση και παραμετροποίηση εργαλείων διαχείρισης περιεχομένου. Συγκεκριμένα αναφέρεται στη δημιουργία ιστοσελίδας εύρεσης ξενοδοχείων στην Ελλάδα χρησιμοποιώντας το ανοικτού περιεχομένου σύστημα διαχείρισης, Joomla. Στόχος είναι να εμπλουτιστεί η εφαρμογή με τους απαραίτητους μηχανισμούς, οι οποίοι θα δίνουν την δυνατότητα στους εγγεγραμμένους χρήστες να παραμετροποιούν την σελίδα σύμφωνα με τις προτιμήσεις τους, ούτως ώστε να προβάλλονται οι δυνατότητες της με τον καλύτερο τρόπο σε ένα εύχρηστο περιβάλλον .

Στο θεωρητικό μέρος της εργασίας μελετήθηκαν θέματα που αφορούν τα συστήματα διαχείρισης περιεχομένου γενικότερα. Έγινε αναφορά σε ορισμούς, στον τρόπο λειτουργίας, στα χαρακτηριστικά και στα πλεονεκτήματά τους. Επιπλέον αναφέρθηκαν και τα εργαλεία που χρειάζονται ώστε να γίνει η εγκατάσταση του Joomla στο οποίο βασίζεται η ανάπτυξη της εφαρμογής. Στο τεχνικό κομμάτι για την κατασκευή της σελίδας με Joomla (CMS) έγινε η ενσωμάτωση ανεξάρτητων εφαρμογών σε αυτήν, των οποίων η υλοποίηση βασίζεται στον συνδυασμό των τεχνολογιών php, mysql και XML, που δίνουν πρόσθετους και εναλλακτικούς τρόπους εμφάνισης του περιεχομένου της ιστοσελίδας στους χρήστες.

Τέλος θα αναφερθούμε αναλυτικά στην υλοποίηση ενός ολοκληρωμένου component με το οποίο οι χρήστες της ιστοσελίδας θα μπορούν με πολύ εύκολο τρόπο να γράφουν κριτικές για τα εστιατόρια των ξενοδοχείων όπου θα εμφανίζεται η τοποθεσία του ξενοδοχείου σε Google Maps και θα παρέχεται και η δυνατότητα εγγραφής σχολίων (comments). Ο κώδικας για το συγκεκριμένο component βασίζεται στην php, mysql και η γεωκωδικοποίηση σε javascript.

Abstract

The purpose of this postgraduate thesis is the development of a web page using a content management system (cms). It refers to a web site, for finding hotels in Greece using the open source content management system, Joomla. The aim is to enrich the application with the necessary mechanisms that will give registered users the possibility to customize the page according to their preferences, in order to have a good looking interface which displays with the best way the capabilities of the webpage.

In the theoretical part of the project they were studied issues related to content management systems in general. It was made a reference to definitions, characteristics and advantages of the cms. Additionally were reported the tools were needed to install the Joomla in which the development of this application is based. On the technical side, the construction of the page with Joomla (CMS) was made with the integration of independent applications to it. These independent applications are based on a combination of technologies php, mysql and XML, which provide additional and alternative ways to display the contents of the web page to all users.

Finally we will explain in detail the development of an entire component with which users of the site could very easily write reviews for hotel's restaurants. The location of these will be displayed on Google Maps and will also have the chance to insert comments. The code for this component is based on php, mysql and the geocoding in javascript.

Περιεχόμενα

Ευχαριστίες.....	3
Περίληψη	4
Abstract	5
1 Εισαγωγή.....	9
1.1 Τι είναι Γνώση.....	9
1.2 Δεδομένα.....	9
1.3 Πληροφορία.....	10
1.4 Γνώση	10
1.4.1 Η γνώση είναι “ακατάστατη”	11
1.4.2 Η γνώση είναι αυτό-οργανωμένη.....	11
1.4.3 Η γνώση αναζητά κοινότητες.....	11
1.4.4 Η γνώση ταξιδεύει με τη γλώσσα.....	11
1.4.5 Η γνώση είναι αφηρημένη	12
1.4.6 Πιο χαλαρά είναι καλύτερα	12
1.4.7 Η γνώση αλλάζει συνεχώς	12
1.4.8 Η γνώση δεν διογκώνεται για πάντα- πάντα κάτι τερματίζεται ή χάνεται	12
1.4.9 Δεν υπάρχει αρχηγός.....	12
1.4.10 Δεν μπορούμε να επιβάλλουμε κανόνες και συστήματα	12
1.4.11 Δεν υπάρχει η τέλεια λύση.....	13
1.4.12 Το πώς ορίζουμε το “πρόβλημα” της γνώσης καθορίζει και το πώς θα το χειριστούμε	13
2 Συστήματα Διαχείρισης Περιεχομένου	14
2.1 Τι είναι σύστημα διαχείρισης περιεχομένου.	14
3 Κατηγορίες Συστημάτων Διαχείρισης Περιεχομένου με βάση την επεξεργασία.	16
3.1 Επεξεργασία Offline.....	16
3.2 Επεξεργασία Online.....	16
3.3 Υβριδικά Συστήματα	16
4 Κατηγορίες Συστημάτων Διαχείρισης Περιεχομένου με βάση αν είναι ανοικτού η κλειστού κώδικα.	17
4.1 CMS ανοικτού κώδικα.....	17
4.2 CMS κλειστού κώδικα.	17
5 Εισαγωγή στο Σύστημα Joomla.....	18
5.1 Τι είναι το Joomla.....	18
5.2 Δομή ενός CMS	18
5.3 Διεπαφές Χρήστη και διαχειριστών.....	19
5.4 Πρότυπα.....	19
5.5 Επεκτάσεις	19
5.6 Ιστορική Αναδρομή	19
6 Εγκατάσταση του Joomla σε περιβάλλον Windows.....	21
6.1 Εγκατάσταση λογισμικού XAMPP	21
6.2 Apache HTTP Server.....	23

6.3	MySQL	23
6.4	PHP.....	23
6.5	Perl.....	24
6.6	Εγκατάσταση Joomla!.....	24
6.7	Επιλογή γλώσσας (Language Choice)	25
6.8	Έλεγχος προ εγκατάστασης (Pre-installation Check).....	25
6.9	Άδεια (Licence).....	26
6.10	Ρύθμιση της βάσης δεδομένων (Database Configuration)	27
6.11	Ρύθμιση FTP (FTP Configuration).....	28
6.12	Κύρια Ρύθμιση (Main Configuration).....	29
6.12.1	Εγκατάσταση προκαθορισμένων δεδομένων δείγματος.....	30
6.12.2	Migration Tools	30
6.13	Τέλος (Finish)	31
7	Εφαρμογή: Υλοποίηση ιστότοπου για εύρεση ξενοδοχείων με τη βοήθεια του Joomla.	33
7.1	Γενική περιγραφή εφαρμογής.....	33
7.2	Ιδιαίτερα χαρακτηριστικά του Joomla.....	33
7.3	Εγκατάσταση και παραμετροποίηση προτύπου (template).....	34
8	Δημιουργία Περιεχομένου.....	36
8.1	Διαχείριση ενοτήτων	36
8.2	Διαχείριση κατηγοριών	37
8.3	Διαχείριση άρθρων.	37
9	Διαχείριση μενού.....	40
9.1	Δημιουργία και παραμετροποίηση μενού.....	40
9.2	Δημιουργία συνδέσμων.....	43
9.3	Μέρος του κώδικα για την εμφάνιση των μενού.....	43
9.4	Διαχείριση χρηστών και πρόσβασης.	46
10	Επεκτάσεις του Joomla που έχουν χρησιμοποιηθεί για τον συγκεκριμένο ιστότοπο.	49
10.1	Εγκατάσταση και ανάλυση των επεκτάσεων.....	49
10.1.1	JComments: component.....	50
10.1.2	Content map: module.....	51
10.1.3	FrontPage Slideshow: component.....	51
10.1.4	Content Map: Plugin.....	52
10.1.5	Sigplus Image Gallery Plus: plugin.	53
11	RSS Feed (Really Simple Syndication)	54
11.1	Τι ακριβώς είναι το RSS Feed	54
11.2	Πως λειτουργεί το RSS Feed στο Joomla.....	55
11.2.1	Παραμετροποίηση του αρχικού module.....	57
11.2.2	Feed Entries	58
12	Δημιουργία κώδικα για την υλοποίηση ενός εξαρτήματος (Component).....	59
12.1	Η δομή ενός component.....	59
12.2	Εγγραφή του component στη βάση δεδομένων.	60
12.3	Δημιουργία μπάρας εργαλείων.....	61
13	Ανάπτυξη κώδικα για το παρασκήνιο (back end) του component.	62
13.1	Δημιουργία πίνακα βάσης δεδομένων	62

13.2	Δημιουργία κλάσης πίνακα.....	63
13.3	Δημιουργία της φόρμας του back end για την εισαγωγή κριτικών.....	64
13.4	Επεξεργασία των δεδομένων.....	70
13.5	Οθόνη για την παρουσίαση και την επεξεργασία της λίστας με τις κριτικές.....	71
13.6	Κατάργηση των δεδομένων μας.....	75
13.7	Δημιουργία και διαχείριση σχολίων.....	76
14	Ανάπτυξη κώδικα για το προσκήνιο (front end) του component.....	82
14.1	Κώδικας για τον τρόπο που θα παρουσιάζονται τα σχόλια.....	86
14.2	Εμφάνιση σχολίων.....	89
14.3	Κώδικας Γεωκωδικοποίησης.....	91
15	Προετοιμασία και εγκατάσταση του component.....	94
16	Δημιουργία Module για τις κριτικές.....	96
16.1	Εγγραφή του module στη βάση μας.....	96
16.2	Κώδικας για το module.....	97
16.3	Προετοιμασία και εγκατάσταση του module.....	100
17	Μελλοντικές επεκτάσεις.....	101
18	Συμπεράσματα.....	102
19	Βιβλιογραφία & Πηγές στο Διαδίκτυο.....	103

1 Εισαγωγή

Στο κεφάλαιο αυτό θα εξηγήσουμε με σύντομο και κατανοητό τρόπο τις έννοιες της γνώσης και της διαχείρισης της γνώσης. Για να μπορέσουμε να καταλάβουμε την έννοια της διαχείρισης γνώσης θα πρέπει να αντιληφθούμε το τι είναι η γνώση και να καταλάβουμε την διαφορά της από τα δεδομένα και τις πληροφορίες.

1.1 Τι είναι Γνώση

Παρακάτω αναφέρονται κάποιοι ορισμοί για την γνώση:

“Η γνώση έχει γίνει η βασική πηγή, για την στρατιωτική δύναμη όπως και για την οικονομική δύναμη ενός έθνους... είναι ριζικά διαφορετική από τις παραδοσιακές βασικές οικονομικές πηγές – γη, έργο, και ακόμα κεφάλαιο.... χρειαζόμαστε συστηματική δουλειά στην ποιότητα και την παραγωγικότητα της γνώσης... το μέγεθος της παραγωγικότητας, αν όχι η επιβίωση, κάθε οργανισμού στην κοινωνία της γνώσης συνεχώς θα εξαρτάται όλο και περισσότερο από αυτούς τους δύο παράγοντες” [Drucker,1994]

“Σε καιρούς δυνατού ανταγωνισμού και συνεχώς μεταβαλλόμενα περιβάλλοντα, η γνώση είναι η βάση για την οικονομική επιτυχία” (Davenport & Prusak, 1998; Drucker, 1993; Nonaka & Takeuchi, 1995; Schreyögg & Geiger, 2003)

“Η γνώση είναι ένα ρευστό μίγμα της πλακισωμένης εμπειρίας, των βασισμένων στα συμφραζόμενα πληροφοριών, της διορατικότητας και της διαίσθησης που παρέχει ένα περιβάλλον και ένα πλαίσιο για νέες εμπειρίες και πληροφορίες. Η γνώση δημιουργείται και εφαρμόζεται από τους ανθρώπους. Στις οργανώσεις, η γνώση είναι ενσωματωμένη στα έγγραφα, τις αποθήκες αλλά και στις οργανωτικές ρουτίνες, τις διαδικασίες, τις πρακτικές, και τους κανόνες.” [1]

Η γνώση είναι κάτι περισσότερο από το να γνωρίζουμε απλά κάτι. Ξέρουμε πολλά, αλλά δεν αποτελούν όλα γνώση. Κάνουμε το διαχωρισμό μεταξύ δεδομένων, πληροφορίας και γνώσης.

ΓΝΩΣΗ

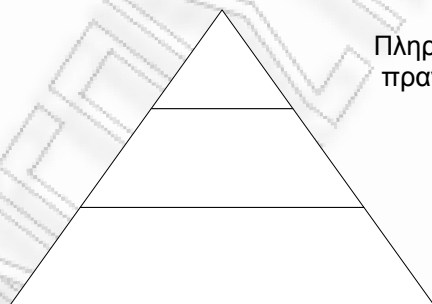
Πληροφορίες, που χρησιμοποιούνται για να πραγματοποιηθούν ή να δημιουργηθούν κάτι.

ΠΛΗΡΟΦΟΡΙΑ

Δεδομένα δομημένα γύρω από ένα θέμα.

ΔΕΔΟΜΕΝΟ

Απλά δεδομένα, χωρίς δομή.



Στις επόμενες τρεις παραγράφους εξηγούνται περισσότερο οι έννοιες: δεδομένο, πληροφορία και γνώση.

1.2 Δεδομένα

Στο πρώτο και βασικότερο επίπεδο βρίσκουμε τα "δεδομένα". Τα δεδομένα είναι ένα σύνολο από διακριτά, αντικειμενικά γεγονότα σχετικά με κάποια ενέργεια. Σε ένα οργανωτικό περιβάλλον, τα δεδομένα περιγράφονται συνήθως ως δομημένες εγγραφές ή συναλλαγές. Στους σύγχρονους οργανισμούς τα δεδομένα αποθηκεύονται συνήθως μέσα σε τεχνολογικά συστήματα.

Χαρακτηριστικό των δεδομένων είναι ότι δεν προσφέρουν κανένα στοιχείο σχετικά με την αιτία και το σκοπό της ενέργειας για την οποία αποτελούν δεδομένα. Δε μας λένε τίποτα για το χαρακτήρα του ατόμου ή των ατόμων που ενήργησαν με κάποιον τρόπο. Δεν εμπριέχεται νόημα στα δεδομένα. Τα δεδομένα απλά περιγράφουν ένα κομμάτι μιας ενέργειας. Είναι το σύνολο των μη δομημένων γεγονότων και διαγραμμάτων που γνωρίζουμε. Τα έχουμε κατά κάποιον τρόπο στη διάθεσή μας. Δεν μπορούν να εξαχθούν συμπεράσματα με βάση αυτά. Στην ουσία, τα δεδομένα είναι η πρώτη ύλη για τη δημιουργία πληροφοριών.

1.3 Πληροφορία

Πληροφορία είναι οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων. Επεξεργασία δεδομένων είναι η διαδικασία κατά την οποία ένας μηχανισμός δέχεται δεδομένα τα επεξεργάζεται σύμφωνα με ένα προκαθορισμένο τρόπο και αποδίδει πληροφορίες.

Η πληροφορία μπορεί να χαρακτηριστεί ως ένα μήνυμα, συνήθως με τη μορφή ενός εγγράφου ή μιας οπτικής ή ακουστικής επικοινωνίας. Όπως και κάθε μήνυμα, η πληροφορία έχει έναν αποστολέα και έναν παραλήπτη. Η πληροφορία έχει σαν σκοπό να αλλάξει τον τρόπο με τον οποίο ο παραλήπτης αντιλαμβάνεται κάτι, να επηρεάσει την κρίση και τη συμπεριφορά του.

Οι πληροφορίες διακινούνται μέσα στους οργανισμούς διαμέσου σκληρών και μαλακών δικτύων. Ένα σκληρό δίκτυο έχει μία ορατή και συγκεκριμένη υποδομή: καλώδια, δορυφορικά πιάτα, ταχυδρομικά γραφεία, διευθύνσεις, κουτιά ταχυδρομείου. Στα μηνύματα που διακινούνται μέσω ενός τέτοιου δικτύου περιλαμβάνονται τα e-mail, παραδοσιακές επιστολές, πακέτα υπηρεσιών και δικτυακές μεταδόσεις. Ένα μαλακό δίκτυο είναι λιγότερο επίσημο και ορατό. Η παράδοση ενός σημειώματος ή ενός αποκόμματος κάποιου άρθρου από έναν υπάλληλο μιας εταιρείας σε κάποιον άλλο αποτελεί παράδειγμα μεταφοράς πληροφοριών μέσω ενός μαλακού δικτύου.

Σε αντίθεση με τα δεδομένα, η πληροφορία εμπριέχει νόημα. Είναι οργανωμένη με βάση κάποιον σκοπό. Τα δεδομένα μετατρέπονται σε πληροφορία, όταν προστίθεται σε αυτά νόημα. Υπάρχουν διάφορες μέθοδοι για τη μετατροπή των δεδομένων σε πληροφορία:

- Καθορισμός του σκοπού για τον οποίο συγκεντρώθηκαν τα δεδομένα
- Κατηγοριοποίηση των δεδομένων με βάση τα στοιχεία που τα συνθέτουν
- Ανάλυση των δεδομένων με μαθηματικές ή στατιστικές μεθόδους
- Αναζήτηση για πιθανά λάθη στα δεδομένα και διόρθωσή τους
- Σύνοψη των δεδομένων σε μια πιο σαφή και περιεκτική μορφή

Εν κατακλείδι, μπορεί κανείς να πει ότι η πληροφορία είναι δεδομένα στα οποία έχει προστεθεί νόημα με την έννοια του περιεχομένου.

1.4 Γνώση

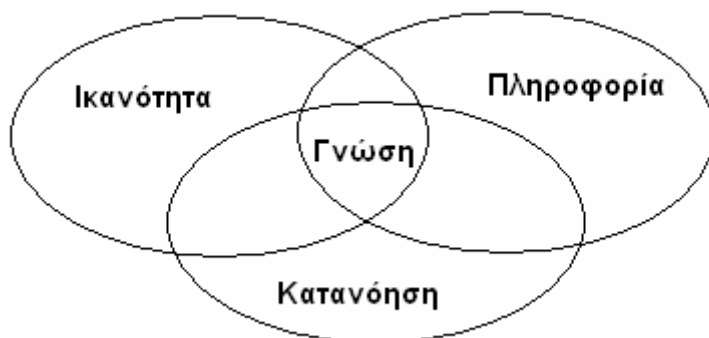
Το πρώτο που χρειάζεται κανείς να ξέρει για τη γνώση ότι δεν είναι κάτι συγκεκριμένο και απλό. Η γνώση προκύπτει από την πληροφορία, όπως η πληροφορία προκύπτει από τα δεδομένα.

Αυτό γίνεται μέσα από διαδικασίες όπως:

Σύγκριση ανάμεσα στις πληροφορίες που υπάρχουν για διάφορα γεγονότα
Ανάλυση των συνεπειών που έχουν οι πληροφορίες στη λήψη αποφάσεων
Σύνδεση ανάμεσα στα διάφορα κομμάτια γνώσης που υπάρχουν σε ένα μέρος
Συζήτηση των διάφορων πληροφοριών με άλλους ανθρώπους

Όλες αυτές οι δραστηριότητες λαμβάνουν χώρα μέσα στο μυαλό των ανθρώπων. Ενώ τα δεδομένα βρίσκονται μέσα σε εγγραφές και συναλλαγές, και οι πληροφορίες σε μηνύματα, η

γνώση προέρχεται από μεμονωμένα άτομα ή ομάδες ατόμων, ή ακόμα και μέσα από οργανωσιακές διαδικασίες. Μεταφέρεται διαμέσου δομημένων μέσων όπως βιβλία και έγγραφα, και μέσα από προσωπική επαφή, είτε πρόκειται για απλή συζήτηση, είτε για διδασκαλία. Με άλλα λόγια, η γνώση είναι πληροφορίες συνδυασμένες με την κατανόηση και την ικανότητα και βρίσκεται μέσα στο μυαλό των ανθρώπων.



Εικόνα: Αλληλεπίδραση πληροφορίας, γνώσης, ικανότητας και κατανόησης.

Παρακάτω παρατίθενται ορισμένες βασικές αρχές σχετικές με τη γνώση. Η κατανόησή τους είναι ιδιαίτερα σημαντική, προκειμένου να μπορέσουν τα άτομα και οι επιχειρήσεις γενικότερα να δημιουργήσουν, να διαμοιράσουν και να διαχειριστούν τη γνώση με τρόπο αποτελεσματικό. Οι αρχές αυτές είναι γνωστές ως “Τα 12 Αξιώματα της Γνώσης” και είναι τα εξής:[2]

1.4.1 Η γνώση είναι “ακατάστατη”

Στους οργανισμούς κάθε όψη της γνώσης είναι συνδεδεμένη με τα πάντα. Δεν μπορούμε να απομονώσουμε το λεγόμενο “στοιχείο της γνώσης”. Η οργανωσιακή γνώση συνδέεται με την κουλτούρα, τις δομές και την τεχνολογία του οργανισμού, καθώς επίσης και με τις ιδιαιτερότητες των ατόμων που τον απαρτίζουν. Έτσι, κάθε προσπάθεια να απομονώσουμε παράγοντες γνώσης αποδεικνύεται μία ιδιαίτερα πολύπλοκη διαδικασία. Σε ό,τι αφορά τη γνώση δεν μπορούμε να επικεντρωθούμε σε ένα μόνο πράγμα. Όσο κι αν προσπαθήσουμε να απομονώσουμε τη γνώση, βρίσκουμε κάτι που συνδέεται άρρηκτα με αυτήν.

1.4.2 Η γνώση είναι αυτό-οργανωμένη

Σε έναν οργανισμό η γνώση δημιουργείται, διατηρείται, τερματίζεται και ανανεώνεται καθημερινά. Η γνώση οργανώνεται από μόνη της. Αυτό γύρω από το οποίο οργανώνεται είναι η ταυτότητα και ο σκοπός του οργανισμού.

1.4.3 Η γνώση αναζητά κοινότητες

Η γνώση τρέφεται από τις κοινότητες. Χαρακτηριστικό παράδειγμα αποτελεί η κοινότητα (ή κοινωνία πλέον) του Internet. Οι κοινότητες αυτές είναι πλέον τόσο ισχυρές που εμπλέκουν σε επικοινωνία ανθρώπους από όλο τον κόσμο.

1.4.4 Η γνώση ταξιδεύει με τη γλώσσα

Η γλώσσα είναι το λεκτικό αποτύπωμα των εμπειριών μας. Χωρίς τη γλώσσα δε θα μπορούσαμε να επικοινωνήσουμε και να μοιραστούμε τις εμπειρίες μας. Κάθε τύπος γνώσης ταξιδεύει με διαφορετική γλώσσα. Κάθε γλώσσα μας εισάγει σε ένα συγκεκριμένο κόσμο εμπειριών. Για παράδειγμα, η παραδοσιακή διαχείριση χρησιμοποιεί τη γλώσσα των

στατιστικών ελέγχων, των επιθεωρήσεων και των ισολογισμών. Κάποιος ο οποίος δε γνωρίζει αυτή τη γλώσσα δεν μπορεί να εισαχθεί στο χώρο της διαχείρισης. Επεκτείνοντας την οργανωσιακή γνώση πρέπει να επεκτείνουμε μαζί και τις γλώσσες που χρησιμοποιούμε για να περιγράψουμε τις επαγγελματικές μας εμπειρίες και απόψεις.

1.4.5 Η γνώση είναι αφηρημένη

Θα ήταν πολύ βολικό να μπορούσαμε να συλλάβουμε επακριβώς όλη τη γνώση σε κείμενα, πατέντες, διανοητικό κεφάλαιο, βιβλιοθήκες και βάσεις δεδομένων. Παρ' όλα αυτά, η υπερβολική τυποποίηση οδηγεί σε ακαμψία της γνώσης. Η γνώση χάνει πλέον την ιδιότητα της ευελιξίας και αυτό-οργάνωσης που έχει και έτσι δεν μπορεί να χρησιμοποιηθεί αποτελεσματικά για τη δημιουργία νέας γνώσης.

1.4.6 Πιο χαλαρά είναι καλύτερα

Τα προσαρμοστικά στα χαρακτηριστικά της γνώσης συστήματα φαίνονται ανοργάνωτα και χωρίς σταθερούς και αναμφισβήτητους κανόνες. Αλλά το ποσοστό επιβίωσης τέτοιων αντιφατικών και αποκεντρωμένων συστημάτων είναι μεγαλύτερο. Αυτό σημαίνει ότι υπάρχει ο κίνδυνος να ξοδέψουμε άσκοπα πόρους και ενέργεια προσπαθώντας να ελέγξουμε σε υπερβολικό βαθμό τις διαδικασίες της γνώσης.

1.4.7 Η γνώση αλλάζει συνεχώς

Δεν υπάρχει κάποια μόνιμη λύση στη διαχείριση γνώσης. Τα μοτίβα της γνώσης αλλάζουν συνεχώς. Η καλύτερη προσέγγιση και λύση κάθε φορά είναι αυτή που επιτρέπει στις υπάρχουσες διαδικασίες να συνεχίζονται και ταυτόχρονα αφήνει κάθε επιλογή ανοικτή. Η ευελιξία στην προσέγγιση και τη σκέψη είναι απαραίτητη. Υπάρχουν διαφορετικές προσεγγίσεις να δοκιμάσουμε. Στην πραγματικότητα, η διαρκώς συνεχιζόμενη συζήτηση για τη γνώση είναι πολύ πιο σημαντική από το σκεφτόμαστε για την καλύτερη μόνιμη λύση.

1.4.8 Η γνώση δεν διογκώνεται για πάντα- πάντα κάτι τερματίζεται ή χάνεται

Πολλοί άνθρωποι στον τομέα της διαχείρισης γνώσης πιστεύουν ακράδαντα ότι η γνώση διογκώνεται συνεχώς. Κάτι τέτοιο δεν ισχύει στην πραγματικότητα. Όπως ακριβώς και στη φύση, η συνεχής διόγκωση δεν υφίσταται στη γνώση. Υπάρχει μεγάλη διαφορά ανάμεσα στο να αναπτύσσουμε τη γνώση και απλά να προσθέτουμε στη γνώση. Το να παύουμε να χρησιμοποιούμε ξεπερασμένες τεχνικές και τρόπους σκέψης, ή ακόμα και να "αποσύρουμε" τμήματα γνώσης και εμπειρίας, συμβάλλει στη ζωτικότητα και την εξέλιξη της γνώσης.

1.4.9 Δεν υπάρχει αρχηγός

Στην ουσία, η γνώση είναι μια κοινωνική διαδικασία. Μόνο όταν οι άνθρωποι συνεργάζονται γεννιέται και αξιοποιείται η γνώση. Αυτό σημαίνει ότι δεν είναι δυνατόν ένα συγκεκριμένο άτομο να αναλάβει την ευθύνη της συλλογικής γνώσης. Οι διαχειριστές της γνώσης δε διαχειρίζονται την ίδια γνώση. Αυτό που κάνουν είναι να βοηθούν στη εύρεση και διαχείριση διαδικασιών για την απόκτηση, δημιουργία, διαμοίραση και εφαρμογή της γνώσης. Ένας διαχειριστής γνώσης μπορεί επίσης να καταφεύγει σε στρατηγικές για την απομάκρυνση εμποδίων και τη δημιουργία κατάλληλων συνθηκών, προκειμένου οι διαδικασίες αυτές να ανθίσουν.

1.4.10 Δεν μπορούμε να επιβάλλουμε κανόνες και συστήματα

Εφόσον η γνώση είναι αυτό-οργανωμένη ο καταλληλότερος τρόπος για να εξελιχθεί είναι να αφαιρέσουμε τα εμπόδια που υπάρχουν στο δρόμο για την αυτό-οργάνωσή της. Η γνώση θα

αναπτυχθεί από μόνη της μόνο σε ένα κατάλληλο περιβάλλον. Είναι άσκοπο να ξοδεύουμε ενέργεια και πόρους δημιουργώντας καθοδηγητικές γραμμές, κανόνες και τεχνολογικά συστήματα για τα οποία κανείς δεν ενδιαφέρεται πραγματικά. Είναι πολύ καλύτερο να δούμε τι δουλεύει αποτελεσματικά και να καταστρώσουμε μεθόδους οι οποίες θα υποστηρίξουν τη φυσική διαδικασία.

1.4.11 Δεν υπάρχει η τέλεια λύση

Δεν υπάρχει κάποια “μαγική” διαδικασία ή καλύτερη πρακτική για την ανάπτυξη της γνώσης και την επέκταση της οργανωσιακής πληροφορίας. Η γνώση πρέπει να υποστηρίζεται σε πολλαπλά επίπεδα και με ποικίλους τρόπους, προκειμένου να ανθίσει. Οι κατάλληλες λύσεις απαιτούν συστηματική προσέγγιση, προσεκτική σκέψη, στοχαστική λογική, πειραματισμό και διαρκή αναπροσαρμογή.

1.4.12 Το πώς ορίζουμε το “πρόβλημα” της γνώσης καθορίζει και το πώς θα το χειριστούμε

Η γνώση έχει πολλές πτυχές. Εάν ένας οργανισμός έχει ως κύριο στόχο την κατοχή γνώσης, τότε οι προσπάθειες επικεντρώνονται στην απόκτηση κωδικοποιημένης γνώσης η οποία προστατεύεται μέσω πνευματικών δικαιωμάτων και κατοχυρωμένων πατεντών. Εάν κύριος στόχος είναι η διαμοίραση της γνώσης, τότε δίνεται έμφαση στην διαρκή ροή της επικοινωνίας μέσα στον οργανισμό και την καταγραφή των γνωστικών διαδικασιών. Η ενασχόληση με όλες τις πτυχές της γνώσης οδηγεί στην αναζήτηση πιο αποτελεσματικών τρόπων για τη δημιουργία, προσαρμογή και εφαρμογή της. Κάτι τέτοιο απαιτεί εις βάθος κατανόηση της αξίας της γνώσης και ιδιαίτερα μεγάλη αφοσίωση.

2 Συστήματα Διαχείρισης Περιεχομένου

Στο κεφάλαιο αυτό θα ασχοληθούμε με μια συγκεκριμένη κατηγορία εργαλείων διαχείρισης γνώσης, τα Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems – CMS).

2.1 Τι είναι σύστημα διαχείρισης περιεχομένου.

Μέχρι πριν από λίγα χρόνια, ο μόνος τρόπος για να διατηρήσει μια εταιρία το site της ενημερωμένο ήταν να συνάψει συμβόλαιο με μια εταιρία παροχής υπηρεσιών συντήρησης. Τα τελευταία χρόνια, όμως, οι ίδιες οι εταιρίες παροχής τέτοιων υπηρεσιών προσφέρουν μια πολλά υποσχόμενη εναλλακτική λύση. Πολλές από αυτές έχουν αναπτύξει ειδικά συστήματα, τα οποία μειώνουν το χρόνο και το κόστος λειτουργίας ενός δικτυακού τόπου.

Ο όρος Content Management Systems (CMS, Συστήματα Διαχείρισης Περιεχομένου) αναφέρεται στις εφαρμογές που επιτρέπουν στον πελάτη να διαχειρίζεται το δικτυακό του περιεχόμενο, όπως κείμενα, εικόνες, πίνακες κ.λπ., με εύκολο τρόπο, συνήθως παρόμοιο με αυτόν της χρήσης ενός κειμενογράφου. Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν την αλλαγή του περιεχομένου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Οι αλλαγές του site μπορούν να γίνουν από οποιονδήποτε υπολογιστή που είναι συνδεδεμένος στο Διαδίκτυο, χωρίς να χρειάζεται να έχει εγκατεστημένα ειδικά προγράμματα επεξεργασίας ιστοσελίδων, γραφικών κ.λπ. Μέσω ενός απλού φυλλομετρητή ιστοσελίδων (browser), ο χρήστης μπορεί να συντάξει ένα κείμενο και να ενημερώσει άμεσα το δικτυακό του τόπο.

Αυτό που αποκαλούμε πολλές φορές "δυναμικό περιεχόμενο" σε ένα website δεν είναι άλλο παρά οι πληροφορίες που παρουσιάζονται στο site και μπορούν να αλλάζουν από τους ίδιους τους διαχειριστές του μέσω κάποιας εφαρμογής, η οποία ουσιαστικά μπορεί να εισάγει (προσθέτει), διορθώνει και να διαγράφει εγγραφές σε πίνακες βάσεων δεδομένων, όπου τις περισσότερες φορές καταχωρούνται όλες αυτές οι πληροφορίες. Αυτό σημαίνει ότι δεν χρειάζεται να δημιουργηθούν πολλές ξεχωριστές ιστοσελίδες για την παρουσίαση των πληροφοριών στο site, αλλά αρκεί ένας ενιαίος σχεδιασμός στα σημεία όπου θέλουμε να εμφανίζεται το περιεχόμενό μας, καθώς και να υπάρχει ο ειδικός σε κάποια συγκεκριμένη γλώσσα προγραμματισμού (ASP, PHP, Coldfusion, Perl, CGI κ.λπ.), ο οποίος αναλαμβάνει να εμφανίσει τις σωστές πληροφορίες στις σωστές θέσεις.

Έτσι, για το δικτυακό τόπο μιας εφημερίδας π.χ., που απαιτεί εύλογα καθημερινή ενημέρωση αλλά δεν χρησιμοποιεί κάποιο σύστημα Content Management, θα πρέπει ο υπεύθυνος για το σχεδιασμό του (designer) να δημιουργήσει μια σελίδα με τα γραφικά, την πλοήγηση και το περιβάλλον διεπαφής (interface) του website, ο υπεύθυνος ύλης να τοποθετήσει το περιεχόμενο στα σημεία της ιστοσελίδας που θέλει, και να ενημερωθούν οι σύνδεσμοι των υπόλοιπων σελίδων ώστε να συνδέονται με την καινούργια. Αφού την αποθηκεύσει, πρέπει να την ανεβάσει στο website μαζί με τις υπόλοιπες ιστοσελίδες που άλλαξαν.

Αντιθέτως, αν ο δικτυακός τόπος λειτουργεί με χρήση κάποιου συστήματος CMS, το μόνο που έχει να κάνει ο διαχειριστής του είναι να ανοίξει τη σχετική φόρμα εισαγωγής νέου άρθρου στη διαχειριστική εφαρμογή του website και να γράψει ή να επικολλήσει (copy-paste) τα στοιχεία που επιθυμεί. Αυτόματα, μετά την καταχώριση γίνονται από το ίδιο το σύστημα διαχείρισης περιεχομένου όλες οι απαραίτητες ενέργειες, ώστε το άρθρο να είναι άμεσα διαθέσιμο στους επισκέπτες και όλοι οι σύνδεσμοι προς αυτό ενημερωμένοι.

Με την αυξητική τάση χρήσης των CMS στην Ελλάδα και το εξωτερικό, γίνεται εμφανές ότι το μέλλον του Διαδικτύου σε ότι αφορά περιεχόμενο και πληροφορίες που πρέπει να

ανανεώνονται τακτικά, ανήκει στα προγράμματα διαχείρισης περιεχομένου, αφού προσφέρουν πολλά πλεονεκτήματα, ταχύτητα και ευκολίες στη χρήση τους.

Τα Συστήματα Διαχείρισης Περιεχομένου μπορούν να χρησιμοποιηθούν και να αντικαταστήσουν ένα συμβόλαιο συντήρησης επάξια. Τα CMS μπορούν να χρησιμοποιηθούν για:

- Ειδήσεις (εφημερίδες, περιοδικά, πρακτορεία ειδήσεων κ.λπ.)
- Παρουσιάσεις εταιριών και προσωπικού
- Καταλόγους προϊόντων
- Παρουσιάσεις προϊόντων
- Online υποστήριξη
- Αγγελίες και ανακοινώσεις
- Παρουσιάσεις και προβολή γεωγραφικών περιοχών
- Διαφημίσεις
- Δελτία Τύπου
- Όρους και συμβόλαια
- Χάρτες, κατευθύνσεις, οδηγίες

Ένα ολοκληρωμένο CMS πρέπει να μπορεί να διαχειρίζεται όλες τις δυναμικές πληροφορίες του site και να προσφέρει υπηρεσίες που εξυπηρετούν πλήρως τις ανάγκες των διαχειριστών του. Μερικά από τα πλεονεκτήματα και τα χαρακτηριστικά ενός ολοκληρωμένου CMS είναι:

- Γρήγορη ενημέρωση, διαχείριση και αρχειοθέτηση του περιεχομένου του δικτυακού τόπου
- Ενημέρωση του περιεχομένου από οπουδήποτε
- Ταυτόχρονη ενημέρωση από πολλούς χρήστες και διαφορετικούς υπολογιστές
- Να μην απαιτούνται ειδικές τεχνικές γνώσεις από τους διαχειριστές του
- Εύκολη χρήση και άμεση γνώση του τελικού αποτελέσματος, όπως γίνεται με τους γνωστούς κειμενογράφους
- Δυνατότητα αναζήτησης του περιεχομένου που καταχωρείται και αυτόματη δημιουργία αρχείου
- Ασφάλεια και προστασία του σχεδιασμού του site από λανθασμένες ενέργειες, που θα μπορούσαν να δημιουργήσουν προβλήματα στην εμφάνισή του
- Διαχωρισμός του περιεχομένου από το σχεδιασμό και την πλοήγηση (navigation) του δικτυακού τόπου
- Αλλαγή σχεδιασμού ή τρόπου πλοήγησης χωρίς να είναι απαραίτητη η ενημέρωση όλων των σελίδων από τον ίδιο το χρήστη
- Αυτόματη δημιουργία των συνδέσμων μεταξύ των σελίδων και αποφυγή προβλημάτων ανύπαρκτων σελίδων (404 error pages)
- Μικρότερος φόρτος στον εξυπηρετητή (server) και χρήση λιγότερου χώρου, αφού δεν υπάρχουν πολλές επαναλαμβανόμενες στατικές σελίδες, από τη στιγμή που η ανάπτυξη των σελίδων γίνεται δυναμικά
- Όλο το περιεχόμενο καταχωρείται στην/στις βάσεις δεδομένων, τις οποίες μπορούμε πιο εύκολα και γρήγορα να τις προστατεύσουμε τηρώντας αντίγραφα ασφαλείας

Ασφαλώς υπάρχουν και άλλα χαρακτηριστικά και πρόσθετες υπηρεσίες, ανάλογα με το CMS, που άλλοτε χρεώνονται επιπλέον και άλλοτε ενσωματώνονται και προσφέρονται δωρεάν προς χρήση, όπως:

- Εφαρμογή διαχείρισης και προβολής διαφημιστικών banners, δημοσκοπήσεων και παραμετροποίησης (personalisation)
- Δυνατότητα παρουσίασης του περιεχομένου σε συνεργαζόμενα sites (syndication)
- Στατιστικά
- Διαχείριση μελών
- Newsletters
- Forum

3 Κατηγορίες Συστημάτων Διαχείρισης Περιεχομένου με βάση την επεξεργασία.

Υπάρχουν τρεις σημαντικοί τύποι συστημάτων CMS: επεξεργασίας offline (offline processing), επεξεργασίας online (online processing) και υβριδικά συστήματα (hybrid systems). Οι όροι αυτοί περιγράφουν το πότε και το πώς εφαρμόζονται τα πρότυπα παρουσίασης (presentation templates) για να διαμορφώσουν μια ιστοσελίδα.

3.1 Επεξεργασία Offline

Αυτά τα συστήματα προεπεξεργάζονται όλο το περιεχόμενο, εφαρμόζοντας πρότυπα πριν από τη δημοσίευση για να παράγουν ιστοσελίδες. Τα συστήματα Sagar Vignette και Bricolage είναι παραδείγματα αυτού του τύπου συστήματος. Δεδομένου ότι τα συστήματα προεπεξεργασίας δεν απαιτούν έναν κεντρικό υπολογιστή για να εφαρμόσουν τα πρότυπα, μπορούν επίσης να χρησιμοποιηθούν και ως καθαρά εργαλεία σχεδίασης. Το σύστημα Adobe Contribute αποτελεί ένα παράδειγμα αυτής της προσέγγισης.

3.2 Επεξεργασία Online

Αυτά τα συστήματα εφαρμόζουν τα πρότυπα κατ' απαίτηση (on-demand). Ο κώδικας HTML παράγεται όταν ένας χρήστης επισκέπτεται τη σελίδα, ή "τραβιούνται" από μια μνήμη cache. Μερικοί από τα πιο γνωστά συστήματα ανοικτού κώδικα (open source) σε αυτή την κατηγορία είναι τα Mambo, Joomla! , Drupal, WordPress, Zikula και Plone.

3.3 Υβριδικά Συστήματα

Τα συστήματα αυτά συνδυάζουν τις offline και online προσεγγίσεις. Μερικά από αυτά τα συστήματα παράγουν εκτελέσιμο κώδικα (π.χ. JSP, PHP, Perl) παρά στατικό HTML, έτσι ώστε το ίδιο το CMS να μη χρειάζεται να τοποθετηθεί σε κάθε Web server. Άλλα τέτοια συστήματα, όπως το Blossom, είναι σε θέση να λειτουργούν είτε σε offline είτε σε online λειτουργία.

Τα Συστήματα Διαχείρισης Περιεχομένου άρχισαν να αναπτύσσονται επίσημα ως εμπορικά προϊόντα λογισμικού το 1995 με δύο αρχικά προϊόντα, το TeamSite της Interwoven και το Vignette Content Management της Vignette. Με την εξέλιξη του Internet, η σημασία των συστημάτων αυτών ως κομμάτι της υποδομής IT ενός οργανισμού άρχισε να γίνεται ολοένα και μεγαλύτερη. Έτσι, προμηθευτές από παρόμοιες αγορές, συμπεριλαμβανομένων και των Documentum και FileNet, εταιρειών με μακρά παράδοση στην ανάπτυξη λογισμικού Διαχείρισης Εγγράφων (Document Management) άρχισαν να αναπτύσσουν τα δικά τους CMS. Άλλοι προμηθευτές, όπως οι Stellent και RedDot Solutions, άρχισαν να κάνουν την εμφάνισή τους. Μέχρι το 2002, τα τμήματα IT είχαν αρχίσει να αναζητούν ένα μοναδικό προμηθευτή που θα μπορούσε να διαχειριστεί όλο το μη δομημένο περιεχόμενο (έγγραφα, ιστοσελίδες, οπτικοακουστικά μέσα κτλ.) και η Διαχείριση Περιεχομένου Web (WCM) έγινε υποσύνολο μιας νέας υπερκατηγορίας, της Επιχειρησιακής Διαχείρισης Περιεχομένου (Enterprise Content Management – ECM).

4 Κατηγορίες Συστημάτων Διαχείρισης Περιεχομένου με βάση αν είναι ανοικτού ή κλειστού κώδικα.

Υπάρχει μια μεγάλη λίστα από CMS κάποια από τα οποία είναι ελεύθερα ανοικτού κώδικα (free open source) και άλλα τα οποία χρειάζεται να αγοράσουμε άδεια για να τα χρησιμοποιήσουμε (licence). Οι δύο αυτές κατηγορίες CMS με βάση των κώδικα που έχουν γραφτεί είναι τα CMS ανοικτού και κλειστού κώδικα.

4.1 CMS ανοικτού κώδικα.

Wordpress: Το WordPress έχει ένα σύστημα προτύπων ιστού το οποίο χρησιμοποιεί έναν επεξεργαστή προτύπων. Οι χρήστες του μπορούν να αλλάζουν τη θέση διαφόρων στοιχείων του γραφικού περιβάλλοντος χωρίς να χρειάζεται να επεξεργάζονται κώδικα PHP ή HTML. Μπορούν επίσης να εγκαθιστούν και να αλλάζουν μεταξύ διαφόρων οπτικών θεμάτων. Μπορούν ακόμα να επεξεργαστούν τον κώδικα PHP και HTML στα οπτικά θέματα, προκειμένου να επιτύχουν προχωρημένες τροποποιήσεις.

Joomla: Είναι ένα σύστημα διαχείρισης περιεχομένου (CMS) με αρκετές δυνατότητες, εξαιρετικά ευέλικτο και φιλικό. Η εφαρμογή αυτή χρησιμοποιείται για τη δημοσίευση στο διαδίκτυο μιας προσωπικής ιστοσελίδας, αλλά και ενός εταιρικού δικτυακού τόπου. Είναι προσαρμόσιμο σε περιβάλλοντα επιχειρηματικής κλίμακας όπως τα intranets μεγάλων επιχειρήσεων ή οργανισμών. Οι δυνατότητες επέκτασής του είναι πρακτικά μεγάλες.

Drupal: Το Drupal είναι ένα ανοικτού κώδικα Σύστημα διαχείρισης περιεχομένου (CMS). Χρησιμοποιείται για την δημιουργία και την διαχείριση πολλών και διαφορετικών ιστότοπων. Η σταθερότητα, ευελιξία και η ασφάλεια του είναι μερικά απ τα σημεία αναφοράς του.

Xoops: Το σύστημα XOOPS είναι και αυτό ένα open-source CMS γραμμένο σε PHP. Είναι σχεδιασμένο τμηματικά και έχει αντικειμενοστραφή χαρακτηριστικά. Η βάση δεδομένων που χρησιμοποιεί το XOOPS είναι MySQL. Το όνομα XOOPS είναι ακρωνύμιο των λέξεων eXtensible Object Oriented Portal System.

4.2 CMS κλειστού κώδικα.

Vignette CMS: Τα προϊόντα της Vignette βοηθούν τις επιχειρήσεις να αποκτήσουν και να διαχειρίζονται τις πληροφορίες που χρειάζονται. Είναι μια έμπειρη εταιρία αποδοτικότητας, η οποία στοχεύει στην αύξηση της παραγωγικότητας, μείωση του κόστους και στην βελτίωση της εμπειρίας του χρήστη. Οι Intranet, extranet και internet λύσεις της συμπεριλαμβάνουν portal, integration, enterprise content management και δυνατότητες συνεργασίας που μπορούν να αποδώσουν μοναδικά προτερήματα.

Jalios JCMS: Είναι ένα enterprise content management (ECM) που αναπτύχθηκε από την Jalios, μια γαλλική εταιρεία που ιδρύθηκε το 2001 και εδρεύει στο Παρίσι, Γαλλία. Περιλαμβάνει, μεταξύ άλλων, τα ακόλουθα χαρακτηριστικά: διαχείριση περιεχομένου, διαχείριση εγγράφων, collaboration, workflow και πύλες.

Dynamicweb: Είναι προϊόν της Dynamicweb Software Ltd η οποία εξειδικεύεται στην παροχή web-based λύσεις λογισμικού για την επαγγελματική αγορά. Όλα τα προϊόντα είναι modular και έχουν επεκταθεί εύκολα για να ταιριάζουν στις ανάγκες των πελατών. Η Dynamicweb Software Ltd αναπτύσσει τρία κύρια προϊόντα Synkron Via, Dynamicweb ηλεκτρονικού εμπορίου και Dynamicweb CMS. Το Dynamicweb CMS είναι ένα πολυγλωσσικό και φιλικό Σύστημα Διαχείρισης Περιεχομένου (CMS), το οποίο μπορεί να χρησιμοποιηθεί για τη διαχείριση του

περιεχομένου στους δικτυακούς τόπους, extranets και Intranets. Βασίζεται στην τεχνολογία Microsoft .NET και έχει επιλεγθεί σε περισσότερες από 3.000 εταιρίες και Οργανισμούς σε διάφορες χώρες. (Iterating, 2009)

Powerfront CMS: Το PowerFront παρέχει μια ολοκληρωμένη λύση διαχείρισης περιεχομένου που μπορεί να περιλαμβάνει: τη διαχείριση περιεχομένου, τον σχεδιασμό ιστοσελίδων, θέματα ασφάλειας, το ηλεκτρονικό εμπόριο, procurement, reporting options και υποστήριξη. Στόχος της είναι η υποστήριξη ιστοσελίδων των επιχειρήσεων, intranets, extranets ή procurement websites.

5 Εισαγωγή στο Σύστημα Joomla

Στο κεφάλαιο αυτό θα ασχοληθούμε με το Σύστημα Διαχείρισης Περιεχομένου *Joomla*. Θα εξηγήσουμε τι ακριβώς είναι αυτό το εργαλείο και θα κάνουμε μια σύντομη ιστορική αναδρομή.

5.1 Τι είναι το Joomla

Το Joomla! είναι λογισμικό ανοικτού κώδικα (open source) το οποίο υλοποιεί τη λειτουργικότητα συστήματος διαχείρισης περιεχομένου (Content Management System – **CMS** από τώρα και στο εξής) στον Παγκόσμιο Ιστό (World Wide Web) και σε ενδοδίκτυα (intranets).. Οι λέξεις-κλειδιά είναι “διαχείριση” και “περιεχόμενο” και αναφέρονται σε ένα σύστημα λογισμικού το οποίο χρησιμοποιείται για την οργάνωση, ταξινόμηση και αρχειοθέτηση πληροφορίας. Το διαδίκτυο με τη σημερινή του μορφή έχει εκατομμύρια ιστότοπους (websites) οι οποίοι είναι βασισμένοι σε CMS. Οι ιστότοποι αυτοί έχουν ομάδα διαχείρισης (administration team) καθώς και τελικούς χρήστες (users). Πρακτικά, τα CMS είναι ιδιαίτερα δημοφιλή γιατί επιτρέπουν τη δημιουργία και συντήρηση εξελιγμένων ιστότοπων με ελάχιστο ή μηδενικό προγραμματιστικό κόστος, μιας και προσφέρουν όλη την απαραίτητη υποδομή για την υλοποίηση όλων των εργασιών που είναι απαραίτητες για αυτό.

Η ιδιαιτερότητα των CMS είναι ότι αποτελούν λογισμικό του οποίου η διαχείριση μπορεί να γίνει μέσω ενός φυλλομετρητή (π.χ. Mozilla Firefox, Safari, Internet Explorer κ.α.). Επιπρόσθετα, όλη η λειτουργία του συστήματος αυτού είναι διαθέσιμη μέσω της διεπαφής του φυλλομετρητή, αφού αυτό είναι εγκατεστημένο και προσβάσιμο μέσω μιας διεύθυνσης διαδικτύου. Είναι γραμμένο σε γλώσσα PHP και χρησιμοποιεί τη βάση δεδομένων MySQL εξ ορισμού.

5.2 Δομή ενός CMS

Τα κυριότερα δομικά στοιχεία ενός CMS είναι τα παρακάτω:

- Διεπαφή χρήστη (Front-End)
- Διεπαφή διαχειριστών (Back-End)
 - Ρυθμίσεις συστήματος (configuration settings)
 - Ρυθμίσεις πρόσβασης χρηστών (Access control settings)
- Πρότυπα (Templates)
- Επεκτάσεις (Extensions)

5.3 Διεπαφές Χρήστη και διαχειριστών

Η διεπαφή χρήστη αποτελεί το μέρος του CMS που είναι διαθέσιμο στους επισκέπτες (απλούς ή διαπιστευμένους) του ιστότοπου. Από την άλλη πλευρά, η διεπαφή διαχειριστών αποτελεί το επίπεδο διαχείρισης για την ομάδα συντήρησης και προσφέρει λειτουργίες ρύθμισης συστήματος, διαχείρισης χρηστών (προσθήκη / αφαίρεση / θέματα πρόσβασης), διαχείρισης επεκτάσεων και τέλος του περιεχομένου του ιστότοπου αυτού καθεαυτού. Το Joomla επιτρέπει στους διαχειριστές να θέτουν καθολικές παραμέτρους διαμόρφωσης (global configuration parameters) οι οποίες εφαρμόζονται σε όλα τα άρθρα. Όλες οι σελίδες του συστήματος συμμορφώνονται εξ ορισμού με αυτές τις παραμέτρους, αλλά κάθε σελίδα μπορεί να έχει τις δικές της ρυθμίσεις για κάθε παράμετρο. Για παράδειγμα, μπορούμε να επιλέξουμε να εμφανίζεται ο συντάκτης ενός άρθρου, να μην εμφανίζεται, ή απλά να επιλέξουμε την καθολική παράμετρο "εμφάνιση συντάκτη" ("show author").

5.4 Πρότυπα

Τα πρότυπα (templates) αποτελούν το αισθητικό και λειτουργικό επίπεδο του ιστοτόπου που θα βασιστεί σε CMS. Μπορούμε να θεωρήσουμε τα πρότυπα ως κενές μακέτες πάνω στις οποίες θα τοποθετηθεί το περιεχόμενο που θα επιλέξουμε να εμφανίζεται από το CMS. Στα πρότυπα καθορίζονται οι παράμετροι εμφάνισης της σελίδας (χρώματα, διαστάσεις, τυπογραφία) καθώς επίσης παρέχονται όλα τα αρχεία γραφικών που θα χρησιμοποιηθούν για την εμφάνισή της.

5.5 Επεκτάσεις

Οι επεκτάσεις αποτελούν κομμάτια λογισμικού (software components) τα οποία ακολουθούν ένα καθορισμένο προγραμματιστικό μοντέλο, το οποίο τους επιτρέπει να αλληλεπιδρούν με το CMS και να επεκτείνουν ακολουθώντας τη λειτουργικότητά του. Οι επεκτάσεις κάνουν ένα CMS όπως το Joomla! εξαιρετικά ισχυρό και αποτελούν σημαντικό μοχλό ανάπτυξης και διάδοσης αυτού, μέσω μιας εξαιρετικά δραστήριας κοινότητας προγραμματιστών.

Ένα παράδειγμα αυτών των επεκτάσεων είναι τα "Plugins" (πρωτίτερα γνωστά ως "Mambots"). Τα Plugins είναι επεκτάσεις του υπόβαθρου (background) του Joomla και τα οποία επεκτείνουν τις λειτουργικές δυνατότητες του συστήματος. Πέρα από τα Plugins, υπάρχουν διαθέσιμες και άλλες πιο πολύπλοκες επεκτάσεις. Τα "Συστατικά" ("Components") επιτρέπουν στους διαχειριστές να εκτελούν εργασίες όπως η δημιουργία κοινοτήτων, η επέκταση των χαρακτηριστικών των χρηστών, η δημιουργία αντιγράφων ασφαλείας (backups) του ιστοχώρου, η μετάφραση περιεχομένου και η δημιουργία διευθύνσεων URL που είναι πιο φιλικές στο χρήστη. Τα "Modules" εκτελούν εργασίες όπως η εμφάνιση ενός ημερολογίου ή η εισαγωγή αυτοσχέδιου κώδικα μέσα στον κώδικα του συστήματος. Αυτή την στιγμή το Joomla στην επίσημη ιστοσελίδα <http://extensions.joomla.org/> του απαριθμεί περισσότερες από 5000 επεκτάσεις.

5.6 Ιστορική Αναδρομή

Στα μέσα του 2000, η αγορά της Διαχείρισης Περιεχομένου Web διασπάστηκε ακόμη περισσότερο με την εμφάνιση νέων προμηθευτών που ήρθαν να συμπληρώσουν τους παραδοσιακούς προμηθευτές WCM. Σήμερα, τα Συστήματα Διαχείρισης Περιεχομένου Web χωρίζονται σε διάφορες κατηγορίες: *Επιχειρησιακά (Enterprise)* με κυριότερα τα Vignette, Interwoven, Documentum και Oracle, *Μεσαίας Αγοράς (Mid-Market)* με κυριότερα τα Ektron, PaperThin και Ingeniux, *Ανοικτού Κώδικα (Open Source)* με κυριότερα τα Joomla, Drupal, PHP-Nuke και Alfresco, και *SaaS (Software as a Service – Λογισμικό ως Υπηρεσία)* με κυριότερα τα Clickability, Crownpeak και Hot Banana.

Το Joomla απέκτησε πνοή στις 17 Αυγούστου 2005 από την ομάδα ανάπτυξης του Mambo, του οποίου θεωρείται απόγονος. Εκείνη την εποχή το όνομα Mambo ήταν το εμπορικό σήμα της Miro International Pty Ltd, η οποία δημιούργησε ένα μη κερδοσκοπικό οργανισμό με σκοπό τη χρηματοδότηση του όλου εγχειρήματος και την νομική του προστασία από μηνύσεις. Η ομάδα ανάπτυξης ισχυρίστηκε ότι πολλά στοιχεία της δομής του ιδρύματος ήταν ενάντια σε προηγούμενες συμφωνίες που είχαν γίνει από την εκλεγμένη επιτροπή, παρουσίαζαν έλλειψη σε διαβουλεύσεις με τους βασικούς μετόχους και περιελάμβαναν εγκυκλίους που παραβίαζαν θεμελιώδεις αρχές της αρχιτεκτονικής ανοικτού κώδικα.

Μέσα σε αυτό το κλίμα, η ομάδα ανάπτυξης δημιούργησε μια ιστοθέση με την ονομασία OpenSourceMatters με σκοπό να διαμοιράσει πληροφορίες σε χρήστες, υπεύθυνους ανάπτυξης, σχεδιαστές και στο ευρύ κοινό γενικότερα. Ο αρχηγός του εγχειρήματος Andrew Eddie ή αλλιώς "MasterChief" έγραψε μία ανοικτή επιστολή η οποία αναρτήθηκε στο δημόσιο forum του ιστοχώρου mamboserver.com. Χίλιοι άνθρωποι έγιναν μέλη του ιστοχώρου opensourcematters.org μέσα σε μία μέρα, οι περισσότεροι από τους οποίους εξέφρασαν μηνύματα συμπαράστασης και υποστήριξης.

Στις δύο εβδομάδες που ακολούθησαν την ανακοίνωση του Andrew Eddie, οι ομάδες αναδιοργανώθηκαν και η κοινότητα συνέχιζε να μεγαλώνει. Ο καθηγητής Eben Moglen και η εταιρεία του Software Freedom Law Center (SFLC) βοήθησαν την ομάδα ανάπτυξης του Joomla ξεκινώντας τον Αύγουστο του 2005. Η SFLC παρέχει μέχρι σήμερα νομική υποστήριξη και καθοδήγηση στο εγχείρημα Joomla.

Στις 18 Αυγούστου 2005 ο Andrew Eddie ζήτησε τη συμβολή της κοινότητας για προτεινόμενα ονόματα για το εγχείρημα. Η ομάδα ανάπτυξης τόνισε ότι αυτή θα έπαιρνε την τελική απόφαση για το όνομα βασιζόμενη στη συμβολή της κοινότητας. Το όνομα το οποίο επέλεξε τελικά η κοινότητα δεν ήταν στη λίστα με τα ονόματα που είχαν προταθεί από την κοινότητα. Αντίθετα, επιλέχθηκε μέσα από διαβουλεύσεις με μία επαγγελματική συμβουλευτική ομάδα, η οποία πληρώθηκε από την Open Source Matters. Στις 1 Σεπτεμβρίου 2005, το νέο όνομα, "Joomla", ανακοινώθηκε.

Στις 7 Σεπτεμβρίου 2005 η ομάδα ανάπτυξης ζήτησε την υποβολή λογότυπων από την κοινότητα, κάλεσε την κοινότητα να ψηφίσει για το νέο λογότυπο και ανακοίνωσε την απόφαση της κοινότητας στις 22 Σεπτεμβρίου 2005.

Η πρώτη έκδοση του Joomla (Joomla 1.0.0) ανακοινώθηκε στις 16 Σεπτεμβρίου 2005. Στην ουσία επρόκειτο για μια επανέκδοση του Mambo 4.5.2.3 εμπλουτισμένη με διάφορες διορθώσεις λειτουργικών προβλημάτων (bugs) και θεμάτων ασφαλείας. Το Joomla κέρδισε το βραβείο Packt Publishing Open Source Content Management System το 2006 και το 2007.

Στις 27 Οκτωβρίου 2008 η Packt Publishing ανακηρύσσει τον Johan Janssens "Most Valued Person" (MVP) για την δουλειά του σαν έναν από τους βασικούς προγραμματιστές του 1.5 Joomla Framework and Architecture. Το 2009 ο Louis Landry παρέλαβε το βραβείο "Most Valued Person" για τον ρόλο του ως συντονιστής του προγραμματισμού και της αρχιτεκτονικής του Joomla. Στις 22 Ιανουαρίου 2008 βγήκε η έκδοση 1.5 του Joomla με πιο πρόσφατη την 1.5.1.8 (28 Μαΐου 2010). Τον Μάιο και τον Ιούνιο του 2010 βγήκαν κάποιες beta versions 1.6 του Joomla για δοκιμαστικούς σκοπούς.

6 Εγκατάσταση του Joomla σε περιβάλλον Windows.

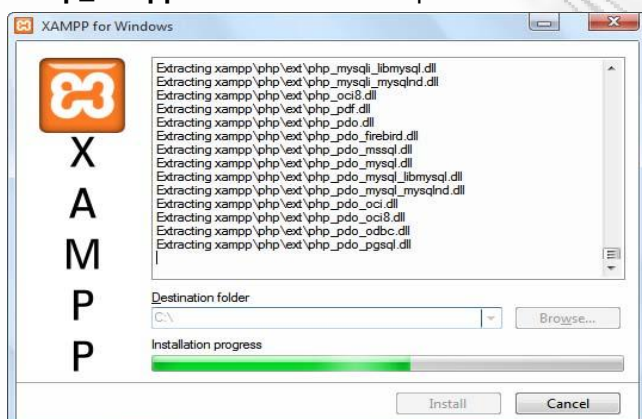
6.1 Εγκατάσταση λογισμικού XAMPP

Για να μπορέσουμε να εγκαταστήσουμε το Joomla είναι απαραίτητο να έχουμε έναν server εγκατεστημένο πάνω στον οποίο θα μπορεί να τρέξει το Joomla. Ο πιο εύκολος τρόπος να εγκαταστήσουμε έναν server στον υπολογιστή μας είναι με την βοήθεια του Xampp. Το λογισμικό Xampp διατίθεται δωρεάν και μπορούμε να το κατεβάσουμε από την ιστοσελίδα (<http://www.apachefriends.org/en/xampp.html>.) το οποίο με ελάχιστες ρυθμίσεις και χωρίς εγκατάσταση, παρέχει ένα πλήρες σύστημα Apache, PHP, MySQL μαζί με το λογισμικό phpMyAdmin.

Το XAMPP είναι ένα αρκτικόλεξο για:

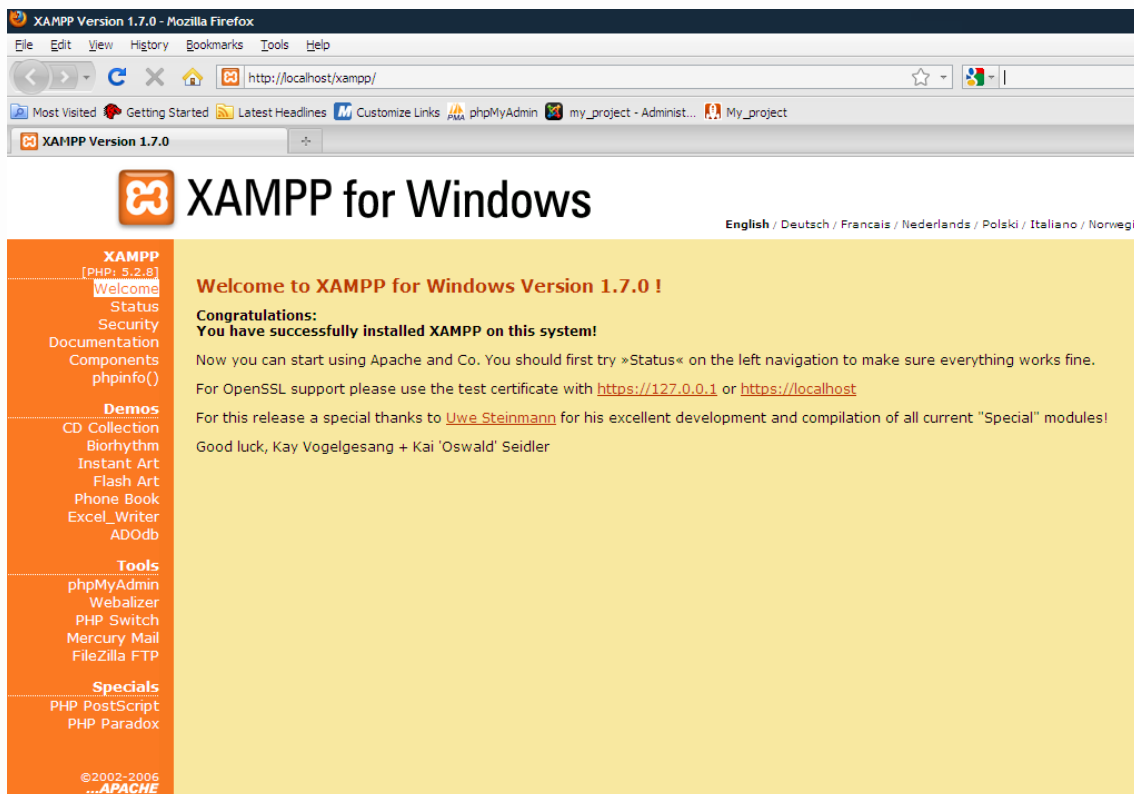
- **X** (meaning cross-platform) X (δηλαδή cross-platform)
- **A**pache HTTP Server
- **M**ySQL
- **P**HP
- **P**erl

Για να εγκαταστήσουμε το Xampp πατάμε το exe αρχείο, στο extract path δίνουμε τον κεντρικό κατάλογο του σκληρού μας δίσκου (για ευκολία π.χ. C:\) και το αρχείο **setup_xampp.bat**. εκτελείται αυτόματα. Το XAMPP θα εγκατασταθεί στον κατάλογο C:\xampp

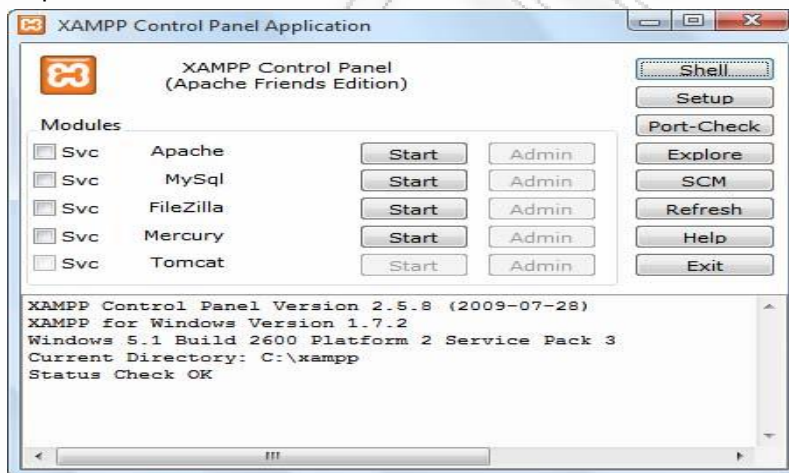


Εικόνα: Xampp installer

Στη συνέχεια ανοίγουμε έναν φυλλομετρητή (π.χ. Mozilla Firefox) και πηγαίνουμε στη διεύθυνση <http://localhost/xampp> (ή <http://127.0.0.1/xampp> εάν έχουμε Windows Vista). Εάν όλα έχουν πάει καλά θα πρέπει να μας εμφανιστεί μια σελίδα όμοια αυτής της εικόνας.



Όπως γίνεται εύκολα αντιληπτό, στην αριστερή πλευρά της εικόνας 4.2.1-1 φαίνονται οι λειτουργίες που είναι ενσωματωμένες στο xampp και στην κεντρική οθόνη η περιγραφή της λειτουργίας που επιλέγει ο χρήστης. Επίσης με τη βοήθεια του "XAMPP Control Panel" μπορούμε να ξεκινήσουμε ή να σταματήσουμε οποιαδήποτε λειτουργία του Xampp. Δείτε την παρακάτω εικόνα.



Εικόνα: XAMPP Control Panel

Για να ανεβάσουμε αρχεία στον Web Server: Πηγαίνουμε στον κατάλογο **C:\xampplite\htdocs** δημιουργούμε ένα κατάλογο π.χ. my_project. Τα αρχεία που θα τοποθετηθούν στον κατάλογο αυτό θα είναι προσβάσιμα στη διεύθυνση http://localhost/my_project (ή http://127.0.0.1/my_project)

6.2 Apache HTTP Server

Ο Apache HTTP Server, κοινώς αποκαλούμενο απλώς Apache IP είναι ένας web server που διαδραματίζει βασικό ρόλο στην αρχική ανάπτυξη του World Wide Web και το 2009 έγινε ο πρώτος web server που θα ξεπεράσει τα 100 εκατομμύρια web site. Netscape Communications Corporation Sun Java System Web Server Unix Ο Apache ήταν η πρώτη βιώσιμη εναλλακτική λύση για την Netscape Communications Corporation web server (γνωστός σήμερα ως Sun Java System Web Server), και που έκτοτε έχει εξελιχθεί και για άλλα Unix-web servers που βασίζονται στην άποψη της λειτουργικότητας και των επιδόσεων. Linux Η πλειοψηφία των web servers χρησιμοποιώντας Apache είναι Linux web servers.

Ο Apache αναπτύσσεται και συντηρείται από μια ανοιχτή κοινότητα προγραμματιστών, υπό την αιγίδα του Apache Software Foundation. operating systems Unix GNU Free BSD Linux Solaris Novell NetWare Mac OS X Microsoft Windows OS/2 TPF eComStation Η εφαρμογή είναι διαθέσιμη για μια μεγάλη ποικιλία από λειτουργικά συστήματα, συμπεριλαμβανομένων των Unix, το GNU, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS / 2, TPF, και eComStation. Apache License free software open source software και χαρακτηρίζεται ως ελεύθερο λογισμικό και λογισμικό ανοικτού κώδικα.

Από τον Απρίλιο του 1996 Apache έχει τα πιο δημοφιλή HTTP server για το World Wide Web. [update]websites[3] Από τον Μάρτιο του 2009 [ενημέρωση] Apache εξυπηρετούνται άνω των 46% όλων των δικτυακών τόπων και άνω του 66% της εκατ. πολυσύχναστες.

6.3 MySQL

Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) το οποίο έχει περισσότερα από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα λειτουργεί ως διακομιστής που παρέχει πολλαπλούς χρήστες πρόσβαση σε μια σειρά από βάσεις δεδομένων.

Το έργο του πηγαίου κώδικα είναι διαθέσιμο, υπό τους όρους της GNU General Public License, καθώς και στο πλαίσιο ποικίλων συμφωνιών. Η MySQL ανήκει και χρηματοδοτείται από μια και μόνο για το κέρδος-επιχείρηση, στη σουηδική εταιρεία MySQL AB, τώρα μια θυγατρική της Sun Microsystems, που κατέχει τα πνευματικά δικαιώματα για τα περισσότερα από τα codebase. Oracle Corp. Oracle database Στις 20 Απριλίου, 2009 Oracle Corp., που αναπτύσσει και πωλεί το ιδιόκτητο βάσης δεδομένων της Oracle, ανακοίνωσαν μια συμφωνία για την απόκτηση Sun Microsystems.

6.4 PHP

Η PHP είναι μια scripting γλώσσα που αρχικά είχε σχεδιαστεί για την παραγωγή δυναμικών ιστοσελίδων. command line interface standalone graphical applications. Έχει εξελιχθεί στο να περιλαμβάνει μια γραμμή εντολών και μπορεί να χρησιμοποιηθεί σε standalone γραφικών εφαρμογών.

Ενώ PHP δημιουργήθηκε αρχικά από Rasmus Lerdorf το 1995, η κύρια εφαρμογή της PHP είναι τώρα παράγεται με την ομάδα της PHP και χρησιμεύει ως το *de facto* πρότυπο για την PHP, διότι δεν υπάρχει επίσημων προδιαγραφών.

Η PHP είναι μια ευρέως χρησιμοποιούμενη, γενικού σκοπού scripting γλώσσα η οποία είναι ειδικά κατάλληλη για ανάπτυξη web και μπορεί να ενσωματωθεί στην HTML. Είναι γενικά εκτελείται σε έναν web server, λαμβάνοντας ως PHP κώδικα και να δημιουργήσει ιστοσελίδες ως παραγωγή. Μπορεί να αναπτυχθεί σε web servers και σε σχεδόν κάθε λειτουργικό σύστημα και πλατφόρμα δωρεάν. Η PHP είναι εγκατεστημένη σε πάνω από 20 εκατομμύρια ιστοσελίδες και 1 εκατ. web servers.

6.5 Perl

Η Perl είναι μια υψηλού επιπέδου, δυναμική γλώσσα προγραμματισμού. Η Perl είχε αρχικά αναπτυχθεί από τον Larry Wall, μια γλωσσολόγος εργασίας ως διαχειριστής συστήματος για τη NASA, το 1987, ως γενικό σκοπό Unix scripting γλώσσα για να κάνει πιο εύκολη την επεξεργασία έκθεσης. Από τότε, έχει υποστεί πολλές αλλαγές και αναθεωρήσεις και να γίνει ευρέως δημοφιλής μεταξύ των προγραμματιστών. Ο Larry Wall συνεχίζει να επιβλέπει την ανάπτυξη του πυρήνα της γλώσσας, και την επερχόμενη έκδοση, Perl 6.

Η Perl δανείζεται στοιχεία από άλλες γλώσσες προγραμματισμού συμπεριλαμβανομένης της C, shell scripting (sh), AWK, και sed. Η γλώσσα δημιουργεί κείμενο εγκατάστασης και επεξεργασίας δεδομένων, με διευκόλυνση και εύκολο χειρισμό των αρχείων κειμένου. Χρησιμοποιείται επίσης για τον προγραμματισμό των γραφικών, το σύστημα διοίκησης, του προγραμματισμού του δικτύου, των εφαρμογών που απαιτούν βάση δεδομένων και η πρόσβαση CGI προγραμματισμού για το Web. Το όνομα Perl είναι η μετάφραση του: "το ελβετικό στρατό αλυσοπρίονα των γλωσσών προγραμματισμού" λόγω της ευελιξίας και της προσαρμοστικότητας.

6.6 Εγκατάσταση Joomla!

Αφού παρουσιάσαμε τις διαδικασίες ρύθμισης του συστήματος που θα φιλοξενήσει το Joomla! Στη συνέχεια αυτού του κεφαλαίου θα εστιάσουμε αφενός στην εγκατάσταση, αφετέρου στις αρχικές ρυθμίσεις που χρειάζονται για να έχουμε ένα πλήρως λειτουργικό σύστημα διαχείρισης περιεχομένου.

Το πρώτο βήμα που χρειάζεται να κάνουμε είναι να προχωρήσουμε στη μεταφόρτωση του Joomla από την επίσημη ιστοσελίδα <http://www.joomla.org>. Στο σημείο αυτό πρέπει να αναφερθεί ότι δεκάδες ανεξάρτητοι ερασιτέχνες και επαγγελματίες προγραμματιστές έχουν δημιουργήσει προσαρμοσμένες εκδόσεις του Joomla με προεγκατεστημένες γλώσσες και πακέτα λογισμικού για να διευκολύνουν τους αρχάριους. Τα αρχεία που θα χρειαστούμε για να ξεκινήσουμε είναι:

- Διανομή Joomla 1.5.x. Η τελευταία έκδοση είναι η 1.5.18
<http://www.joomla.org/download.html/> Joomla_1.5.18-Stable- Full_Package.zip
- Ελληνική μετάφραση για Joomla 1.5.18
http://joomlancode.org/gf/project/jtranslation/frs/?action=FrsReleaseBrowse&frs_package_id=25
el-GR.site.joomla1.5.18-v1.0.0.tar.gz

Στη συνέχεια αποσυμπίεζουμε τη διανομή του Joomla εντός ενός φακέλου με όνομα **joomla** ή όπως αλλιώς θέλει ο καθένας στο root του εξυπηρετητή. Το root του εξυπηρετητή είναι ο φάκελος στον οποίο βρίσκονται τα αρχεία για μία σελίδα η οποία θα απαντά στο url http://ονομα_server/. Ανάλογα με το περιβάλλον στο οποίο είναι εγκατεστημένος ο εξυπηρετητής αλλάζει και η τοποθεσία του φακέλου root. Ενδεικτικά:

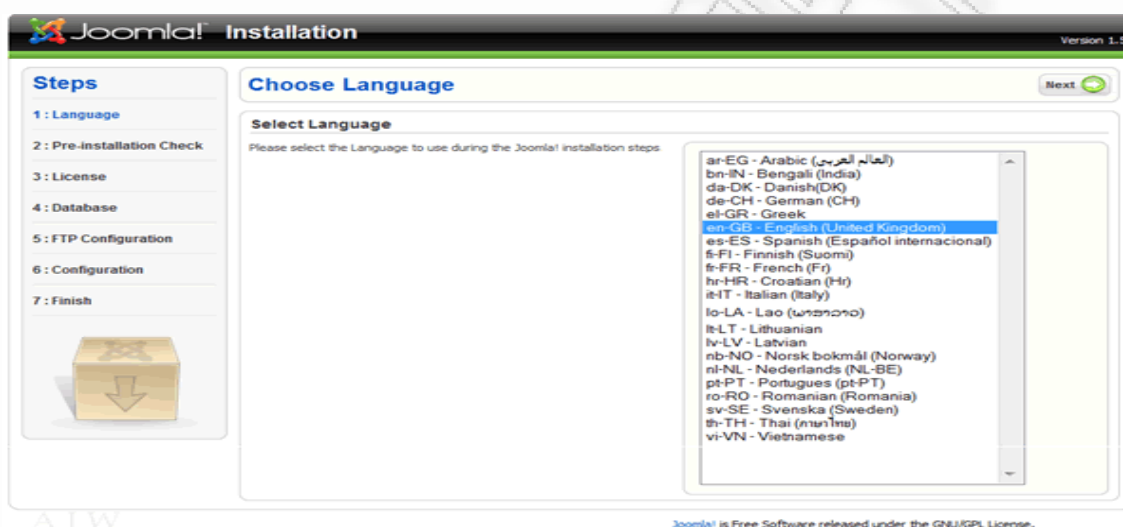
- Σε περιβάλλον Unix είναι συνήθως /var/www
- Σε περιβάλλον Windows και IIS είναι C:\inetpub\wwwroot
- Σε περιβάλλον Windows και Apache είναι C:\φάκελος που έχουμε τον Apache\htdocs
- Στην περίπτωση μας (όπου έχουμε εγκαταστήσει τον εξυπηρετητή Apache σε περιβάλλον Windows στο φάκελο C:\xampp), ο root κατάλογος είναι C:\xampp\htdocs.

Εφόσον έχουν πάει όλα καλά με την αποσυμπίεση του Joomla θα πρέπει να έχει δημιουργηθεί ένας φάκελος εντός του root, ο οποίος θα περιέχει όλα τα αρχεία της διανομής του Joomla. Για να ελέγξουμε ότι είμαστε έτοιμοι να προχωρήσουμε στο επόμενο βήμα της εγκατάστασης μας, αρκεί να ξεκινήσουμε (εάν δεν το έχουμε κάνει ήδη) τον εξυπηρετητή

Apache και τη βάση MySQL, να ανοίξουμε τον φυλλομετρητή (browser) της επιλογής μας (κατά προτίμηση Mozilla Firefox) και στη συνέχεια να πάμε στη δ/ση: http://localhost/my_project/index.php ή http://127.0.0.1/my_project/index.php (my_project έχω ονομάσει το φάκελο εγκατάστασης του Joomla).

6.7 Επιλογή γλώσσας (Language Choice)

Από τη στιγμή που θα εισάγουμε τη διεύθυνση URL το πρώτο βήμα για την εγκατάσταση του Joomla είναι η επιλογή της γλώσσας εγκατάστασης. Όπως παρατηρούμε στην εικόνα παρακάτω, η εγκατάσταση του Joomla γίνεται σε πολυγλωσσικό περιβάλλον και αποτελείται από 7 βήματα τα οποία παρουσιάζονται με τη μορφή ενός εύχρηστου wizard. Επιλέγουμε την γλώσσα προτίμησης μας για την εγκατάσταση του Joomla και πατάμε στο κουμπί «Next» (βρίσκεται πάνω δεξιά) για να μεταφερθούμε στο επόμενο βήμα της εγκατάστασης.

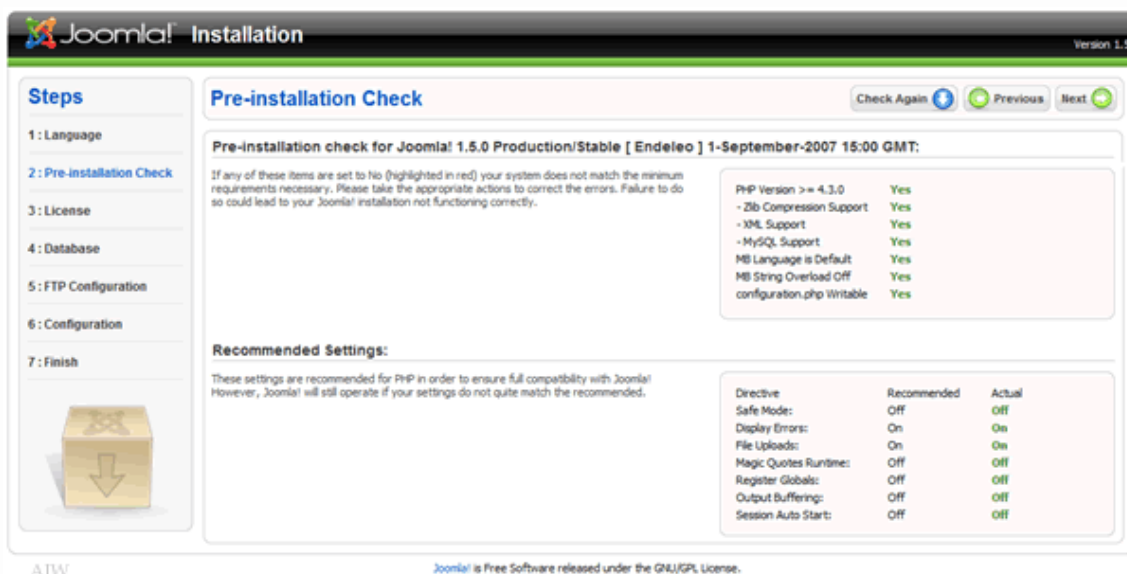


Εικόνα: Πρώτη εικόνα εγκατάστασης Joomla

Τονίζεται ότι τα αρχεία γλωσσών που βρίσκονται στο πακέτο του Joomla έχουν σχέση μόνο με τη διαδικασία εγκατάστασης και δεν επηρεάζουν το Front-end και Back-end του συστήματος, τα οποία εξ ορισμού είναι γραμμένα στα αγγλικά. Παρ' όλα αυτά υπάρχουν διάφορες γλώσσες στις οποίες έχει μεταφραστεί το Joomla. (έχω δώσει link παραπάνω)

6.8 Έλεγχος προ εγκατάστασης (Pre-installation Check)

Η επόμενη οθόνη που εμφανίζεται στο φυλλομετρητή δείχνει αν ο διακομιστής μας πληρεί όλα τα απαραίτητα κριτήρια για την επιτυχημένη ολοκλήρωση της εγκατάστασης του Joomla



Εικόνα: Οθόνη ελέγχου προ εγκατάστασης

Εάν σε οποιοδήποτε από αυτά τα κριτήρια υπάρχει από δίπλα η λέξη No (με εξαίρεση την περίπτωση το αρχείο configuration.php να είναι εγγράψιμο ή όχι), τότε πρέπει να διορθώσουμε το πρόβλημα, προτού προχωρήσουμε με την εγκατάσταση, καθώς όλα αυτά τα κριτήρια αποτελούν βασικές προϋποθέσεις για μια επιτυχημένη εγκατάσταση. Ενδεχόμενη αποτυχία εκπλήρωσης αυτών των κριτηρίων μπορεί να οδηγήσει σε προβληματική λειτουργία της ιστοθέσης, ή την εκτέλεση επιπλέον διαδικασιών μετά το τέλος της εγκατάστασης.

Το δεύτερο κομμάτι ρυθμίσεων σε αυτό το βήμα περιλαμβάνει συγκεκριμένες ρυθμίσεις μέσα στον κώδικα PHP. Μολονότι οι ρυθμίσεις αυτές είναι προτιμώμενες, η εγκατάσταση του Joomla θα συνεχιστεί ακόμα και σε περίπτωση μη τήρησής τους, αν και ενδέχεται η ιστοθέση μας να είναι λιγότερο ασφαλής αν οι ρυθμίσεις αυτές δε διορθωθούν. Η διόρθωση των ρυθμίσεων μπορεί να γίνει ανά πάσα στιγμή. Στην οθόνη του φυλλομετρητή η πρώτη στήλη δείχνει τις προτεινόμενες ρυθμίσεις και η δεύτερη τις υπάρχουσες ρυθμίσεις του διακομιστή.

Τονίζεται ότι δε χρειάζεται να βγούμε από τη διαδικασία εγκατάστασης, προκειμένου να αλλάξουμε κάποια ρύθμιση. Μόλις πραγματοποιήσουμε μία ρύθμιση, απλά πατάμε το κουμπί *Check Again* και γίνεται επανεκτίμηση της κατάστασης. Υπάρχει επίσης η δυνατότητα να επιστρέψουμε στην οθόνη επιλογής γλώσσας πατώντας το πλήκτρο *Previous*. Μόλις τελειώσουμε με τις απαιτούμενες ρυθμίσεις και προϋποθέσεις πατάμε το πλήκτρο *Next*.

6.9 Άδεια (Licence)

Το τρίτο βήμα της εγκατάστασης έχει να κάνει με την άδεια χρήσης του λογισμικού Joomla η οποία είναι η GNU General Public License (GPL). Η άδεια αυτή διέπει τα προγράμματα που ανήκουν στην κατηγορία «ελεύθερου λογισμικού – λογισμικού ανοικτού κώδικα (ΕΛ/ΛΑΚ)».

Αναλυτικότερα, οι κάτοχοι ενός τέτοιου προγράμματος μπορούν:

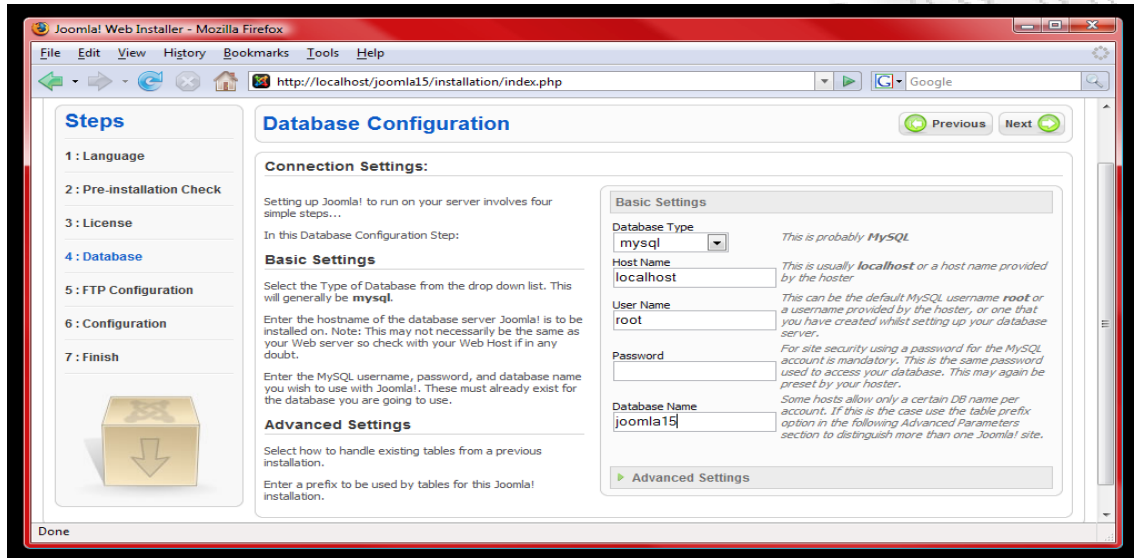
- Να τρέξουν το πρόγραμμα για οποιοδήποτε λόγο
- Να μελετήσουν τη λειτουργία του προγράμματος και να το τροποποιήσουν
- Να διανείμουν αντίγραφα του προγράμματος έτσι ώστε να βοηθήσουν τον πλησίον
- Να βελτιώσουν το πρόγραμμα και να προσφέρουν τις βελτιώσεις στο κοινό, έτσι ώστε να ωφεληθεί ολόκληρη η κοινότητα

Ο κώδικας σε προγράμματα τα οποία ακολουθούν την άδεια GPL είναι ανοιχτός και διαθέσιμος σε όλους. Επιπρόσθετα, απαγορεύεται η πώληση ενός προγράμματος με αυτή την άδεια. Τυχόν κέρδη, μπορούν να αποκομισθούν μέσω υπηρεσιών σχετικών με αυτό (π.χ. εκπαίδευση, εγκατάσταση, τροποποίηση) και εξόδων συσκευασίας. Για να προχωρήσουμε στο «Δημιουργία ιστότοπου με την χρήση cms»

επόμενο βήμα της εγκατάστασης, πατάμε στο κουμπί «Επόμενο» και με αυτή την κίνηση, συμφωνούμε στην αποδοχή της άδειας GNU/GPL.

6.10 Ρύθμιση της βάσης δεδομένων (Database Configuration)

Το επόμενο βήμα είναι ιδιαίτερα σημαντικό μιας και αφορά τις ρυθμίσεις της βάσης δεδομένων του συστήματος, όπως φαίνεται στην παρακάτω εικόνα:



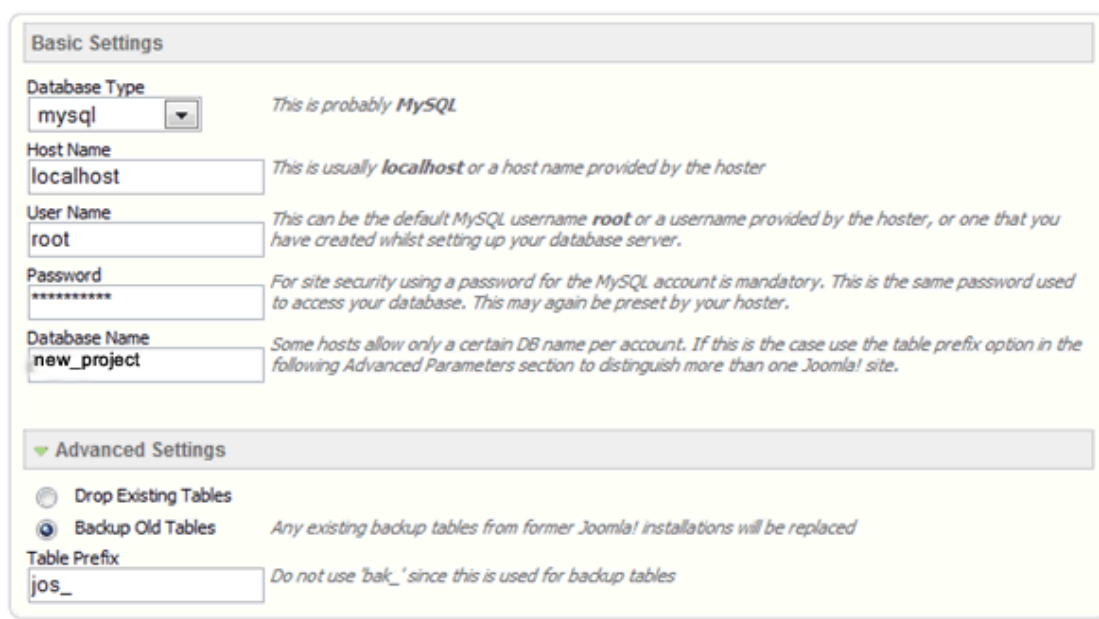
Εικόνα: Ρυθμίσεις βάσης δεδομένων

Στο σημείο αυτό εισάγουμε τα στοιχεία τα οποία έχουμε καταγράψει από την προηγούμενη διαδικασία δημιουργίας της βάσης δεδομένων στον εξυπηρετητή MySQL. Στο πρώτο drop-down box, επιλέγουμε ως είδος βάσης δεδομένων «mysql». Το όνομα του διακομιστή είναι το πλήρες όνομα του συστήματος ή μια διεύθυνση IP. Πιο συγκεκριμένα, εάν η βάση βρίσκεται στο ίδιο μηχάνημα με αυτό που βρίσκεται και ο εξυπηρετητής Apache τότε εισάγουμε «localhost» ή «127.0.0.1». Σε πραγματικά περιβάλλοντα δεν είναι σπάνιο, για λόγους ασφάλειας και απόδοσης να χρησιμοποιείται ξεχωριστός εξυπηρετητής για τις βάσεις δεδομένων. Στην περίπτωση αυτή, εισάγουμε το όνομα του μηχανήματος (π.χ. databaseserver.mydomain.gr) ή την IP διεύθυνσή που του έχει αποδοθεί.

Στη συνέχεια, εισάγουμε το όνομα χρήστη και τον κωδικό χρήστη που έχουμε δημιουργήσει προηγουμένως. Ξεν πρέπει να ξεχνάμε ότι ο χρήστης αυτός θα πρέπει να έχει πλήρη δικαιώματα πάνω στη βάση δεδομένων που θα χρησιμοποιήσουμε για τις ανάγκες του Joomla. Το τελευταίο πεδίο που πρέπει να συμπληρωθεί είναι αυτό του ονόματος της βάσης δεδομένων MySQL.

Οι προχωρημένες ρυθμίσεις, δεν είναι απαραίτητο να συμπληρωθούν στην περίπτωση όπου έχουμε μια καινούργια εγκατάσταση του Joomla σε μια άδεια βάση δεδομένων. Στην περίπτωση όπου είχαμε μια προηγούμενη αποτυχημένη (ή και μη) εγκατάσταση τότε μπορούμε να επιλέξουμε τη λειτουργία «διαγραφή υπαρχόντων πινάκων» αντί της προεπιλογής «λήψης αντιγράφων ασφαλείας των υπαρχόντων πινάκων». Τέλος, μπορούμε να έχουμε πολλαπλές εγκαταστάσεις του Joomla στην ίδια βάση δεδομένων, αλλάζοντας απλά το πρόθεμα των πινάκων το οποίο κατά προεπιλογή είναι «jos_».

Μετά τη συμπλήρωση όλων των απαραίτητων στοιχείων η οθόνη ρύθμισης της βάσης δεδομένων θα έχει την παρακάτω μορφή:



The screenshot shows the 'Basic Settings' section of the Joomla! installation wizard. It includes the following fields and options:

- Database Type:** A dropdown menu set to 'mysql'. A note says 'This is probably MySQL'.
- Host Name:** A text input field containing 'localhost'. A note says 'This is usually localhost or a host name provided by the hoster'.
- User Name:** A text input field containing 'root'. A note says 'This can be the default MySQL username root or a username provided by the hoster, or one that you have created whilst setting up your database server.'
- Password:** A text input field with masked characters '*****'. A note says 'For site security using a password for the MySQL account is mandatory. This is the same password used to access your database. This may again be preset by your hoster.'
- Database Name:** A text input field containing 'new_project'. A note says 'Some hosts allow only a certain DB name per account. If this is the case use the table prefix option in the following Advanced Parameters section to distinguish more than one Joomla! site.'

Below the basic settings is the 'Advanced Settings' section, which is currently collapsed. It contains:

- Drop Existing Tables:** A radio button that is not selected.
- Backup Old Tables:** A radio button that is selected. A note says 'Any existing backup tables from former Joomla! installations will be replaced'.
- Table Prefix:** A text input field containing 'jos_'. A note says 'Do not use 'bak_' since this is used for backup tables'.

Εικόνα: Τελική οθόνη ρυθμίσεων βάσης δεδομένων

Αφού έχουμε ολοκληρώσει τις ρυθμίσεις προχωράμε στο επόμενο βήμα πατώντας του κουμπι «Next».

6.11 Ρύθμιση FTP (FTP Configuration)

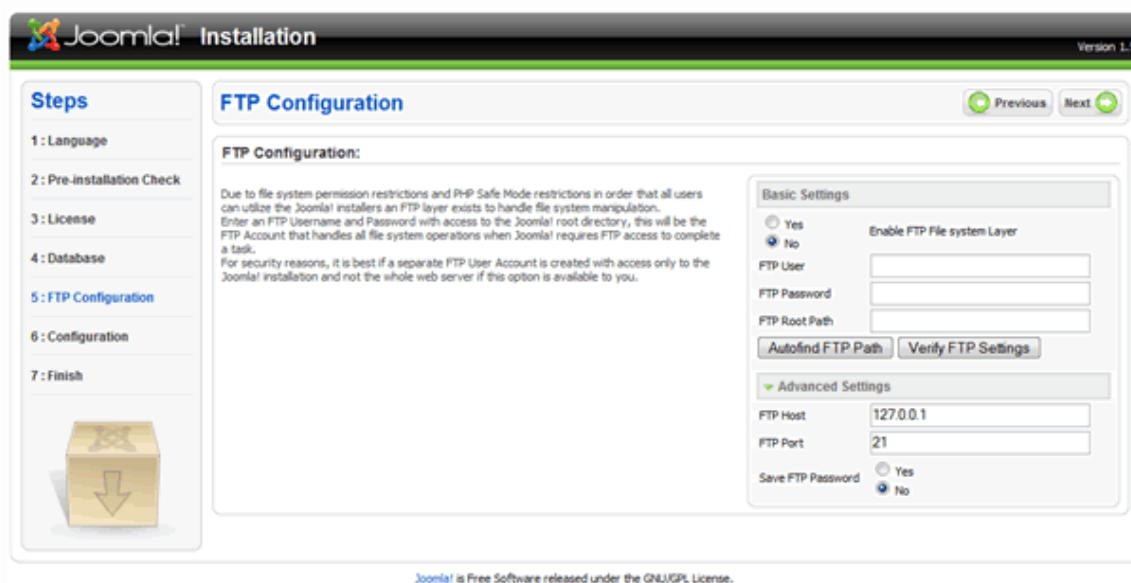
Το πέμπτο βήμα αφορά εξυπηρετητές Unix και αποτελεί το επίπεδο FTP. Αν έχουμε εξυπηρετητή Apache σε Windows μπορούμε να το αγνοήσουμε και να προχωρήσουμε στο επόμενο βήμα. Το επίπεδο FTP αποτελεί καινούργιο χαρακτηριστικό στην έκδοση 1.5 του Joomla. Πρακτικά, δίνει λύση σε προβλήματα ασφαλείας που παρουσιάζονται σε εξυπηρετητές Apache όπου για να λειτουργήσει σωστά το Joomla, επιβάλλεται μια σειρά από υποφακέλους αρχείων να είναι εγγράψιμοι από το χρήστη www-data που αντιστοιχεί στον εξυπηρετητή. Αυτό, εισάγει προβλήματα ασφαλείας από την άποψη ότι στην περίπτωση όπου τρέχουν πολλαπλές ιστοσελίδες στο ίδιο σύστημα, στην περίπτωση όπου καταφέρει κάποιος κακόβουλος χρήστης να αποκτήσει τον έλεγχο σε μία από αυτές, έχει τη δυνατότητα να κάνει το ίδιο και σε όλες τις υπόλοιπες. Εναλλακτικό του επιπέδου FTP στους εξυπηρετητές Unix, είναι η εγκατάσταση του module suexec ή suhrp, όπου η γλώσσα PHP εκτελείται όχι με προνόμια του χρήστη εξυπηρετητή αλλά με τα προνόμια του χρήστη του φακέλου του οποίου είναι τοποθετημένα τα αρχεία, παρέχοντας έτσι απομόνωση μεταξύ πολλαπλών εγκαταστάσεων στο ίδιο σύστημα.

Όπως φαίνεται και στην παρακάτω εικόνα, για να κάνουμε χρήση του επιπέδου FTP χρειαζόμαστε τα παρακάτω στοιχεία:

Όνομα εξυπηρετητή (συνήθως localhost ή 127.0.0.1)

Όνομα χρήστη και κωδικός χρήστη

Κεντρικός φάκελος FTP (είναι η διαδρομή που έχουμε εγκαταστήσει το Joomla όπως αυτή φαίνεται από τον FTP server, τυπικά : /var/www/joomla)



Εικόνα: Οθόνη ρύθμισης FTP

Για να ενεργοποιήσουμε το επίπεδο FTP, επιλέγουμε «Ναι» και εισάγουμε τα στοιχεία που συλλέξαμε παραπάνω. Είναι χρήσιμο προτού προχωρήσουμε στο επόμενο βήμα, να πατήσουμε το κουμπί «Επιβεβαίωση Ρυθμίσεων FTP», ούτως ώστε να ελέγξουμε ότι τα στοιχεία τα οποία έχουμε εισάγει είναι σωστά και κατά συνέπεια ότι λειτουργεί το επίπεδο FTP σωστά. Σημειώνεται ότι η ρύθμιση του στρώματος FTP δεν κρίνεται απαραίτητη στην περίπτωση τοπικής εγκατάστασης, με εξαίρεση ίσως την εγκατάσταση σε τοπικό δίκτυο. Αφού έχουμε ολοκληρώσει τη διαδικασία προχωράμε στο προτελευταίο βήμα πατώντας «Next».

6.12 Κύρια Ρύθμιση (Main Configuration)

Το προτελευταίο βήμα είναι ιδιαίτερα σημαντικό μιας και σε αυτό γίνονται οι τελικές ρυθμίσεις που αφορούν τον δικτυακό μας τόπο. Όπως φαίνεται και στην εικόνα στο βήμα αυτό εισάγουμε τα στοιχεία που αφορούν το όνομα (τίτλο) της ιστοσελίδας, την ηλεκτρονική διεύθυνση e-mail καθώς και δύο πεδία στα οποία συμπληρώνουμε και επιβεβαιώνουμε τον κωδικό με τον οποίο θα έχουμε πρόσβαση στην ιστοθέση ως υπερδιαχειριστές, σε συνδυασμό πάντα με το εξορισμού όνομα χρήστη για τον υπερδιαχειριστή που είναι *admin*.

Εικόνα: Οθόνη βασικών ρυθμίσεων ιστότοπου

Το τελευταίο τμήμα σχετίζεται με τα δεδομένα τα οποία θέλουμε να εισάγουμε στη διαδικασία εγκατάστασης. Στην ουσία έχουμε τρεις επιλογές:

Να εισάγουμε τα εξ ορισμού δεδομένα δείγματος (default sample data) που περιέχονται μέσα στο πακέτο του Joomla!

Να "μεταναστεύσουμε" (migrate) δεδομένα από κάποια προηγούμενη έκδοση του Joomla, χρησιμοποιώντας τα διαθέσιμα "εργαλεία μετανάστευσης" (migration tools).

Να μην κάνουμε τίποτα από τα δύο και να ξεκινήσουμε μία καινούργια και άδεια εγκατάσταση του Joomla. Επιπρόσθετα, έχουμε τη δυνατότητα να φορτώσουμε υπάρχοντα δεδομένα από αντίγραφα ασφαλείας τα οποία έχουν δημιουργηθεί από κάποια άλλη ιστοθέση,

6.12.1 Εγκατάσταση προκαθορισμένων δεδομένων δείγματος

Τα προκαθορισμένα δεδομένα δείγματος (default sample data) είναι μία συλλογή από Τμήματα (Sections), Κατηγορίες (Categories), Άρθρα (Articles), Web Links, News Feeds και διάφορα άλλα δείγματα δεδομένων τα οποία έχουν δημιουργηθεί με σκοπό την επίδειξη των δυνατοτήτων που προσφέρει το Joomla!. Τα δεδομένα αυτά περιέχονται σε ένα αρχείο SQL μέσα στο πακέτο εγκατάστασης.

Εξ ορισμού η επιλογή *Install Default Sample Data* είναι ήδη μαρκαρισμένη, αλλά προκειμένου να εγκαταστήσουμε τα δεδομένα αυτά, πρέπει να κάνουμε κλικ στο πλήκτρο *Install Sample Data*. Στην περίπτωση επιτυχημένης εγκατάστασης το πλήκτρο αυτό μετατρέπεται σε ένα μήνυμα επιβεβαίωσης ότι τα δεδομένα εγκαταστάθηκαν με επιτυχία (*Sample Data Installed Successfully*).

6.12.2 Migration Tools

Η δυνατότητα εισαγωγής περιεχομένου από δικτυακό τόπο ο οποίος ήταν βασισμένος στην προηγούμενη έκδοση του Joomla (1.0). Οι δύο εκδόσεις (1.0 και 1.5) έχουν τεράστιες διαφορές από πλευράς αρχιτεκτονικής και δεν θεωρείται υπερβολή εάν αναφέρουμε ότι η έκδοση 1.5 είναι γραμμένη από την αρχή και έχει ελάχιστη σχέση από τεχνικής πλευράς με την προηγούμενη. Για να διευκολύνει, συνεπώς, τη διαδικασία αναβάθμισης από την μία έκδοση στην άλλη, η προγραμματιστική ομάδα του Joomla παρέχει μία εφαρμογή με όνομα com_migrator η οποία «Δημιουργία ιστότοπου με την χρήση cms»

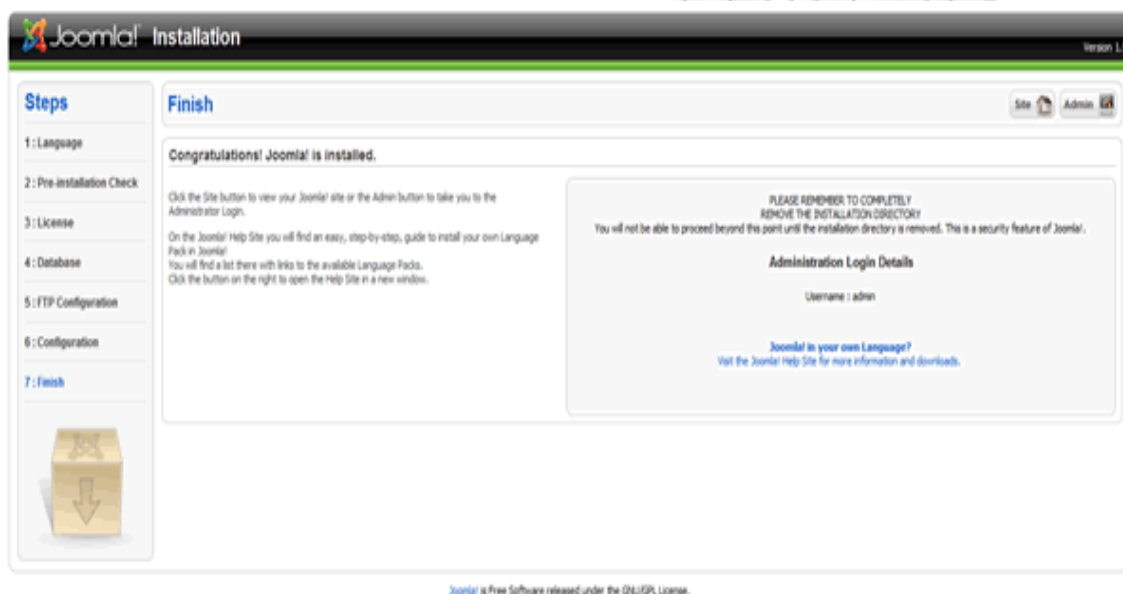
λαμβάνει ένα αντίγραφο των δεδομένων του παλιού ιστότοπου. Τα δεδομένα αυτά στη συνέχεια τα μετατρέπει σε μορφή συμβατή με την έκδοση 1.5. Στη συνέχεια, μπορούμε να εισάγουμε πλέον τα δεδομένα στο νέο δικτυακό τόπο. Περισσότερες πληροφορίες πάνω στη διαδικασία αυτή μπορούμε να βρούμε στη διεύθυνση:

[http://docs.joomla.org/Migrating from 1.0.x to 1.5 Stable](http://docs.joomla.org/Migrating_from_1.0.x_to_1.5_Stable)

Μετά από όλα αυτά είμαστε έτοιμοι να προχωρήσουμε στο επόμενο βήμα πατώντας το πλήκτρο Next.

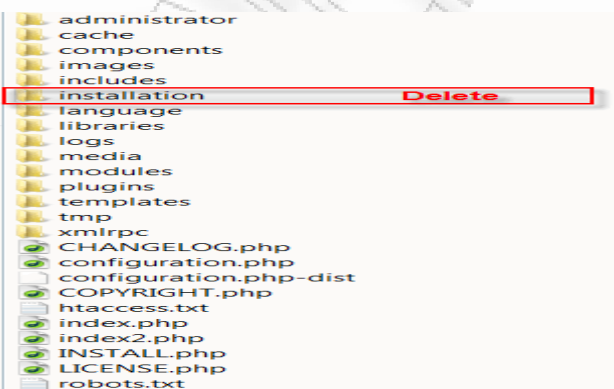
6.13 Τέλος (Finish)

Από τη στιγμή που φτάνουμε σε αυτή την οθόνη, σημαίνει ότι η εγκατάστασή μας έχει ολοκληρωθεί με επιτυχία και είμαστε πλέον έτοιμοι να προχωρήσουμε με τη διαχείριση της ιστοθέσης μας.



Εικόνα: Τελική οθόνη εγκατάστασης

Στο βήμα αυτό για να ολοκληρώσουμε τη διαδικασία εγκατάστασης επιβάλλεται να διαγράψουμε το φάκελο με όνομα installation από τον φάκελο εγκατάστασης του Joomla. Μόλις γίνει η διαγραφή είμαστε έτοιμοι να επισκεφτούμε την σελίδα διαχείρισης του ιστότοπου αλλά και τον ίδιο.



Εικόνα: Αφαίρεση του φακέλου Installation από την εγκατάσταση

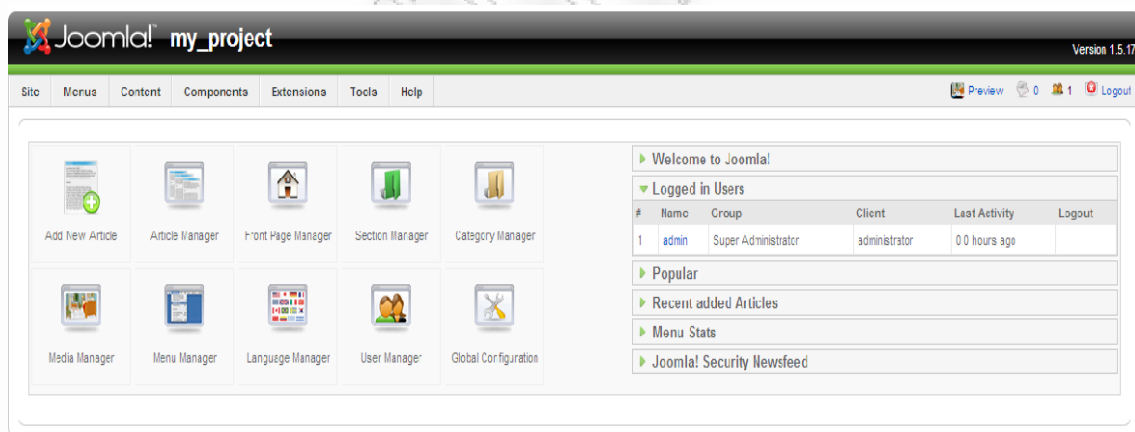
Μόλις σβήσουμε το συγκεκριμένο φάκελο, μπορούμε να επιλέξουμε ανάμεσα στα πλήκτρα Site (Ιστοθέση) ή Διαχείριση (Admin). Το πρώτο μας μεταφέρει στην αρχική σελίδα της ιστοθέσης μας (το λεγόμενο Front-end) με διεύθυνση http://localhost/my_project/ και το δεύτερο στη σελίδα διαχείρισης (Administration) του συστήματος (το λεγόμενο Back-end) με διεύθυνση http://localhost/my_project/administrator/.



Joomla! is Free Software released under the GNU/GPL License.

Εικόνα: Οθόνη σύνδεσης στο Back-End

Στην οθόνη αυτή συμπληρώνουμε το όνομα χρήστη *admin* και τον κωδικό πρόσβασης που εισάγαμε κατά τη διαδικασία της εγκατάστασης, και στη συνέχεια πατάμε το πλήκτρο *Login*.



Joomla! is Free Software released under the GNU/GPL License.

Εικόνα: Αρχική οθόνη Back-End

Μετά από όλα αυτά η εγκατάσταση έχει πλέον ολοκληρωθεί και είμαστε έτοιμοι να αναπτύξουμε και να διαχειριστούμε την ιστοθέση μας.

7 Εφαρμογή: Υλοποίηση ιστότοπου για εύρεση ξενοδοχείων με τη βοήθεια του Joomla.

Στην συγκεκριμένη ενότητα θα αναφερθούμε κατ' αρχήν σε μια γενική περιγραφή της ιδέας αυτής της εφαρμογής και σε ορισμένα ιδιαίτερα χαρακτηριστικά της. Κατόπιν θα περιγράψουμε το σχεδιασμό του ιστότοπου μέσα από τον πίνακα ελέγχου του Joomla και θα αναλύσουμε ορισμένα κομμάτια κώδικα και κάποιες παραμετροποιήσεις που έχουν γίνει.

7.1 Γενική περιγραφή εφαρμογής.

Ο στόχος του ιστότοπου είναι να παρουσιάζει ξενοδοχεία που βρίσκονται ανά την Ελλάδα και να δίνει πληροφορίες σχετικά με τις τιμές και τις παροχές που προσφέρουν και με τη βοήθεια των google maps, απεικονίζεται στο χάρτη το ακριβές σημείο που βρίσκεται το κάθε ξενοδοχείο. Επίσης παρέχονται κριτικές κυρίως για τα εστιατόρια των μεγάλων ξενοδοχείων της Αθήνας, πληροφορίες για ενοικιάσεις σκαφών- αυτοκινήτων ,για τα μέσα μεταφοράς και μέσω συνδέσμου (link) για τον καιρό που επικρατεί στην Ελλάδα. Επίσης υπάρχει η δυνατότητα για του εγγεγραμμένους χρήστες του ιστότοπου να μπορούν να λαμβάνουν μέρος σε δημοσκοπήσεις, να γράφουν τα σχόλια τους για τα ξενοδοχεία , να καταχωρούν το δικό τους άρθρο, να φορτώνουν φωτογραφίες βίντεο ή αρχεία Ms Office και τέλος να μπορούν να αλλάξουν τα χρώματα του ιστότοπου όπως ακριβώς αρέσει στον καθένα. Ιδίως οι χρήστες που έχουν παραπάνω δικαιώματα θα είναι σε θέση να αλλάξουν ακόμη και το περιεχόμενο του ιστότοπου. Τέλος με την βοήθεια του RSS Feed έχουμε στον δικό μας ιστότοπο πληροφορίες για θέματα που παρουσιάζονται από άλλους χώρους του διαδικτύου.

7.2 Ιδιαίτερα χαρακτηριστικά του Joomla.

Σύμφωνα με τη δομή περιεχομένου του Joomla! είναι αναγκαίο να καθοριστούν οι Ενότητες και οι Κατηγορίες περιεχομένου. Ενδεικτικά για τη συγκεκριμένη περίπτωση επιλέγονται οι εξής ενότητες και κατηγορίες:

- Αρχική
 - Ξενοδοχεία
 - Αθήνα
 - Πειραιάς
 - Δωμάτια
 - Κυκλάδες
 - Δωδεκάνησα
 - Ενοικιάσεις
 - Αυτοκινήτων
 - Ιστιοπλοικών
 - Μέσα μεταφοράς
 - Δελτίο καιρού
 - Web links
 - Κριτικές
-

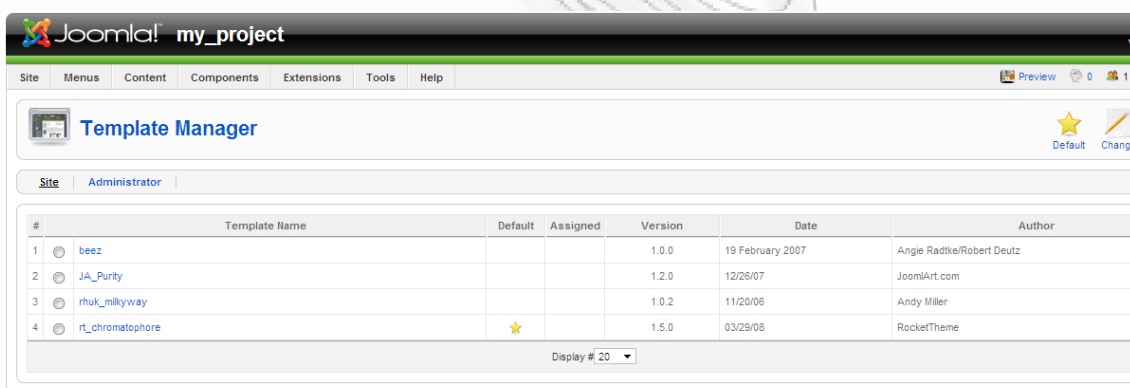
Το περιεχόμενο των ενότητων και κατηγοριών αποτελείται από άρθρα αποτελούμενα από κείμενο και εικόνες. Τα άρθρα με τη σειρά τους οργανώνονται σε κατηγορίες. Ένα άρθρο μπορεί να ανήκει σε μία ή καμία κατηγορία (μη κατηγοριοποιημένο). Με την ίδια λογική οι ενότητες περιέχουν μία ή περισσότερες κατηγορίες. Με τον τρόπο αυτό έχουμε μια δένδροειδή δομή οργάνωσης η οποία με τρόπο απλό και κατανοητό δίνει τη δυνατότητα, ταξινόμησης της πληροφορίας για δικτυακούς τόπους που περιέχουν από μερικές έως χιλιάδες σελίδες.

Επιπλέον όπως γνωρίζουμε το Joomla στηρίζεται στις επεκτάσεις του τις οποίες αποτελούν, τα components, τα modules και τα plugins. Σε αυτά λοιπόν βασίζονται τόσο το οριζόντιο και το κύριο μενού όσο και οι δημοσκοπήσεις, τα τελευταία νέα, τα RSS η φόρμα log in και πολλές άλλες επεκτάσεις που έχουμε χρησιμοποιήσει και θα αναφερθούμε σε κάποιες από αυτές σε επόμενες ενότητες.

7.3 Εγκατάσταση και παραμετροποίηση προτύπου (template)

Το Joomla δίνει την δυνατότητα να χρησιμοποιήσουμε κάποια από τα προεγκατεστημένα template που παρέχει είτε να κατεβάσουμε ελεύθερα η επί πληρωμή διάφορα είδη από την παγκόσμια κοινότητα του Joomla. Αφού έχουμε επιλέξει το πρότυπο παρουσίασης, θα πρέπει να το εγκαταστήσουμε και να παραμετροποιηθεί πριν από την προσθήκη του περιεχομένου. Η εγκατάσταση του προτύπου γίνεται από το σύστημα διαχείρισης του Joomla! με την επιλογή Extensions και Install/Uninstall. Επιλέγουμε (Browse..) το συμπιεσμένο αρχείο που περιέχει το πρότυπο από το σκληρό δίσκο μας, στην προκειμένη περίπτωση το αρχείο με όνομα rt_chromatophore_j15.tgz και κατόπιν Upload File & Install.

Από το βασικό μενού διαχείρισης κατόπιν επιλέγουμε τη λειτουργία Extensions και Template Manager και εμφανίζεται η λίστα με τα εγκατεστημένα πρότυπα παρουσίασης. Πατάμε το στρογγυλό πεδίο δίπλα στο όνομα του προτύπου rt_chromatophore που εγκαταστήσαμε και κατόπιν το κουμπί Default ώστε το πρότυπο αυτό να είναι το προκαθορισμένο, όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: Οθόνη επιλογής προτύπου.

Ακολουθεί η παραμετροποίηση του προτύπου κλικάροντας πάνω στο όνομα του προτύπου rt_chromatophore από τη λίστα (Extensions, Template Manager). Κάθε πρότυπο διαθέτει τις δικές του παραμέτρους, όπως τις έχει καθορίσει και υλοποιήσει ο κατασκευαστής του. Το συγκεκριμένο πρότυπο επιτρέπει τις εξής ρυθμίσεις:

- Πλάτος του template
- Πλάτος δεξιάς και αριστερής στήλης
- Ενεργοποίηση ή απενεργοποίηση επιλογή χρωμάτων. Το συγκεκριμένο πρότυπο μας παρέχει την δυνατότητα να αλλάζουμε τα χρώματα του ιστότοπού σύμφωνα με τις προτιμήσεις του κάθε χρήστη. Πατώντας το color chooser κατεβαίνει ένα πάνελ με μια παλέτα χρωμάτων κανουμε την επιλογή που θέλουμε και στη συνέχεια πατάμε apply colors και αποθηκεύονται τα χρώματα.
- Τύπος μενού διαθέτει 3 επιλογές για το πώς θα είναι το drop down menu.
- Επιλογή γραμματοσειράς και μέγεθος γραμματοσειράς
- Μέγιστος αριθμός ενθεμάτων (modules) ανά σειρά

Οι θέσεις που διαθέτει το συγκεκριμένο πρότυπο για να τοποθετηθούν τα modules φαίνονται στην παρακάτω εικόνα:

«Δημιουργία ιστότοπου με την χρήση cms»



Εικόνα: Θέσεις προτύπου.

Επειδή θα χρειαστούμε μια επιπλέον θέση στο template ώστε να τοποθετήσουμε ένα συγκεκριμένο module αναγκαζόμαστε να τη δημιουργήσουμε. Για να γίνει αυτό βέβαια θα πρέπει να επέμβουμε στον κώδικα του template και συγκεκριμένα σε δύο αρχεία το `templateDetails.xml` και το `index.php` που βρίσκονται στον κατάλογο `\xampp\htdocs\my_project\templates\rt_chromatophore`. Η καινούρια θέση θα βρίσκεται πάνω από τις θέσεις `advert 1`, `advert 2`, `advert 3`, `advert 4`, `advert 5` και θα έχει όνομα `home_gallery`.

Ανοίγουμε το αρχείο `templateDetails.xml` με το notepad ή οποιονδήποτε άλλο editor χρησιμοποιεί ο καθένας και πηγαίνουμε στην γραμμή 65 και τοποθετούμε τον κώδικα `<position>home_gallery</position>` εν συνεχεία ανοίγουμε το αρχείο `index.php` και στην γραμμή 132 κάτω από το `<div id="maincol2">` γράφουμε `<jdoc:include type="modules" name="home_gallery" style="none" />`. Η δήλωση `jdoc:include` είναι μία μέθοδο για τα template του Joomla όπου εμφανίζει το περιεχόμενο της σελίδας που προβάλλεται. Υπάρχουν πολλών ειδών τέτοιες δηλώσεις η κάθε μια όμως επιστρέφει ένα διαφορετικό μέρος από την σελίδα ενός ιστότοπου με Joomla. Η συγκεκριμένη αναφέρεται σε τύπο module με όνομα `home_gallery` και χωρίς κάποιο στυλ.

Ένας άλλος τρόπος που θα μπορούσαμε να προσθέσουμε αυτή την γραμμή κώδικα είναι να πάμε στον πίνακα ελέγχου διαχειριστή του Joomla πηγαίνουμε στην επιλογή `extensions->template manager` όπου και εμφανίζονται όλα τα πρότυπα. Πατάμε πάνω στο `rt_chromatophore` και στην επόμενη σελίδα που θα ανοίξει επιλέγουμε το κουμπί `edit html` που ουσιαστικά μας ανοίγει το αρχείο `index.php` αλλά μέσα από το interface του Joomla.

Τέλος για να προσθέσουμε την εικόνα που θα έχει η κεντρική σελίδα του ιστότοπου μας το μόνο που κάνουμε είναι να βάλουμε την φωτογραφία που θέλουμε στο φάκελο `template/images` και στη συνέχεια στο αρχείο `index.php` στη γραμμή 90 να βάλουμε το όνομα της φωτογραφίας `<a href="<?php echo $this->baseurl; ?>" class="nounder">`.

8 Δημιουργία Περιεχομένου

8.1 Διαχείριση ενότητων

Η ενότητα (Section) είναι η ανώτερη βαθμίδα ταξινόμησης στο Joomla. Για να προσθέσουμε μια νέα ενότητα αρκεί να μπούμε στη σελίδα διαχείρισης και στη συνέχεια να πατήσουμε από τη λίστα εικονιδίων το “Section Manager”. Στη συνέχεια στη σελίδα που μας εμφανίζεται πατάμε το εικονίδιο “New”. Στο σημείο αυτό εμφανίζεται η παρακάτω εικόνα με μια σειρά από επιλογές.

Εικόνα: Δημιουργία νέας ενότητας.

Οι επιλογές που θα πρέπει να συμπληρώσουμε είναι:

- Ο τίτλος της ενότητας (Ξενοδοχεία)
- Ο εναλλακτικός τίτλος της ενότητας (Alias).
- Δημοσίευση ενότητας (Yes: Δημοσιευμένη, No: Μη δημοσιευμένη / offline)
- Προτεραιότητα ενότητας (εφόσον υπάρχουν πολλές, η σειρά με την οποία θα εμφανίζεται)
- Επίπεδο πρόσβασης. Καθορίζει την πρόσβαση που θα παρέχει ανάλογα με τους χρήστες. Public : Όλοι οι χρήστες, Registered: Όλοι οι εγγεγραμμένοι χρήστες, Special: Όσοι χρήστες έχουν προνόμια special.
- Γενική φωτογραφία της ενότητας.
- Τοποθέτηση της φωτογραφίας σε σχέση με τον τίτλο.
- Κείμενο περιγραφής της ενότητας.

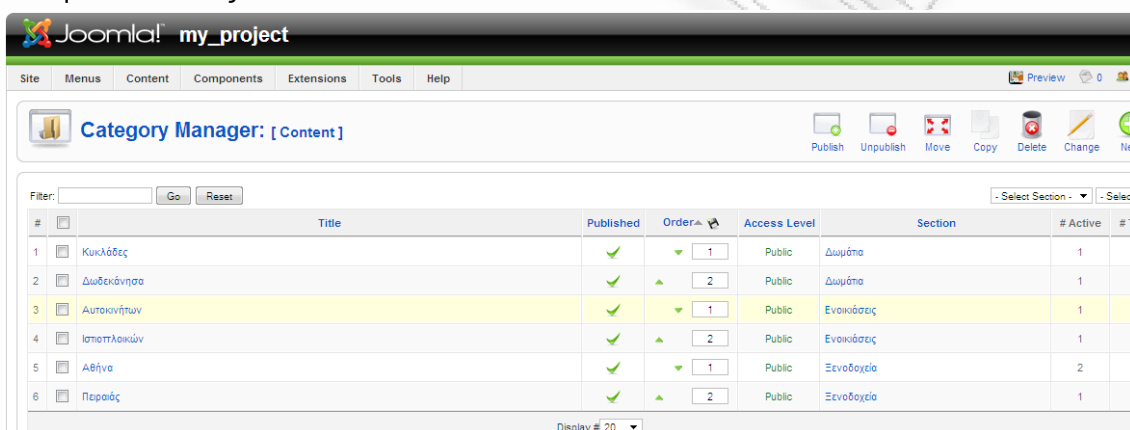
Αφού συμπληρώσουμε τα παραπάνω πεδία, αποθηκεύουμε τις αλλαγές πατώντας πάνω δεξιά το εικονίδιο «Save». Όταν αποθηκεύσουμε την ενότητα, επιστρέφουμε στη σελίδα διαχείρισης ενότητων. Εκεί έχουμε τη δυνατότητα για τα παρακάτω:

- Ενεργοποίηση – Απενεργοποίηση ενότητας (Publish / Unpublish)
- Αντιγραφή κατηγοριών/άρθρων (Copy – Αντιγράφει τις κατηγορίες και τα άρθρα που βρίσκονται στη συγκεκριμένη ενότητα σε άλλη)
- Διαγραφή ενότητας (Delete – Για να διαγραφεί μια ενότητα πρέπει να μην περιέχει ούτε κατηγορίες, ούτε άρθρα.)
- Αλλαγή στοιχείων ενότητας (Change)
- Προσθήκη νέας (New)
- Λήψη βοήθειας (Help)

Για να προβούμε στις παραπάνω ενέργειες επιλέγουμε μέσω των checkboxes αριστερά από τα ονόματα, τις ενότητες που μας ενδιαφέρουν και κάνουμε κλικ στο αντίστοιχο εικονίδιο. Κατά τον ίδιο ακριβώς τρόπο δημιουργούμε και τις υπόλοιπες ενότητες του ιστοτοπού μας που είναι τα Δωμάτια και οι Ενοικιάσεις.

8.2 Διαχείριση κατηγοριών

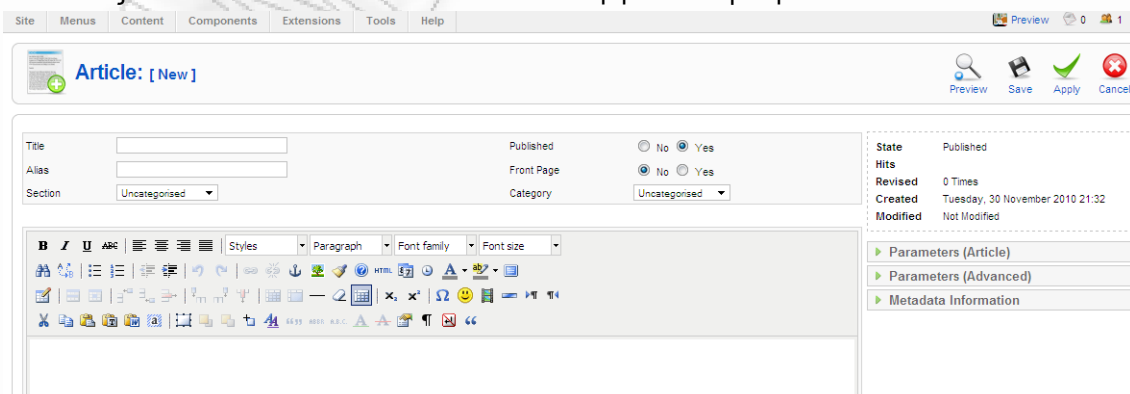
Ιεραρχικά, η κατηγορία (Category) είναι η ενδιάμεση βαθμίδα ταξινόμησης στο Joomla. Για να προσθέσουμε μια νέα κατηγορία αρκεί να μπούμε στη σελίδα διαχείρισης και στη συνέχεια να πατήσουμε από τη λίστα εικονιδίων το “Category Manager”. Στη συνέχεια στη σελίδα που μας εμφανίζεται πατάμε το εικονίδιο “New”. Στο σημείο αυτό εμφανίζεται μια εικόνα με μια σειρά από επιλογές που είναι ακριβώς ίδιες με αυτές των ενότητων με την επιπλέον διαφορά ότι περιέχει ακόμη μία επιλογή που είναι η ενότητα στην οποία ανήκει η κατηγορία. Δημιουργούμε μία προς μία τις κατηγορίες που έχουμε σχεδιάσει, προσέχοντας ώστε να ανήκουν στις σωστές ενότητες. Αυτές είναι Αθήνα, Πειραιάς που ανήκουν στην ενότητα Ξενοδοχεία. Κυκλάδες, Δωδεκάνησα που ανήκουν στην κατηγορία Δωμάτια και Αυτοκινήτων Ιστιοπλοϊκών που ανήκουν στην ενότητα Ενοικιάσεις.



Εικόνα: Διαχείριση κατηγοριών.

8.3 Διαχείριση άρθρων.

Τα άρθρα είναι το κατώτερο επίπεδο οργάνωσης και ιεράρχησης της πληροφορίας στο Joomla. Όπως αναφέρθηκε και προηγουμένως, τα άρθρα αποτελούνται από μία ή περισσότερες σελίδες. Για να προσθέσουμε ένα νέο άρθρο αρκεί να μπούμε στη σελίδα διαχείρισης και στη συνέχεια να πατήσουμε από τη λίστα εικονιδίων το “Article Manager”. Στη συνέχεια στη σελίδα που μας εμφανίζεται πατάμε το εικονίδιο “New”. Εναλλακτικά μπορούσαμε να πατήσουμε απευθείας στο εικονίδιο «Add new article» και να εμφανιστεί η παρακάτω εικόνα.



Εικόνα: Βασικές επιλογές άρθρου.

«Δημιουργία ιστοτόπου με την χρήση cms»

Οι επιλογές είναι οργανωμένες σε δύο κύρια πλαίσια, αριστερά και δεξιά. Στο αριστερό μέρος υπάρχουν οι εξής ιδιότητες.

- Title: Τίτλος του άρθρου.
- Alias: Εναλλακτικός τίτλος του άρθρου.
- Section: Η ενότητα στην οποία ανήκει το άρθρο.
- Published: Δημοσίευση άρθρου (Yes: Δημοσιευμένο, No: Μη δημοσιευμένο / offline)
- Front Page: Εμφάνιση στην πρώτη σελίδα του Joomla
- Category: Η κατηγορία στην οποία ανήκει το άρθρο
- Ακριβώς από κάτω από τις επιλογές υπάρχει WYSIWYG editor στον οποίο μπορούμε να εισάγουμε περιεχόμενο και να μετατραπεί αυτόματα σε γλώσσα HTML.

Στα δεξιά υπάρχει ένας πίνακας στατιστικών όπου αναφέρει τα παρακάτω στοιχεία:

- State: κατάσταση άρθρου (published : δημοσιευμένο, unpublished: offline)
- Hits: αριθμός επισκέψεων στο συγκεκριμένο άρθρο
- Revised: φορές που έχει διορθωθεί το άρθρο
- Created: ημερομηνία δημιουργίας του άρθρου
- Modified: εάν έχουν γίνει αλλαγές σε σχέση με το αρχικό άρθρο

Ακριβώς κάτω από τον πίνακα στατιστικών υπάρχουν 3 πλαίσια επιλογών Γενικές παράμετροι άρθρου Parameters (Article), προχωρημένες παράμετροι άρθρου Parameters (Advanced), μεταπληροφορίες (Metadata Information). Στις βασικές παραμέτρους του άρθρου μπορούμε να ρυθμίσουμε να φαίνονται τα εξής:

- Author: Ο συγγραφέας του άρθρου.
- Author Alias: Το εναλλακτικό όνομα του συγγραφέα του άρθρου.
- Access Level: Επίπεδο πρόσβασης. Καθορίζει την πρόσβαση που θα παρέχει ανάλογα με τους χρήστες. Public : Όλοι οι χρήστες, Registered: Όλοι οι εγγεγραμμένοι χρήστες, Special: Όσοι χρήστες έχουν προνόμια special.
- Start Publishing Η ημερομηνία και ώρα στην οποία το άρθρο θα είναι διαθέσιμο στους επισκέπτες
- Finish Publishing Η ημερομηνία και ώρα στην οποία το άρθρο θα πάψει να είναι διαθέσιμο στους επισκέπτες

Στη συνέχεια καθορίζονται οι προχωρημένες παράμετροι του άρθρου που μας επιτρέπουν να καθορίσουμε τα εξής:

- Show Title: Εμφάνιση ή όχι του τίτλου του άρθρου
- Title Linkable: Να εμφανίζεται ή όχι ο τίτλος του άρθρου με τη μορφή web link
- Intro Text: Να εμφανίζεται ή όχι όταν κάνει κάποιος κλικ στην προεπισκόπηση ενός άρθρου η πρώτη του σελίδα
- Section Name: Να εμφανίζεται ή όχι η ενότητα στην οποία ανήκει ένα άρθρο
- Section Title Linkable: Να εμφανίζεται ή όχι η ενότητα στην οποία ανήκει ένα άρθρο με τη μορφή web link.
- Category Name: Να εμφανίζεται ή όχι η κατηγορία στην οποία ανήκει ένα άρθρο
- Category Title Linkable: Να εμφανίζεται ή όχι η κατηγορία στην οποία ανήκει ένα άρθρο με τη μορφή web link
- Article Rating: Να επιτρέπεται ή όχι, οι επισκέπτες να βαθμολογούν ένα άρθρο
- Author Name: Να εμφανίζεται ή όχι ο συγγραφέας του άρθρου

- Created Date and Time: Να εμφανίζεται ή όχι, η ημερομηνία και ώρα που γράφτηκε ένα άρθρο
- Modified Date and Time: Να εμφανίζεται ή όχι, η ημερομηνία και ώρα που διορθώθηκε ένα άρθρο
- PDF Icon: Να εμφανίζεται ή όχι, εικονίδιο για μετατροπή του άρθρου σε αρχείο PDF
- Print Icon: Να εμφανίζεται ή όχι, εικονίδιο για εκτύπωση του άρθρου
- E-mail Icon: Να εμφανίζεται ή όχι, εικονίδιο για αποστολή του άρθρου σε e-mail
- Content Language: Επιλογή της γλώσσας στην οποία είναι γραμμένο το άρθρο
- Alternative Read More text: Όταν το άρθρο εμφανίζεται σε μία λίστα (π.χ. λίστα άρθρων κατηγορίας) τότε εμφανίζεται μόνο η πρώτη του σελίδα μαζί με ένα σύνδεσμο που μας οδηγεί στην πλήρη έκδοση. Η επιλογή αυτή μας επιτρέπει να δώσουμε το δικό μας κείμενο για αυτόν τον σύνδεσμο αντί του «Διαβάστε περισσότερα».

Αφού συμπληρώσουμε τα παραπάνω πεδία, αποθηκεύουμε τις αλλαγές πατώντας πάνω δεξιά το εικονίδιο «Save». Όταν αποθηκεύσουμε το άρθρο, επιστρέφουμε στη σελίδα διαχείρισης άρθρων. Εκεί έχουμε τη δυνατότητα για τα παρακάτω:

- Αρχαιοθήκη – Αποαρχαιοθήκη (Archive / Unarchive Τοποθέτηση άρθρου στο αρχείο και αφαίρεση αντίστοιχα)
- Ενεργοποίηση – Απενεργοποίηση άρθρου/ων (Publish / Unpublish)
- Μετακίνηση άρθρων (Move – Μετακινεί το άρθρο σε άλλη ενότητα)
- Αντιγραφή άρθρων (Copy – Αντιγράφει την κατηγορία και τα άρθρα που βρίσκονται στη συγκεκριμένη κατηγορία σε άλλη ενότητα)
- Διαγραφή άρθρου (Trash)
- Διόρθωση άρθρου (Edit)
- Προσθήκη νέου άρθρου (Add)
- Παράμετροι (Parameters)
- Λήψη βοήθειας (Help)

Για να προβούμε στις παραπάνω ενέργειες επιλέγουμε μέσω των checkboxes αριστερά από τα ονόματα, τις κατηγορίες που μας ενδιαφέρουν και κάνουμε κλικ στο αντίστοιχο εικονίδιο. Στη σελίδα διαχείρισης άρθρων που φαίνεται στην παρακάτω εικόνα έχουμε τη δυνατότητα και για ορισμένες άλλες ενέργειες. Πιο συγκεκριμένα, εάν κάνουμε κλικ με το ποντίκι στο εικονίδιο του άρθρου στη στήλη «Published» το δημοσιεύουμε ή το από δημοσιεύουμε. Αντίστοιχα, εάν κάνουμε κλικ στο εικονίδιο στη στήλη «Front Page», ενεργοποιούμε ή απενεργοποιούμε την εμφάνιση του άρθρου στην πρώτη σελίδα. Επιπρόσθετα, στη στήλη «Order» έχουμε τη δυνατότητα να αλλάξουμε τη σειρά με την οποία θα εμφανίζονται τα άρθρα εάν έχουμε επιλέξει για αυτά προβολή κατά σειρά. Κάνοντας κλικ στη στήλη «Access Level» ενός άρθρου τροποποιούμε τις επιλογές εμφάνισής του σε (public, registered, special). Αν κάνουμε κλικ στα ονόματα ενότητων ή κατηγοριών στις στήλες «Section» και «Category» οδηγούμαστε στις αντίστοιχες σελίδες διαχείρισης. Τα ονόματα συγγραφέα που εμφανίζονται στη στήλη «Author» μας οδηγούν στην σελίδα ρυθμίσεων για το συγγραφέα του άρθρου. Οι τρεις τελευταίες στήλες «Date», «Hits», «ID» μας ενημερώνουν για την ημερομηνία συγγραφής του άρθρου, τον αριθμό επισκέψεων και τον αύξοντα αριθμό αυτού, αντίστοιχα. Εάν κάνουμε κλικ στα ονόματα των στηλών τότε ταξινομούμε τα άρθρα σύμφωνα με τις αντίστοιχες ιδιότητες των στηλών. Επιπρόσθετα, παρέχεται φίλτρο αναζήτησης (Filter) όπου μπορούμε να εμφανίσουμε μόνο τα άρθρα των οποίων οι τίτλοι περιέχουν το κείμενο αναζήτησης της επιθυμίας μας. Παράλληλα, προσφέρονται φίλτρα ενότητων, κατηγοριών, συγγραφέα και κατάστασης, όπου μπορούμε να κάνουμε οποιοδήποτε συνδυασμό. Για παράδειγμα μπορούμε να αναζητήσουμε όλα τα δημοσιευμένα άρθρα του, τα οποία ανήκουν στην ενότητα «Ξενοδοχεία», στην κατηγορία «Αθήνα» και τα έχει γράψει ο χρήστης «Administrator».

#	Title	Published	Front Page	Order	Access Level	Section	Category	Author	Date	Hits	ID
1	Μέσα Μαρίνης Μεταφοράς Αθήνα	✓	✗	2	Public			Administrator	22.06.10	16	66
2	ΚΥΡΙΑΚΟΣ STUDIOΣ - ΑΦΑΝΤΟΥ ΡΟΔΟΣ	✓	✗	1	Public	Δωμάτιο	Δωδεκάνησo	Administrator	28.06.10	4	67
3	Athina Studios Paros	✓	✗	1	Public	Δωμάτιο	Κυκλάδες	Administrator	18.06.10	16	63
4	Ace rent a car	✓	✗	1	Public	Ενοικίασoς	Αυτοκινήτων	Administrator	22.06.10	0	65
5	JEANNEAU Sun Odyssey 54 DS	✓	✗	1	Public	Ενοικίασoς	Ισποπλοικών	Administrator	22.06.10	4	64
6	Hotel Ledra Marriott Athens	✓	✓	1	Public	Ξενοδοχεία	Αθήνα	Administrator	18.06.10	38	62
7	Best Western Esperia Palace Athens	✓	✓	2	Public	Ξενοδοχεία	Αθήνα	Administrator	16.06.10	17	59
8	Castella Hotel Piraeus	✓	✓	1	Public	Ξενοδοχεία	Πειραιάς	Administrator	18.06.10	17	61

Εικόνα: Διαχείριση άρθρων.

9 Διαχείριση μενού.

Τα μενού στο Joomla είναι ένα από τα βασικότερα δομικά στοιχεία οργάνωσης και παρουσίασης του περιεχομένου. Ένας ιστότοπος βασισμένος στο Joomla μπορεί να έχει ένα ή περισσότερα μενού τοποθετημένα σε διάφορα σημεία του. Παράλληλα, τα μενού έχουν αυθαίρετα μεγάλο αριθμό επιλογών, οι οποίες μπορούν να οργανωθούν σε αυθαίρετο βάθος, προσφέροντας έτσι τη δυνατότητα δημιουργίας εξαιρετικά πολύπλοκων δενδροειδών δομών. Τα μενού παρέχουν τη διεπαφή με όλες σχεδόν τις λειτουργίες του front-end του Joomla, οι οποίες δεν υπάρχει δυνατότητα να εμφανιστούν με διαφορετικό τρόπο. Αναλυτικότερα, ένα στοιχείο μενού μπορεί να είναι:

Εσωτερικοί σύνδεσμοι προς:

Άρθρα (Articles)

Επαφές (Contacts)

Ροές δεδομένων (News Feeds)

Δημοσκοπήσεις (Polls)

Αναζήτηση (Search)

Εφαρμογές εισόδου/εξόδου χρήστη (User)

Συνδέσμους προς άλλες σελίδες (Web Links)

Περιτύλιγμα (Wrapper)

Εξωτερικοί σύνδεσμοι

Διαχωριστικό

Σύνδεσμος προς άλλο υπάρχον στοιχείο μενού

9.1 Δημιουργία και παραμετροποίηση μενού.

Έχοντας δημιουργήσει τις ενότητες περιεχομένου, μπορούμε να δημιουργήσουμε μενού με συνδέσμους που θα εμφανίζονται σε όλες τις σελίδες του ιστότοπου και θα οδηγούν σε κάθε ενότητα. Τα μενού για το Joomla! είναι ενθέματα και με την εγκατάσταση του Joomla! δημιουργείται αυτόματα ένα ένθεμα που ονομάζεται Main Menu. Το ένθεμα αυτό είναι διαχειρίσιμο με 2 τρόπους:

- Από την κεντρική σελίδα διαχείρισης, επιλέγοντας Menus και Menu Manager, όπου εμφανίζεται η λίστα με όλα τα διαθέσιμα μενού. Στη σελίδα αυτή μπορεί να μεταβληθεί το όνομα του μενού και τα στοιχεία από τα οποία αποτελείται:

The screenshot shows the Joomla! administration interface for 'my_project'. The top navigation bar includes 'Site', 'Menus', 'Content', 'Components', 'Extensions', 'Tools', and 'Help'. The 'Menu Manager' section is active, displaying a table with the following data:

#		Title	Type	Menu Item(s)	# Published	# Unpublished
1	<input type="radio"/>	Main Menu	mainmenu		14	-

Εικόνα: Δημιουργία μενού.

- Από την κεντρική σελίδα διαχείρισης, επιλέγοντας Menus και Main Menu, όπου εμφανίζεται η λίστα με τα στοιχεία του κύριου μενού. Στη σελίδα αυτή μπορούν να προσθαφαιρεθούν στοιχεία στο κύριο μενού, το οποίο αρχικά αποτελείται από ένα μόνο στοιχείο – δηλ. έναν σύνδεσμο προς την αρχική σελίδα:

The screenshot shows the Joomla! administration interface for 'my_project' in the 'Menu Item Manager' for the 'mainmenu'. The interface includes a toolbar with icons for 'Menus', 'Default', 'Publish', 'Unpublish', 'Move', 'Copy', 'Trash', 'Change', and 'New'. Below the toolbar is a table of menu items:

#	<input type="checkbox"/>	Menu Item	Default	Published	Order	Access Level	Type
1	<input type="checkbox"/>	Αρχική	★	✓	14	Public	Articles » Front Page
2	<input type="checkbox"/>	Ξενοδοχεία		✓	15	Public	Articles » Section / Blog
3	<input type="checkbox"/>	↳ Αθήνα		✓	1	Public	Articles » Category / Blog
4	<input type="checkbox"/>	↳ Περαίες		✓	2	Public	Articles » Category / Blog
5	<input type="checkbox"/>	Δωμάτια		✓	16	Public	Articles » Section / Blog
6	<input type="checkbox"/>	↳ Κυκλάδες		✓	1	Public	Articles » Category / Blog
7	<input type="checkbox"/>	↳ Δωδεκάνησα		✓	2	Public	Articles » Category / Blog
8	<input type="checkbox"/>	Ενοικιάσεις		✓	17	Public	Articles » Section / Blog

Εικόνα: Επεξεργασία μενού.

Επιλέγουμε το δεύτερο τρόπο και καταρχήν αλλάζουμε τον τίτλο του συνδέσμου Home σε Αρχική. Κατόπιν πατάμε το κουμπί New για να προσθέσουμε στο μενού συνδέσμους προς τις ενότητες περιεχομένου. Το είδος του συνδέσμου που προσθέτουμε στο μενού είναι "Internal link", "Articles", δηλ. σύνδεσμος στο περιεχόμενο.

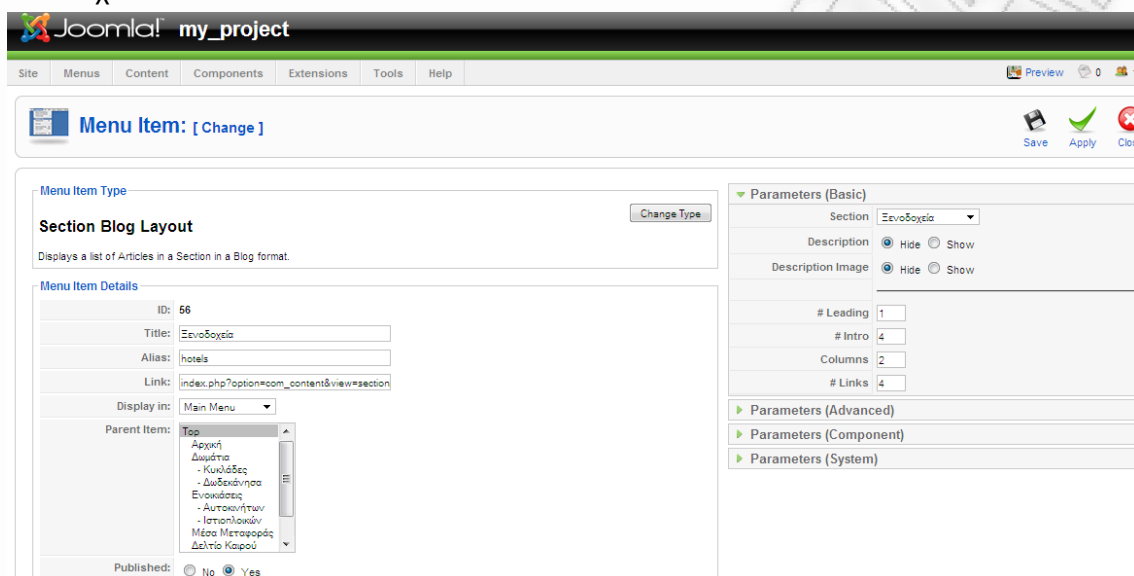
The screenshot shows the Joomla! administration interface for 'my_project' in the 'Menu Item Manager' for the 'mainmenu'. The 'Menu Item' is set to 'New'. Below the title bar is a tree view titled 'Select Menu Item Type' with the following items:

- Internal Link
- Articles
- Attachments
- Contacts
- Frontpage SlideShow
- Hotel Restaurant Reviews
- JComments
- News Feeds
- Polls
- Search
- User
- Web Links
- Wrapper
- External Link
- Separator
- Alias

Εικόνα: Επιλογή είδος συνδέσμου για το μενού.

«Δημιουργία ιστότοπου με την χρήση cms»

Στο επόμενο βήμα επιλέγουμε το είδος του περιεχομένου στο οποίο θα οδηγήει το μενού και επιλέγουμε “Section Blog Layout” για να εμφανίζονται όλα τα άρθρα της συγκεκριμένης ενότητας. Κατόπιν εισάγουμε τα στοιχεία του συνδέσμου και συγκεκριμένα τον τίτλο (Title), το ψευδώνυμο (alias), την ενότητα στην οποία οδηγεί (Section) και τον τρόπο που θα εμφανίζεται το περιεχόμενο της ενότητας. Σύμφωνα με το σχεδιασμό, ο χρήστης στη σελίδα κάθε ενότητας θα βλέπει τα 4 πιο πρόσφατα άρθρα σε 2 στήλες, οπότε στις βασικές παραμέτρους επιλέγουμε Leading:1, Intro:4, Columns:2 και Links: 4 ώστε να εμφανίζονται 4 σύνδεσμοι προς τα παλιότερα άρθρα. Το πρώτο στοιχείο μενού που δημιουργούμε είναι προς την ενότητα Ξενοδοχεία.



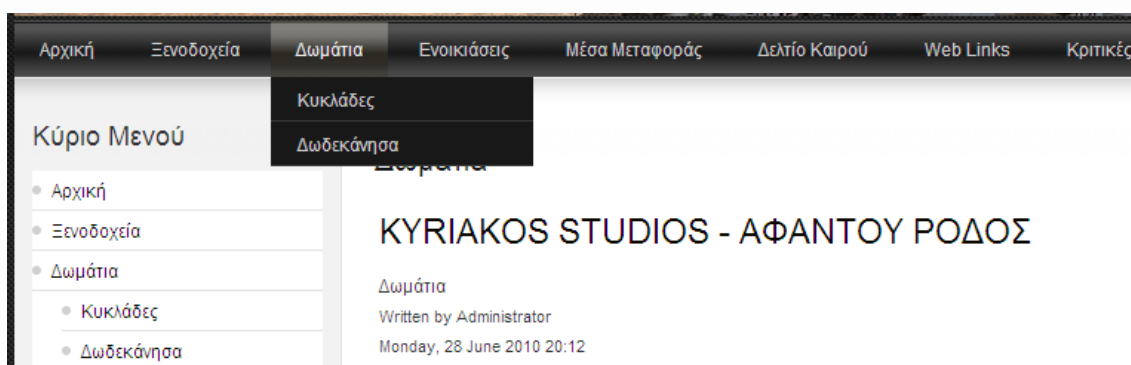
: Εικόνα: Δημιουργία συνδέσμου στο μενού για τα ξενοδοχεία.

Στην παραπάνω εικόνα παρατηρούμε πως μπορούμε να ρυθτίσουμε κάποιες παραμέτρους για το component όπως το να είναι σε μορφή συνδέσμου το όνομα της ενότητας της κατηγορίας και του άρθρου. Ακόμη να φαίνονται οι τίτλοι από τα άρθρα, ο συγγραφέας του άρθρου, η ημερομηνία και πολλές ακόμη επιλογές. Επιπλέον υπάρχει η δυνατότητα να εμφανίζεται ένα εικονίδιο στο πάνω μέρος του άρθρου που να το μετατρέπει σε pdf μορφή. Ελέγχουμε την δημόσια όψη του ιστότοπου και διαπιστώνουμε ότι έχει προστεθεί η επιλογή Ξενοδοχεία στο κύριο μενού, το οποίο εμφανίζεται στην αριστερή στήλη.

Το επόμενο πρόβλημα που θέλουμε να λύσουμε, είναι να εμφανίσουμε το κύριο μενού και πάνω και αριστερά, ώστε να είναι εύκολο για το χρήστη να εντοπίσει αυτό που θέλει. Για να το πετύχουμε, δεν είναι απαραίτητο να δημιουργήσουμε νέο μενού, αλλά πρέπει να δημιουργήσουμε ένα νέο ένθεμα (module), το οποίο να εμφανίζει το βασικό μενού επάνω από το κείμενο. Οπότε επιλέγουμε Extensions, Module Manager και New. Στον τύπο του ενθέματος επιλέγουμε Menu και συμπληρώνουμε τα υπόλοιπα στοιχεία του μενού: τίτλος του μενού (Title: Top main menu), θέση (Position:top), Menu Name: mainmenu και Menu Style: List. Στην δική μας περίπτωση όμως το template που χρησιμοποιούμε αντλεί κατευθείαν τα στοιχεία που έχουμε στο κύριο μενού (mod_mainmenu) και τα παρουσιάζει στο οριζόντιο μενού. Προσθέτοντας στοιχεία στο μενού μας (Menus, Main menu), και τα δύο ενθέματα (πάνω και αριστερά) ενημερώνονται αυτόματα και ταυτόχρονα.

Επίσης δημιουργούμε υπομενού για τις κατηγορίες. Για παράδειγμα για την κατηγορία Αθήνα δημιουργούμε ένα στοιχείο μενού (Menu Item), το οποίο έχει ως πατέρα (Parent Item) το στοιχείο μενού με τίτλο Ξενοδοχεία. Στο οριζόντιο μενού οι υποκατηγορίες φαίνονται βάζοντας το ποντίκι στο tab με το όνομα Ξενοδοχεία και εμφανίζεται ένα υπομενού (drop down menu) . Κατά τον ίδιο τρόπο δημιουργούμε συνδέσμους για τις υπόλοιπες ενότητες και κατηγορίες του ιστότοπου μας για το κύριο και το οριζόντιο μενού. Το αποτέλεσμα που θα έχει ο ιστότοπος μας όσον αφορά τα μενού φαίνεται στην παρακάτω εικόνα:

«Δημιουργία ιστότοπου με την χρήση cms»




Εικόνα: Εμφάνιση μενού και υπομενού.

9.2 Δημιουργία συνδέσμων.

Το Joomla! διαθέτει ένα σύνολο από βασικές επεκτάσεις που εγκαθίστανται μαζί με το σύστημα. Μία από αυτές είναι οι σύνδεσμοι (Web Links). Για τη δημιουργία συνδέσμων, δημιουργούμε πρώτα κατηγορίες επιλέγοντας από το μενού διαχείρισης Components, Web Links, Categories και New. Αφού δημιουργήσουμε τις κατηγορίες συνδέσμων δημιουργούμε συνδέσμους επιλέγοντας Components, Web Links, Links και New.

Για την εμφάνιση των συνδέσμων στις σελίδες μας, δημιουργούμε ένα Στοιχείο Μενού στο κύριο μενού, το οποίο είναι τύπου Web Links. Το στοιχείο αυτό το ονομάζουμε Web Links και περιέχει κάποιους συνδέσμους (links) που μας βοηθούν σε ορισμένες λειτουργίες του ιστότοπου.

Web links

#	Web Link	Display #	Hits
1	 Google	20	2

- Μηχανές Αναζήτησης (1)
- Ειδήσεις σχετικά με την Ελλάδα (1)
- Έυρεση συντεταγμένων (1)

Εικόνα: Εμφάνιση Web Links.

9.3 Μέρος του κώδικα για την εμφάνιση των μενού.

Στην ενότητα αυτή παραθέτουμε ένα μέρος του κώδικα, που βρίσκεται στο αρχείο helper.php στον κατάλογο modules/mod_mainmenu/, και είναι υπεύθυνος για τον τρόπο που μαζεύονται τα δεδομένα ενός άρθρου και τον τρόπο που παρουσιάζονται σαν σύνδεσμοι (link) στα μενού που έχουμε δημιουργήσει. Η μέθοδος getItemData επιστρέφει το αντικείμενο που χρησιμοποιήθηκε για να αρχικοποιησει το συγκεκριμένο item. Σαν παραμέτρους η μέθοδος παίρνει δύο μεταβλητές την ¶ms, η οποία έχει τις ρυθμίσεις του module από το xml αρχείο του και την &item τύπου string. Στην συνέχεια γίνεται έλεγχος για το αν υπάρχουν μενού και υπομενού τα οποία φέρνει η getItem σύμφωνα με το id του κάθε μενού.

Ο όρος CDATA χρησιμοποιείται για να περάσει δεδομένα κειμένου χωρίς να ληφθούν υπόψη από τον XML μεταγλωτιστή. Η switch δίνει κάποιες επιλογές στο πως θα είναι σε περίπτωση που έχουμε φορτώσει η διάταξη μιας φωτογραφίας για ένα μενού. Εν συνεχεία εάν υπάρχουν στοιχεία για τα μενού τυπώνονται σε μορφή συνδέσμου έτσι ώστε αν πατήσουμε σε κάποιο μενού να μεταφερθούμε στην ανάλογη σελίδα που έχουμε ζητήσει. Και για περισσότερη

ασφάλεια τοποθετείται στις παραμέτρους του μενού και ένα κλειδί ασφαλείας (access key) που του ορίζουμε μια συγκεκριμένη τιμή.

```
function _getItemData(&$params, $item)
{
    $data = null;

    // Menu Link is a special type that is a link to another item
    if ($item->type == 'menulink')
    {
        $menu = &JSite::getMenu();
        if ($newItem = $menu->getItem($item->query['Itemid'])) {
            $tmp = clone($newItem);
            $tmp->name = '<span><![CDATA['.$item->name.']></span>';
            $tmp->mid = $item->id;
            $tmp->parent = $item->parent;
        } else {
            return false;
        }
    } else {
        $tmp = clone($item);
        $tmp->name = '<span><![CDATA['.$item->name.']></span>';
    }

    $iParam = new JParameter($tmp->params);
    if ($params->get('menu_images') && $iParam->get('menu_image') &&
    $iParam->get('menu_image') != -1) {
        switch ($params->get('menu_images_align', 0)){
            case 0 :
                $imgalign='align="left"';
                break;

            case 1 :
                $imgalign='align="right"';
                break;

            default :
                $imgalign="";
                break;
        }

        $image = 'alias.'" />';
        if($tmp->ionly){
```

```

        $tmp->name = null;
    }
} else {
    $image = null;
}
switch ($tmp->type)
{
    case 'separator' :
        return '<spanclass="separator">'.$image.$tmp->name.'</span>';
        break;

    case 'url' :
        if ((strpos($tmp->link, 'index.php?') === 0) && (strpos($tmp->link, 'Itemid=') === false)) {
            $tmp->url = $tmp->link.'&Itemid='.$tmp->id;
        } else {
            $tmp->url = $tmp->link;
        }
        break;

    default :
        $router = JSite::getRouter();
        $tmp->url = $router->getMode() == JROUTER_MODE_SEF ?
'index.php?Itemid='.$tmp->id : $tmp->link.'&Itemid='.$tmp->id;
        break;
}

// Print a link if it exists
if ($tmp->url != null)
{
    // Handle SSL links
    $iSecure = $iParams->def('secure', 0);
    if ($tmp->home == 1) {
        $tmp->url = JURI::base();
    } elseif (strcasecmp(substr($tmp->url, 0, 4), 'http') && (strpos($tmp->link, 'index.php?') !== false)) {
        $tmp->url = JRoute::_($tmp->url, true, $iSecure);
    } else {
        $tmp->url = str_replace('&', '&amp;', $tmp->url);
    }
}

switch ($tmp->browserNav)
{
    default:
    case 0:

```

```

// _top
$data = '<a href="'. $tmp->url.'">'. $image.$tmp-
>name.</a>';
break;
case 1:
// _blank
$data = '<a href="'. $tmp->url.'"
target="_blank">'. $image.$tmp->name.</a>';
break;
case 2:
// window.open
$attrs =
'toolbar=no,location=no,status=no,menubar=no,scrollbars=yes,resizable=yes,'.$this->_params-
>get('window_open');

// hrm...this is a bit dickey
$link = str_replace('index.php', 'index2.php', $tmp->url);
$data = '<a href="'. $link.'"
onclick="window.open(this.href,\'targetWindow\',\''. $attrs.'\');return false;">'. $image.$tmp-
>name.</a>';
break;
}
} else {
$data = '<a>'. $image.$tmp->name.</a>';
}
return $data;
}
}

```

9.4 Διαχείριση χρηστών και πρόσβασης.

Η πληροφορία που οργανώνεται σε ενότητες, κατηγορίες και άρθρα όπως αναφέρθηκε και προηγουμένως, έχει τη σημαντική ιδιότητα της περιορισμένης πρόσβασης. Αναλυτικότερα, το Joomla ορίζει μια ιεραρχία χρηστών η οποία έχει συγκεκριμένα δικαιώματα πρόσβασης τόσο στο front-end όσο και στο διαχειριστικό back-end του συστήματος. Οι ομάδες επιπέδων πρόσβασης χωρίζονται σε αυτές για το front-end και σε αυτές για το back-end. Οι ομάδες του front-end είναι οι

παρακάτω:

- Guest (επισκέπτης)
 - Έχουν πρόσβαση μόνο στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
- Registered (εγγεγραμμένοι χρήστες)
 - Έχουν πρόσβαση μόνο στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
- Author (Συγγραφέας)

- Έχουν πρόσβαση μόνο στο front-end
- Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
- Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
- Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «special»
- Μπορούν να δημιουργήσουν νέα άρθρα και να διορθώσουν άρθρα τα οποία αυτοί δημιούργησαν
- Δεν μπορούν να δημοσιεύσουν άρθρα
- Editor (Διορθωτής)
- Publisher (Εκδότης)
 - Έχουν πρόσβαση μόνο στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «special»
 - Μπορούν να δημιουργήσουν νέα άρθρα και να διορθώσουν άρθρα τα οποία αυτοί δημιούργησαν
 - Μπορούν να διορθώσουν άρθρα και άλλων χρηστών
 - Μπορούν να δημοσιεύσουν άρθρα
 - Αντίστοιχα υπάρχουν ομάδες χρηστών οι οποίες έχουν πρόσβαση και στο διαχειριστικό back-end του ιστότοπου και παρατίθενται παρακάτω:
- Manager (Διευθυντής)
 - Έχουν πρόσβαση και στο back-end και στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «special»
 - Μπορούν να δημιουργήσουν νέα άρθρα και να διορθώσουν άρθρα τα οποία αυτοί δημιούργησαν
 - Μπορούν να διορθώσουν άρθρα και άλλων χρηστών
 - Μπορούν να δημοσιεύσουν άρθρα
- Administrator (Διαχειριστής)
 - Έχουν πρόσβαση και στο back-end και στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «special»
 - Μπορούν να δημιουργήσουν νέα άρθρα και να διορθώσουν άρθρα τα οποία αυτοί δημιούργησαν
 - Μπορούν να διορθώσουν άρθρα και άλλων χρηστών
 - Μπορούν να δημοσιεύσουν άρθρα
 - Μπορούν να δημιουργήσουν νέους χρήστες
 - Μπορούν να εγκαταστήσουν νέες προγραμματιστικές επεκτάσεις
- Super Administrator (Υπερ-διαχειριστής)
 - Έχουν πρόσβαση και στο back-end και στο front-end
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «public»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «registered»
 - Βλέπουν τα άρθρα τα οποία είναι πρόσβασης «special»
 - Μπορούν να δημιουργήσουν νέα άρθρα και να διορθώσουν άρθρα τα οποία αυτοί δημιούργησαν

- Μπορούν να διορθώσουν άρθρα και άλλων χρηστών
- Μπορούν να δημοσιεύσουν άρθρα
- Μπορούν να δημιουργήσουν νέους χρήστες
- Μπορούν να εγκαταστήσουν νέες προγραμματιστικές επεκτάσεις
- Μπορούν να αλλάξουν το template του ιστότοπου
- Μπορούν να αλλάξουν τις γενικές ρυθμίσεις συστήματος (global configuration)

Η εγγραφή των χρηστών στον δικτυακό μας τόπο μπορεί να γίνει με δύο τρόπους:

α) με την εγγραφή μέσω φόρμας από το front-end (εφόσον αυτό είναι ενεργοποιημένο). Οι νέοι χρήστες που θα δημιουργηθούν μέσω αυτής της διαδικασίας είναι εκ προεπιλογής απενεργοποιημένοι.

β) με τη δημιουργία των χρηστών μέσω του back-end από τους διαχειριστές

Για να δημιουργήσουμε έναν νέο χρήστη αρκεί να κάνουμε κλικ στην κεντρική σελίδα διαχείρισης του Joomla! στο εικονίδιο με τίτλο «User Manager». Στη συνέχεια, στη σελίδα διαχείρισης χρηστών (εικόνα 4.1) κάνουμε κλικ στο εικονίδιο «New». Ακολούθως, στην επόμενη σελίδα συμπληρώνουμε τα στοιχεία του χρήστη (εικόνα 4.2).

#	Name	Username	Logged In	Enabled	Group
1	Administrator	admin	✓	✓	Super Administrator
2	Nikos	nikos		✓	Publisher
3	nikos	nikos7		✓	Administrator

Εικόνα: Σελίδα διαχείρισης χρηστών.

Από τη σελίδα διαχείρισης χρηστών εάν κάνουμε κλικ στο εικονίδιο δίπλα στο χρήστη στη στήλη «Enabled», μπορούμε να τον ενεργοποιήσουμε ή να τον απενεργοποιήσουμε. Επιπρόσθετα, το Joomla! μας παρέχει τη δυνατότητα αναζήτησης ενός χρήστη με βάση το username του καθώς και του φιλτραρίσματος των υπάρχοντων χρηστών με βάση την ομάδα στην οποία ανήκουν αλλά και το εάν είναι εντός του συστήματος αυτή τη στιγμή. Για παράδειγμα μπορούμε να επιλέξουμε τους χρήστες που είναι αυτή τη στιγμή συνδεδεμένοι στο σύστημα και ανήκουν στην ομάδα των διορθωτών. Επιπρόσθετα, έχουμε τη δυνατότητα να διαγράψουμε έναν ή περισσότερους χρήστες, όπως επίσης και να τους διακόψουμε την τρέχουσα σύνδεση τους.

Εικόνα: Εισαγωγή νέου χρήστη.

10 Επεκτάσεις του Joomla που έχουν χρησιμοποιηθεί για τον συγκεκριμένο ιστότοπο.

Η έκδοση 1.5 άλλαξε ριζικά τη φιλοσοφία των επεκτάσεων του Joomla. Παρέχει πλέον ένα ενοποιημένο περιβάλλον όπου ο διαχειριστής μπορεί μέσω μίας και μόνο σελίδας να ανεβάσει στον δικτυακό τόπο τις επεκτάσεις που επιθυμεί ανεξαρτήτως τύπου. Οι τύποι επεκτάσεων που υποστηρίζει το Joomla είναι:

- Αρθρώματα (modules)
- Εξαρτήματα (components)
- Γλώσσες (languages)
- Πρόσθετα (plugins)
- Πρότυπα (templates)

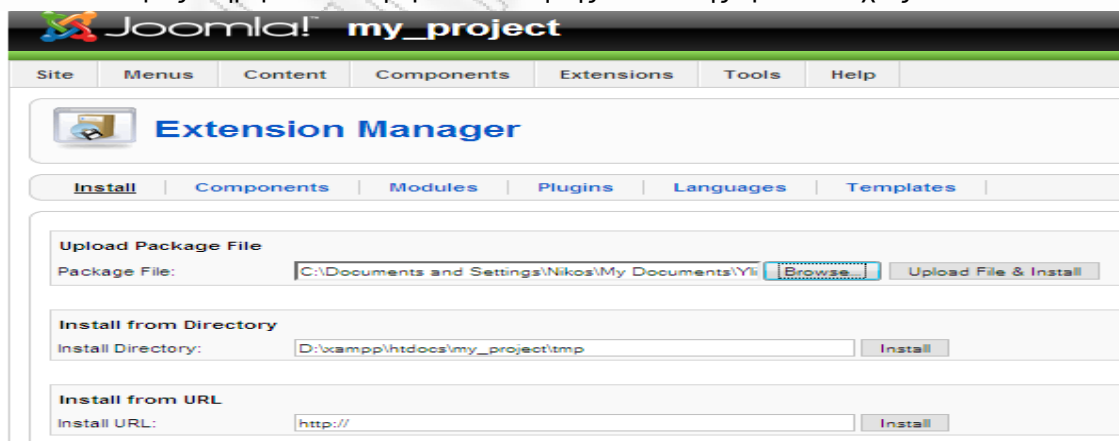
Οι επεκτάσεις αυτές είναι προσβάσιμες από το μενού «Extensions». Το Joomla διαθέτει έναν πολύ ενημερωμένο και οργανωμένο κατάλογο επεκτάσεων ο οποίος υπάρχει στη διεύθυνση <http://extensions.joomla.org>. Από εκεί μπορούμε να μεταφορτώσουμε πλήθος επεκτάσεων που καλύπτουν ένα ευρύτατο πεδίο λειτουργιών.

10.1 Εγκατάσταση και ανάλυση των επεκτάσεων

Στον δικτυακό τόπο που δημιουργήσαμε χρησιμοποιήθηκαν οι εξής επεκτάσεις:

- JComments: component
- Content Map: module
- Front Page Slideshow: module
- Content Map: plugin
- Sigplus Image Gallery Plus: plugin

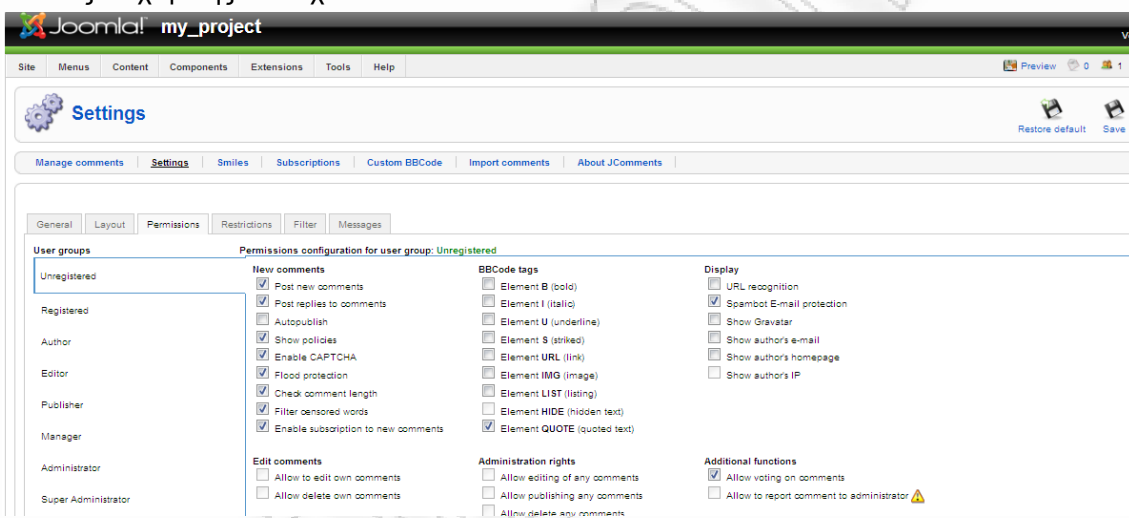
Για να εγκαταστήσουμε μια επέκταση που έχουμε κατεβάσει από το Internet, αρκεί να πάμε στο μενού «Extensions» και στη συνέχεια να κάνουμε κλικ στην επιλογή «Install/Uninstall». Στη συνέχεια θα μεταβούμε στη σελίδα εγκατάστασης, κάνουμε κλικ στο κουμπί «Browse» και επιλέγουμε από το δίσκο το αρχείο εγκατάστασης της επέκτασης. Πατάμε το κουμπί «Upload File and Install». Εάν όλα πήγαν καλά θα πρέπει να μας εμφανιστεί μήνυμα το οποίο θα μας ενημερώνει ότι η εγκατάσταση της επέκτασης έγινε επιτυχώς.



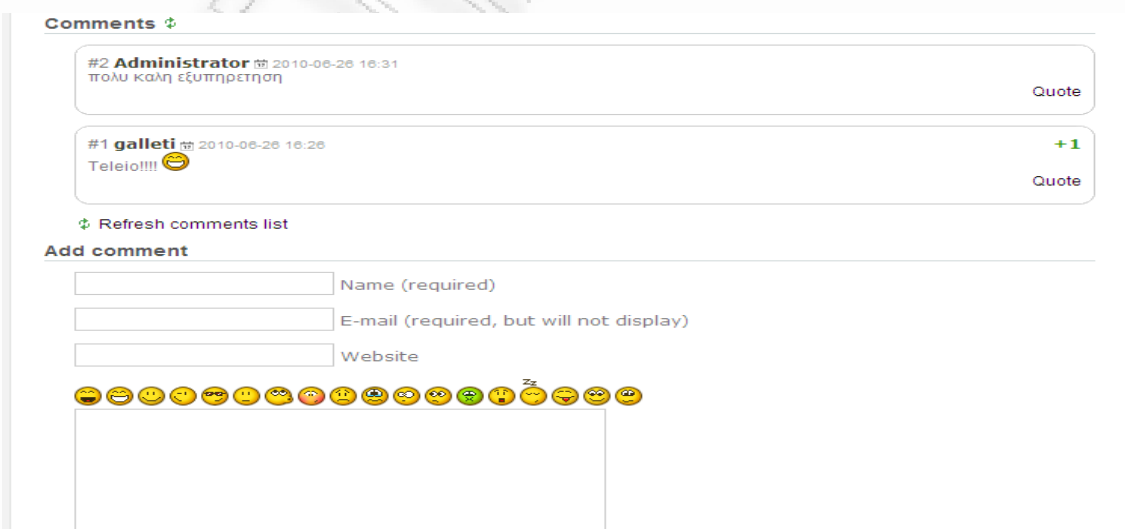
Εικόνα: Σελίδα εγκατάστασης επέκτασης.

10.1.1 JComments: component.

Το JComments είναι μια επέκταση τύπου component και μας δίνει την δυνατότητα να γράφουν οι χρήστες σχόλια για τα άρθρα που υπάρχουν στον δικτυακό μας τόπο. Πατώντας στο tab component από το administration panel του Joomla εμφανίζεται ένα συρόμενο μενού και κλικάρουμε το JComments. Με αυτόν τον τρόπο μεταφερόμαστε στη σελίδα διαχείρισης του συγκεκριμένου component όπου βρίσκονται ορισμένες ρυθμίσεις. Υπάρχει το μενού διαχείρισης των σχολίων το οποίο περιέχει όλα τα σχόλια που έχουν γίνει στο site όπου αναφέρονται επίσης το θέμα του άρθρου που έχει σχολιαστεί, από ποιόν έχει γίνει το σχόλιο η ώρα και η ημερομηνία. Ο υπερδιαχειριστής του ιστότοπου μπορεί να διαγράψει, να δημοσιεύσει ή όχι το σχόλιο και να το τροποποιήσει. Στο επόμενο πεδίο υπάρχουν επιπρόσθετες ρυθμίσεις που επιτρέπουν να διαμορφωθεί το component όπως αρμόζει σε κάθε διαχειριστή. Για παράδειγμα μπορούμε να διαλέξουμε σε ποιές κατηγορίες θέλουμε να εμφανίζεται η φόρμα για τα σχόλια, τη εμφάνιση θα έχει η συγκεκριμένη φόρμα, το εύρος των χαρακτήρων που θα έχει κάθε σχόλιο και επιπλέον ποιοι χρήστες του διαδικτυακού τόπου θα μπορούν να διατυπώνουν σχόλια η ακόμη και να απαντάνε σε σχόλιο που έγινε από άλλον χρήστη. Αφού τελειώσουμε τις ρυθμίσεις πατάμε το εικονίδιο save που βρίσκεται στην δεξιά γωνία της σελίδας. Στις παρακάτω δύο εικόνες μπορούμε να παρατηρήσουμε τόσο τον τρόπο εμφάνισης της φόρμας όσο και της σελίδας διαχείρισης των σχολίων.



Εικόνα: Σελίδα διαχείρισης JComments.



Εικόνα: Φόρμα εγγραφής και παρουσίαση των σχολίων.

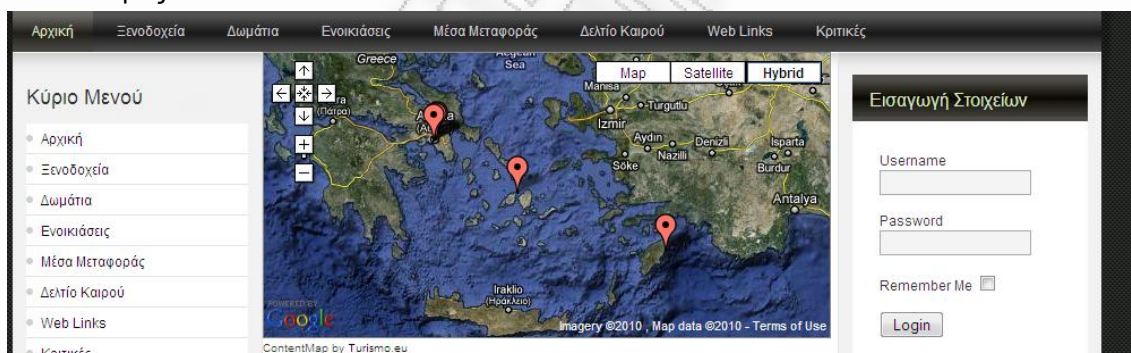
«Δημιουργία ιστότοπου με την χρήση cms»

10.1.2 Content map: module.

Το content map είναι μια επέκταση τύπου module και παρουσιάζει τα άρθρα που έχουμε στο site μας πάνω σε χάρτη της Google (google maps). Στο administration panel του Joomla πατάμε extensions και στη συνέχεια module manager βρίσκουμε το όνομα του module και κλικάρουμε πάνω σε αυτό. Με αυτόν τον τρόπο μεταφερόμαστε στη σελίδα που περιέχει στο αριστερό μέρος λεπτομέρειες για το module και σε ποία μενού θα εμφανίζεται (στη δική μας περίπτωση θα εμφανίζεται μόνο στην αρχική σελίδα) και στο δεξί μέρος τις βασικές παραμέτρους. Η θέση που το έχουμε τοποθετήσει είναι η home_gallery (την οποία έχουμε περιγράψει τον τρόπο με τον οποίο την δημιουργήσαμε στο κεφάλαιο για τα templates). Για να λειτουργήσει το content map θα πρέπει καταρχήν να έχουμε ένα API Key και το γράφουμε στο πεδίο API key που βρίσκεται στις παραμέτρους. Για να αποκτήσουμε το API key θα πρέπει να μεταβούμε στην σελίδα <http://code.google.com/intl/it-IT/apis/maps/signup.html> και να κάνουμε μια απλή εγγραφή και θα μας σταλεί το API key στο mail που θα έχουμε δώσει.

Για να εμφανιστούν τα άρθρα στο χάρτη θα πρέπει κατά την εγγραφή ενός άρθρου στο πεδίο που αναφέρεται ως key reference να γράψουμε τις συντεταγμένες που αντιστοιχούν στην διεύθυνση του ξενοδοχείου που αναφέρεται στο άρθρο χωρισμένες με ένα κόμμα. **Key Reference** Αφού κάνουμε save στο άρθρο μας δημιουργείται ένα marker στο χάρτη. Έτσι λοιπόν όσα άρθρα περιέχουν συντεταγμένες θα εμφανίζονται απευθείας. Για να βρούμε σε ποιες συντεταγμένες αντιστοιχεί μια διεύθυνση χρησιμοποιούμε ένα εργαλείο που βρίσκεται στην σελίδα <http://www.turismo.eu/it/geo.html> και κάνει αυτήν την μετατροπή.

Υπάρχουν ορισμένες ακόμα ρυθμίσεις που μπορούμε να κάνουμε όπως οι διαστάσεις που θα εμφανίζεται ο χάρτης, τι είδος χάρτη θα έχουμε (satellite, hybrid, map), το πόσα στοιχεία θα περιλαμβάνει ο χάρτης αν θα φαίνεται το όνομα του συγγραφέα του άρθρου και τέλος το χρώμα που θα έχει ο marker. Στην παρακάτω εικόνα δείχνουμε πως παρουσιάζεται το module στον ιστότοπό μας.



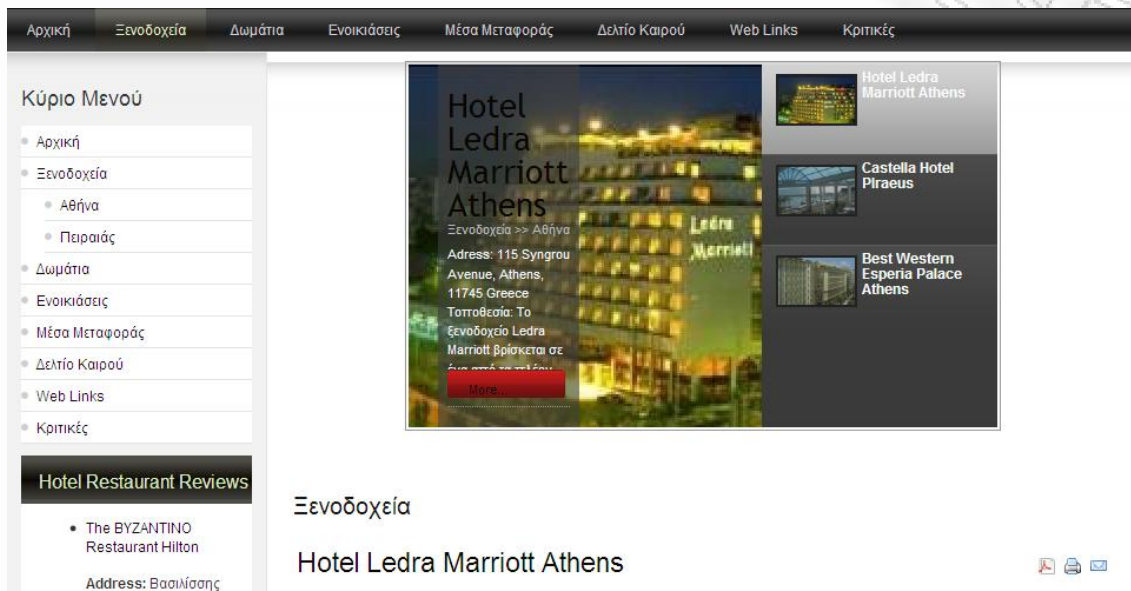
Εικόνα: Εμφάνιση του content map.

10.1.3 FrontPage Slideshow: component.

Το Frontpage slideshow είναι μια επέκταση τύπου module η οποία μας επιτρέπει να παρουσιάζουμε το περιεχόμενο της ιστοσελίδας μας με έναν αρκετά εντυπωσιακό τρόπο. Ο κώδικας του Frontpage βασίζεται κυρίως σε javascript και css και είναι ένα σύστημα προβολής διαφανειών που συνδυάζει κείμενο και κείμενο και εικόνες για να δημιουργήσει «διαφάνειες». Αυτές οι «διαφάνειες» εναλλάσσονται η μία μετά την άλλη με διάφορα εφέ, ενώ το κείμενο μπορεί να εμφανίζεται ολοκληρωμένα οπουδήποτε μέσα στο slideshow (βλέπε τα διάφορα πρότυπα που περιλαμβάνονται).

Αφού εγκαταστήσουμε το module στο administration panel του Joomla πατάμε extensions και στη συνέχεια module manager βρίσκουμε το όνομα του module και κλικάρουμε πάνω σε αυτό. Εκεί υπάρχουν οι παράμετροι για το module. Αρχικά το δημοσιεύουμε σε όποια θέση θέλουμε στην δική μας περίπτωση στη θέση home_gallery που έχουμε δημιουργήσει. Στη

συνέχεια συμπληρώνουμε το ID της κατηγορίας που θέλουμε να παρουσιάσουμε και επιπλέον κάποιες άλλες ρυθμίσεις όπως το ύψος και το πλάτος, την ταχύτητα εναλλαγής των διαφανειών, το όριο των λέξεων που θα υπάρχουν στις διαφάνειες και τέλος αν θέλουμε να υπάρχει εισαγωγικό κείμενο, read more link και να φαίνονται οι ενότητες και οι κατηγορίες. Επίσης το έχουμε ρυθμίσει να φαίνεται μόνο στην σελίδα που βρίσκονται τα ξενοδοχεία.

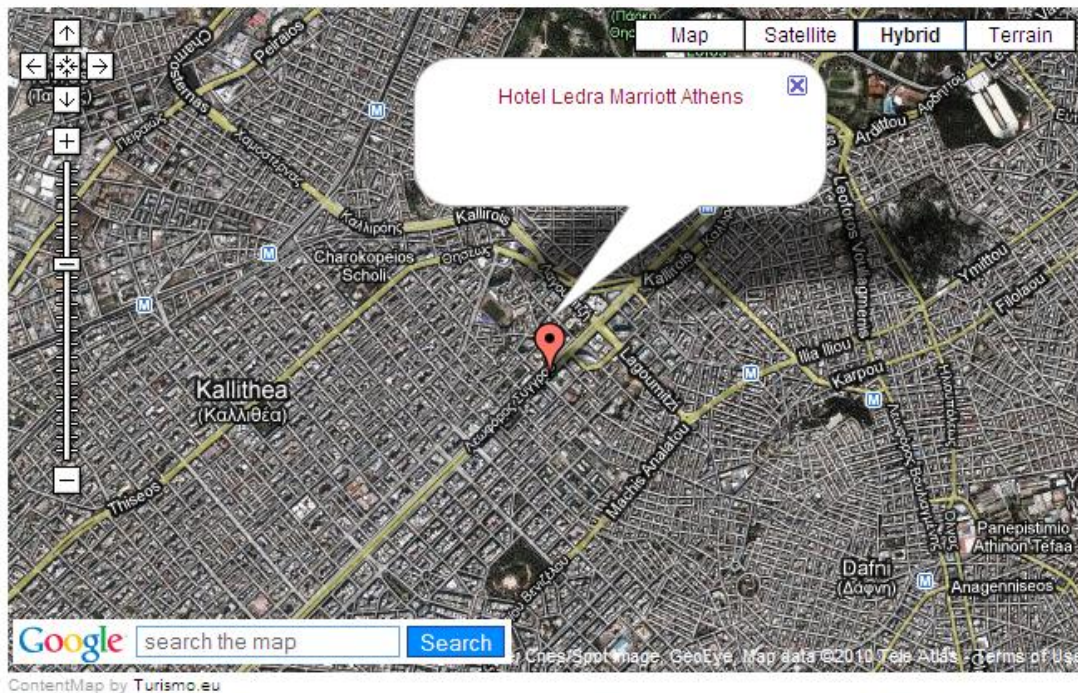


Εικόνα: Εμφάνιση FrontPage Slideshow.

10.1.4 Content Map: Plugin

Το content map είναι μια επέκταση τύπου plugin και παρουσιάζει κάτω από το άρθρο ένα χάρτη της Google (google maps) με το ακριβές σημείο της διεύθυνσης του ξενοδοχείου που αναφέρεται το άρθρο. Στο administration panel του Joomla! πατάμε extensions και στη συνέχεια module manager βρίσκουμε το όνομα του module και κλικάρουμε πάνω σε αυτό. Με αυτόν τον τρόπο μεταφερόμαστε στη σελίδα που περιέχει στο αριστερό μέρος λεπτομέρειες για το module και στο δεξί μέρος τις βασικές παραμέτρους. Όπως και στο content map module έτσι και τώρα για να λειτουργήσει το content map θα πρέπει καταρχήν να έχουμε ένα API Key και το γράφουμε στο πεδίο API key που βρίσκεται στις παραμέτρους. Για να αποκτήσουμε το API key θα πρέπει να μεταβούμε στην σελίδα <http://code.google.com/intl/it-IT/apis/maps/signup.html> και να κάνουμε μια απλή εγγραφή και θα μας σταλεί το API key στο mail που θα έχουμε δώσει.

Για να εμφανιστούν τα άρθρα στο χάρτη θα πρέπει κατά την εγγραφή ενός άρθρου στο πεδίο που αναφέρεται ως key reference να γράψουμε τις συντεταγμένες που αντιστοιχούν στην διεύθυνση του ξενοδοχείου που αναφέρεται στο άρθρο χωρισμένες με ένα κόμμα.
 Key Reference Αφού κάνουμε save στο άρθρο μας δημιουργείται ένα marker στο χάρτη με το επιθυμητό αποτέλεσμα. Μια από τις βασικές παραμέτρους που θα πρέπει να έχουμε υπόψη μας είναι η ενεργοποίηση του χάρτη που θα πρέπει να είναι σκεκαρισμένο το ναι. Άλλες βασικές ρυθμίσεις που μπορούμε να κάνουμε είναι το μήκος και το πλάτος του χάρτη, το είδος του δηλαδή αν θα είναι ορθογώνια εικόνα ή απλή, το επίπεδο zoom του χάρτη και το χρώμα που θα έχει ο τίτλος του άρθρου μέσα στο balloon. Επιπλέον δίνεται η δυνατότητα να ενεργοποιηθεί και το street view. Στην παρακάτω εικόνα φαίνεται ο τρόπος παρουσίασης του plugin.



Εικόνα: Εμφάνιση Content Map plugin.

10.1.5 Sigplus Image Gallery Plus: plugin.

Το Sigplus Image Gallery Plus είναι επέκταση τύπου plugin και αυτόματα δημιουργεί γκαλερί εικόνων τις οποίες μπορούμε να τις δούμε και σε αναδυόμενο κουτί (popup box). Για να λειτουργήσει το συγκεκριμένο plugin και να φτιάξουμε μέσα στο άρθρο και στο σημείο που θέλουμε μια γκαλερί εικόνων αρκεί να χρησιμοποιήσουμε τον κώδικα {gallery}Folder Name{/gallery}. Τις φωτογραφίες που αντιστοιχούν σε κάποιο άρθρο τις έχουμε αποθηκεύσει σε ένα φάκελο με όνομα π.χ Ledra_Marriott και αυτόν τον τοποθετούμε στο βασικό φάκελο που χρησιμοποιεί το plugin και είναι στον κατάλογο images/stories.

Ανάλογα με το πώς θέλει ο καθένας να παραμετροποιήσει το plugin μπορεί να το κάνει μέσα από τις βασικές ρυθμίσεις που βρίσκονται αν στο administration panel του Joomla πατήσουμε extensions και στη συνέχεια plugin manager βρούμε το όνομα του module και κλικάρουμε πάνω σε αυτό. Εκεί μπορούμε να ρυθμίσουμε τον αριθμό των εικόνων που θα έχει η γκαλερί το ύψος και το πλάτος τους. Ακόμη το στυλ που θα έχει το popup box και το πώς θα φαίνονται τα κουμπιά πλοήγησης (Next, Prev). Επίσης στις παραμέτρους για προχωρημένους ρυθμίζουμε την ποιότητα των thumbnails και την σειρά με την οποία θα εμφανίζονται οι εικόνες είτε σύμφωνα με το όνομα είτε με την ημερομηνία που τροποποιήθηκαν τελευταία φορά. Στη παρακάτω εικόνα φαίνεται ο τρόπος παρουσίας του plugin.

ΤΥΠΟΣ Δωματίου	ΛΑΜΗΛΗ ΠΕΡΙΟΔΟΣ 26 Απρ. - 10 Ιουν. & 25 Σεπτ. - 30 Οκτ.	ΜΕΣΗ ΠΕΡΙΟΔΟΣ 11 Ιουν. - 12 Ιουλ. & 26 Αυγ. - 24 Σεπ.	ΥΨΗΛΗ ΠΕΡΙΟΔΟΣ 13 Ιουλ. - 25 Αυγ.
Στούντο (2 άτομα)	35.00 €	45.00 €	75.00 €
Στούντο (3 άτομα)	45.00 €	55.00 €	90.00 €

Οι ανωτέρω τιμές είναι ανά δωμάτιο / διανυκτέρευση

Address: Λεωφ. Συγγρού 115 (Ledra Marriott Hotel)
Cuisine: ΜΕΣΟΓΕΙΑΚΗ ΚΟΥζίΝΑ

Πολιτική

Map Satellite Hybrid Terrain

Εικόνα: Εμφάνιση SigPlus Image Gallery Plus plugin.

11 RSS Feed (Really Simple Syndication)

11.1 Τι ακριβώς είναι το RSS Feed

Το RSS Feed είναι ένας εναλλακτικός τρόπος να ενημερώνονται οι χρήστες του Διαδικτύου για τα γεγονότα. Το Διαδίκτυο αποτελείται πλέον από δισεκατομμύρια σελίδες οι οποίες περιέχουν τέτοιο πλούτο πληροφοριών που είναι σχεδόν αδύνατο για τον οποιοδήποτε να μπορεί να παρακολουθεί διαρκώς ότι νεότερο συμβαίνει στον κόσμο ή στο αντικείμενο που τον ενδιαφέρει. Εδώ έρχεται να δώσει τη λύση το RSS Feed. Πλέον όλες οι πληροφορίες που ενδιαφέρουν τους χρήστες έρχονται στον υπολογιστή τους χωρίς εκείνοι να χρειάζεται να επισκέπτονται κάθε φορά τους σχετικούς δικτυακούς τόπους.

Το RSS τους επιτρέπει να βλέπουν πότε ανανεώθηκε το περιεχόμενο των δικτυακών τόπων που τους ενδιαφέρει. Μπορούν να λαμβάνουν κατευθείαν στον υπολογιστή τους τους τίτλους των τελευταίων ειδήσεων και των άρθρων που επιθυμούν (ή ακόμα και εικόνων ή βίντεο) αμέσως μόλις αυτά γίνουν διαθέσιμα χωρίς να είναι απαραίτητο να επισκέπτονται καθημερινά τους αντίστοιχους δικτυακούς τόπους.

Τα αρχικά RSS σημαίνουν Really Simple Syndication ή Rich Site Summary ή RDF Site Summary. Είναι ένα έγγραφο τύπου XML. Το RSS feed χρησιμοποιείται για να διανέμει περιεχόμενα Διαδικτύου. Είναι μία τροφοδοσία (feed) ειδήσεων, στην οποία κάποιος πρέπει να προσυπογράψει για να την παραλάβει. Δηλαδή κάποιος χρήστης θα επιλέξει ο ίδιος τι είδους νέα τον ενδιαφέρουν για να ενημερωθεί και από πού, ώστε να ρυθμίσει κατάλληλα τον αναγνώστη τροφοδοσιών για να λαμβάνει αυτά τα νέα. Αυτό λοιπόν επιτυγχάνεται πολύ απλά κάνοντας ο χρήστης ένα απλό κλικ σε ένα μικρό εικονίδιο RSS ή XML, που βρίσκεται στην ιστοσελίδα από όπου ο χρήστης επιθυμεί να λάβει αυτές τις ειδήσεις και να προσθέσει ότι RSS θέλει στον αθροιστή της λίστας. Όταν λοιπόν βρίσκονται ανανεωμένες ειδήσεις, μεταδίδονται σε όλους τους συνδρομητές της ιστοσελίδας μέσω των τροφοδοσιών.

Η απλότητα του RSS feed είναι αυτή που το έκανε τόσο διαδεδομένο μεταξύ των κατασκευαστών ιστοσελίδων προκειμένου να διανέμουν νέα, weblogs και περιεχόμενα

Διαδικτύου. Αυτό συνέβη διότι τα RSS feeds αποστέλλονται άμεσα, χωρίς να υπάρχει καμία αναμονή. Οι χρήστες επίσης δεν χρειάζεται να ανησυχούν για τα φίλτρα spam, καθώς τα RSS feeds δεν θεωρούνται spam.

11.2 Πως λειτουργεί το RSS Feed στο Joomla.

Το Joomla χειρίζεται τα RSS Feeds μέσω ενός module που ονομάζεται mod_feed. Μια «τροφοδοσία» (feed) περιέχει μία λίστα από στοιχεία ή εγγραφές, καθένα από αυτά προσδιορίζονται από έναν σύνδεσμο. Κάθε στοιχείο μπορεί να έχει ένα σύνολο άλλων μεταδεδομένων με το οποίο να σχετίζεται. Τα πιο σημαντικά μεταδεδομένα για κάθε εγγραφή περιλαμβάνουν ένα τίτλο για τον σύνδεσμο και μια περιγραφή του. Βασικό ρόλο στο RSS έχουν τα αρχεία xml. Ένα αρχείο xml έχει την παρακάτω μορφή.

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
  <name>Feed Display With Order By Date</name>
  <author>Joomla! Project- Configured by ME</author>
  <creationDate>October 2010</creationDate>
  <copyright>Copyright (C) 2005 - 2010 Open Source Matters. All rights
  reserved.</copyright>
  <license>http://www.gnu.org/licenses/gpl-2.0.html GNU/GPL</license>
  <authorEmail>admin@joomla.org</authorEmail>
  <authorUrl>www.joomla.org</authorUrl>
  <version>1.5.0</version>
  <description>This module allows the displaying of a syndicated feed and include a
  sorting date option</description>
  <files>
  <filename module="mod_feed_by_order">mod_feed_by_order.php</filename>
  </files>
  <params>
    <param name="moduleclass_sfx" type="text" default="" label="Module Class
    Suffix" description="PARAMMODULECLASSSSUFFIX" />
    <param name="@spacer" type="spacer" default="" label="" description="" />
    <param name="rssurl" type="text" size="50" default="" label="Feed URL"
    description="Enter the URL of the RSS/RDF feed" />
    " "
    " "
  </params>
</install>
```

Η πρώτη γραμμή ορίζει την έκδοση της XML που θα χρησιμοποιηθεί και το σύνολο κωδικοποίησης χαρακτήρων. Η δεύτερη γραμμή δηλώνει την εγκατάσταση του module και στη συνέχεια παραθέτονται κάποια στοιχεία σχετικά με το ποιος έχει δημιουργήσει το module, τίτλος, περιγραφή του module κτλ. Στη συνέχεια ακολουθεί ένα tag με όνομα file με το οποίο ενσωματώνεται το php αρχείο που περιέχει των κώδικα που φανερώνει τα RSS στον χρήστη. Ακολουθούν params tags παράμετροι που κρατούν τα link από το οποίο τραβάμε RSS τίτλοι και περιγραφές των άρθρων.

Στο αρχείο `helper.php` που βρίσκεται στο φάκελο του καταλόγου `\modules\mod_feed_by_order` βρίσκεται η κλάση `modFeedByOrderHelper` η οποία βασίζεται στην συνάρτηση `getFeed($params)` που περνάει σαν αντικείμενα σε ένα πίνακα τα δεδομένα από τα url. Παραθέτουμε τον κώδικα της κλάσης.

```
class modFeedByOrderHelper
{
    function getFeed($params)
    {
        // module params
        $rssurl = $params->get('rssurl', "");

        // get RSS parsed object
        $options = array();
        $options['rssUrl'] = $rssurl;
        if ($params->get('cache')) {
            $options['cache_time'] = $params->get('cache_time', 15) ;
            $options['cache_time'] *= 60;
        } else {
            $options['cache_time'] = null;
        }

        $rssDoc =& JFactory::getXMLparser('RSS', $options);
        // enable ordering by date and order based on the date order parameter
        $rssdateorder = $params->get('rssdateorder', "");
        if ($rssdateorder != 2) {
            $rssDoc->enable_order_by_date();
            $rssDoc->order_by_date = $rssdateorder;
        }

        $feed = new stdClass();

        if ($rssDoc != false)
        {
            // channel header and link
            $feed->title = $rssDoc->get_title();
            $feed->link = $rssDoc->get_link();
            $feed->description = $rssDoc->get_description();

            // channel image if exists
            $feed->image->url = $rssDoc->get_image_url();
            $feed->image->title = $rssDoc->get_image_title();

            // items
            $items = $rssDoc->get_items();

            // feed elements
```



```

        $feed->items = array_slice($items, 0, $params->get('rssitems', 5));
    } else {
        $feed = false;
    }

    return $feed;
}
}

```

Στην γραμμή αυτή του κώδικα `$rssDoc =& JFactory::getXMLparser('RSS', $options);` η βοηθητική συνάρτηση του Joomla, που έχει σαν ορίσματα μια μεταβλητή `&type` που καθορίζει ότι αναφερόμαστε σε RSS και η `&options` που είναι πίνακας στην οποία περνάμε τα url από τις τροφοδοσίες (Feeds), επιστρέφει μια αναφορά και καθορίζει το αντικείμενο της κλάσης. Αυτό περιέχει όλες τις παραμέτρους που έχουμε ορίσει στο αρχείο xml.

11.2.1 Παραμετροποίηση του αρχικού module.

Αλλάζουμε το αρχικό module ώστε να δώσουμε σαν επιλογή το να εμφανίζονται οι τροφοδοσίες από κάποιο άλλο site ταξινομημένες ανά ημερομηνία από την πιο παλιά στην πιο νέα και το αντίθετο. Για να γίνει αυτό θα πρέπει να προσθέσουμε ορισμένα κομμάτια κώδικα τόσο στα αρχεία που αναφέρονται στο front end όσο και για το back end. Έτσι λοιπόν στο αρχείο `helper.php` προσθέτουμε τις παρακάτω γραμμές κώδικα κάτω ακριβώς από την `$rssDoc =& JFactory::getXMLparser('RSS', $options);`

```

// enable ordering by date and order based on the date order parameter
    $rssdateorder = $params->get('rssdateorder', "");
    if ($rssdateorder != 2) {
        $rssDoc->enable_order_by_date();
        $rssDoc->order_by_date = $rssdateorder;
    }

```

Επίσης προσθέτουμε στο αρχείο `mod_feed_by_order.xml` που βρίσκεται στον κατάλογο `administrator/modules/mod_feed_by_order/` τις παρακάτω γραμμές κώδικα κάτω από την `<param name="word_count"...../>`

```

<param name="rssdateorder" type="radio" default="2" label="Date Order"
description="PARAMDATEORDER">
    <option value="1">Descending</option>
    <option value="0">Ascending</option>
    <option value="2">No Date Ordering</option>
</param>

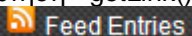
```

Έτσι δημιουργείται η επιλογή στο panel του διαχειριστή για την αύξουσα η φθίνουσα σειρά ανά ημερομηνία των RSS. Τέλος θα πρέπει να προσθέσουμε στο αρχείο `en-GB.mod_feed_order.ini` που βρίσκεται στον κατάλογο `/language/en-GB/` την γραμμή `PARAMDATEORDER=Changes the date ordering of the feeds.` Στον ιστότοπο που φτιάξαμε έχουμε ρυθμίσει το module των RSS να παίρνει τα δεδομένα από τα πολιτιστικά νέα που μας προσφέρει το site <http://www.tanea.gr>. Στην παρακάτω εικόνα φαίνονται οι ρυθμίσεις που έχουμε κάνει και η καινούρια επιλογή `date order` που δημιουργήσαμε.

Feed URL	<input type="text" value="http://www.tanea.gr/default.asp?pid=69&la=1"/>
RTL Feed	<input type="radio"/> Yes <input checked="" type="radio"/> No
Feed Title	<input checked="" type="radio"/> Yes <input type="radio"/> No
Feed Description	<input checked="" type="radio"/> Yes <input type="radio"/> No
Feed Image	<input type="radio"/> Yes <input checked="" type="radio"/> No
Items	<input type="text" value="3"/>
Item Description	<input checked="" type="radio"/> Yes <input type="radio"/> No
Word Count	<input type="text" value="0"/>
Date Order	<input type="radio"/> Descending <input checked="" type="radio"/> Ascending <input type="radio"/> No Date Ordering

Εικόνα: Ρυθμίσεις RSS.

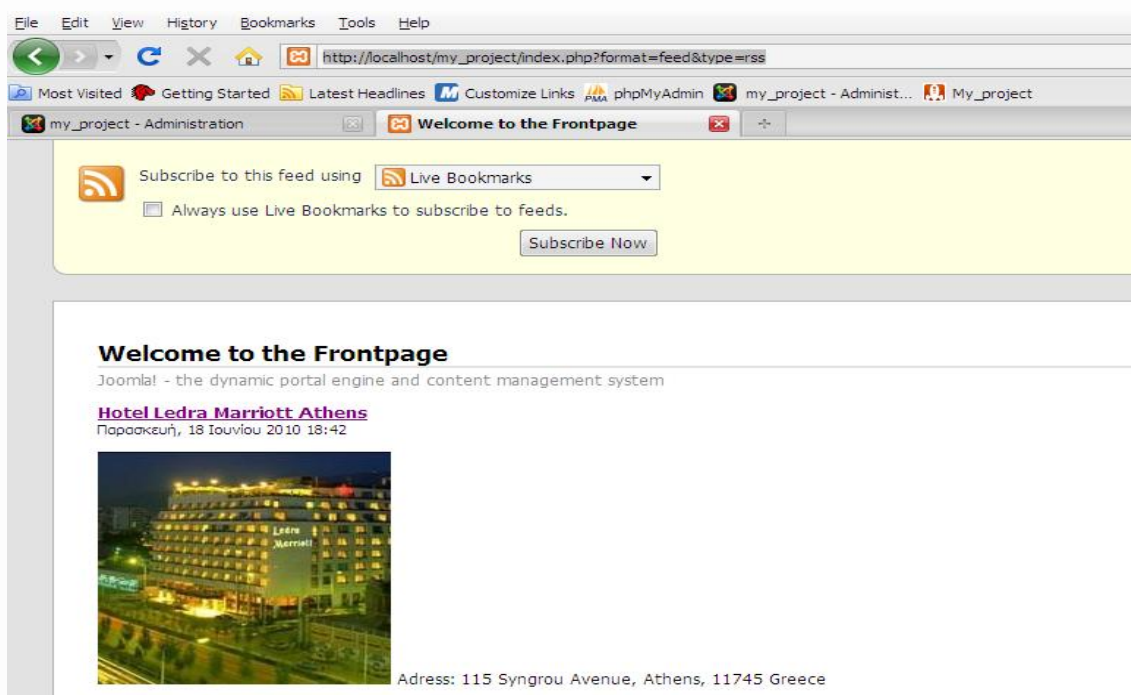
11.2.2 Feed Entries

Όπως παίρνουμε πληροφορίες μέσω rss από άλλους ιστοχώρους έτσι και εμείς μπορούμε να προσφέρουμε κάποια από τα άρθρα μας που πιστεύουμε ότι είναι άξια αναφοράς σε μορφή RSS. Σε ιστότοπους που βασίζονται στο Joomla για αυτό φροντίζει το syndicate module το οποίο βασίζεται στην συνάρτηση `getLink()` που βρίσκεται στο αρχείο `helper.php` και οπτικοποιείται με το σύμβολο 

```
function getLink(&$params)
{
    $document =& JFactory::getDocument();

    foreach($document->_links as $link)
    {
        if(strpos($link, 'application/'. $params->get('format').'+xml'))
        {
            preg_match("#href=\"(.*)\"#s", $link, $matches);
            return $matches[1];
        }
    }
}
```

Η `getLink()` βάζει στη μεταβλητή `$params` τα δεδομένα από το αρχείο `mod_syndicate.xml` και στη συνέχεια με την `JFactory::getDocument()`; μετατρέπει σε link για Rss την σελίδα στην οποία έχουμε τοποθετήσει το module. Η `preg_match()` έχει 3 ορίσματα το πρώτο είναι το πρότυπο (pattern) που ψάχνει η συνάρτηση το δεύτερο είναι το input string και το τρίτο αν το `matches` είναι πίνακας που γεμίζει με τα αποτελέσματα του search, `$matches[0]` περιέχει το κείμενο το οποίο ταιριάζει με το πλήρες pattern και `$matches[1]` θα περιέχει το κείμενο που ταχτοποίησε πρώτο ανάμεσα στις παρενθέσεις. Πατώντας λοιπόν το εικονίδιο Feed Entries μετατρέπεται η σελίδα μας σε RSS και έχει την μορφή της παρακάτω εικόνας με link http://localhost/my_project/index.php?format=feed&type=rss.



Εικόνα: Εμφάνιση σελίδα σε μορφή RSS

12 Δημιουργία κώδικα για την υλοποίηση ενός εξαρτήματος (Component).

Τα εξαρτήματα (components) στο Joomla είναι μαζί με τα αρθρώματα (modules), δύο από τους σημαντικότερους μηχανισμούς επέκτασης της λειτουργικότητάς του. Τα εξαρτήματα είναι αυτόνομα προγράμματα τα οποία αλληλεπιδρούν με τον λειτουργικό πυρήνα του Joomla και ενσωματώνονται εν τέλει πλήρως σε αυτό. Το συγκεκριμένο component που θα δημιουργήσουμε θα αναφέρετε σε αξιολογήσεις εστιατορίων ξενοδοχείων. Οι αξιολογήσεις θα μπορούν να γίνονται από τους εκάστοτε διαχειριστές του ιστότοπου ή από χρήστες οι οποίοι θα έχουν δικαίωμα έκδοσης άρθρων. Επίσης θα υπάρχει και ένας χώρος στο τέλος κάθε αξιολόγησης όπου θα μπορεί ο καθένας, είτε είναι απλός επισκέπτης είτε εγγεγραμμένος χρήστης, να γράψει σχόλια για τα εστιατόρια που είναι αναρτημένα στο site.

12.1 Η δομή ενός component.

Το Joomla χρησιμοποιεί ένα συγκεκριμένο σχέδιο ονομάτων, το οποίο χρησιμοποιείται από όλα τα components. Κάθε εξάρτημα (component) στο σύστημα έχει ένα μοναδικό όνομα χωρίς κενά. Ο κώδικας χωρίζεται σε δύο φακέλους και ονομάζονται αντίστοιχα `com_` και ακολουθεί το όνομα του component που στην δική μας περίπτωση θα ονομάζεται `reviews` (κριτική). Επομένως δημιουργούμε δύο φακέλους που θα ονομάζονται `com_reviews`.

Ο ένας θα τοποθετηθεί στο φάκελο `components` του Joomla και θα αφορά το `front end` και ο άλλος στο φάκελο `administrator/components` και θα αφορά το `back end`. Όταν φορτώσουμε το component τόσο από το `front end` όσο και από το `back end` το Joomla θα ψάξει για 2 αρχεία που θα έχουν το όνομα του component και την κατάληξη `.php`. Άρα δημιουργούμε δύο αρχεία

com_reviews.php και *admin.reviews.php* τα οποία τοποθετούνται αντίστοιχα στον φακέλο *components/com_reviews* και *administrator/components/com_reviews*.

Στο πρώτο αρχείο γράφουμε τον παρακάτω κώδικα

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
echo '<div class="componentheading">Hotel Restaurant Reviews</div>';
?>
```

και στο δεύτερο

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
echo ' Hotel Restaurant Reviews';
?>
```

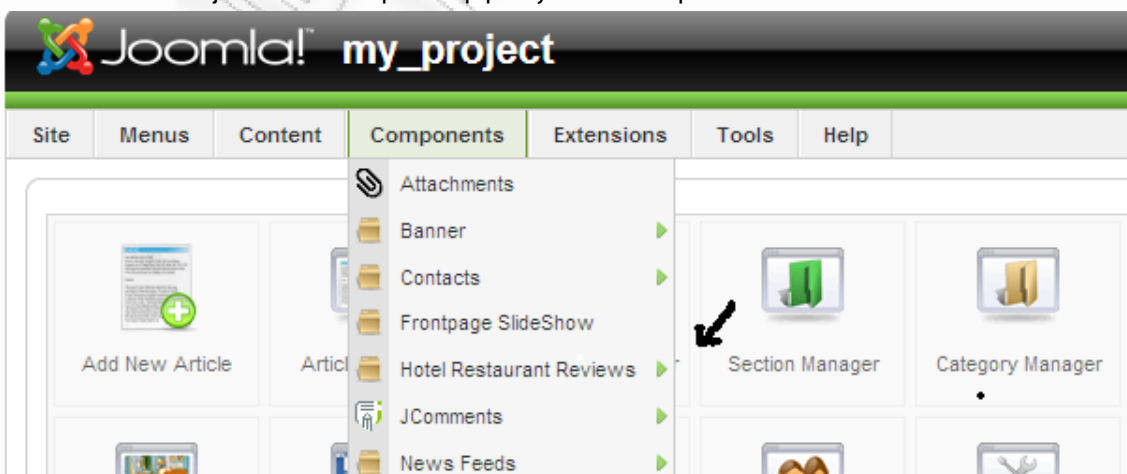
Η συνάρτηση `defined('_JEXEC')` με τη σταθερά `_JEXEC` χρησιμοποιείται για να ελέγξει αν ο κώδικας καλείται μέσα από το σύστημα του Joomla εάν δεν ισχύει αυτό σημαίνει ότι κάποιος προσπαθεί να μπει στο αρχείο μας. Οπότε το Joomla αυτόματα διαμορφώνει το περιβάλλον με ορισμένα μέτρα προστασίας ασφαλείας τα οποία θα αποτρέψουν την επίθεση.

12.2 Εγγραφή του component στη βάση δεδομένων.

Για να μπορούν οι χρήστες να μεταπηδούν μεταξύ front και back end είναι απαραίτητο να δηλώσουμε το component στη βάση δεδομένων. Για να γίνει η εγγραφή στη βάση μας, η οποία έχει το πρόθεμα `jos_`, μπορούμε να χρησιμοποιήσουμε το `interface` του `php my admin` ή αλλιώς γράφοντας ένα `sql query` όπως το παρακάτω:

```
INSERT INTO jos_components (name, link, admin_menu_link,
admin_menu_alt, option, admin_menu_img, params)
VALUES (Hotel Restaurant Reviews, option=com_reviews,
option=com_reviews, Manage Reviews, com_reviews,
js/ThemeOffice/component.png, );
```

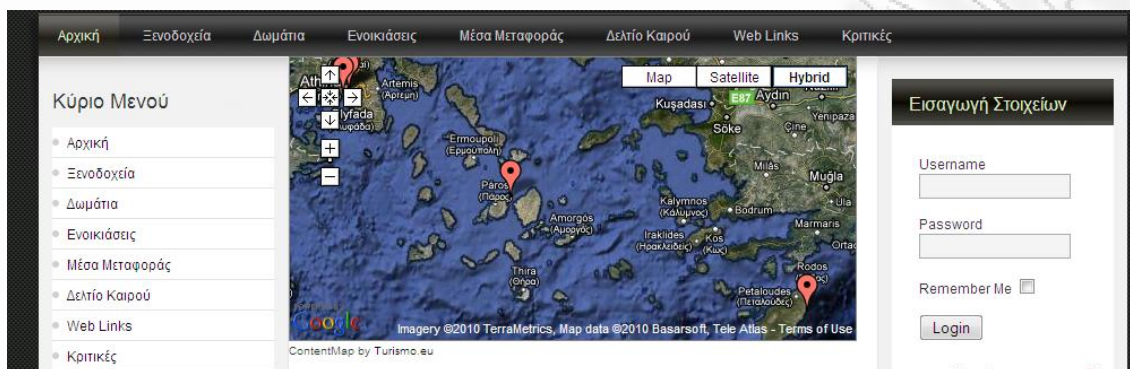
Προσθέτοντας αυτό το αρχείο δίνουμε στο σύστημα κάποιες βασικές πληροφορίες για το component. Αυτές είναι το όνομα του component, οι σύνδεσμοι (links) που μας επιτρέπουν να εναλλασσόμαστε μεταξύ front και back end, η φωτογραφία για το μενού του component και τέλος ορισμένες παραμέτρους. Αν όλα έχουν γίνει επιτυχώς κάνοντας ανανέωση στη σελίδα του `administrator` του Joomla θα δούμε να εμφανίζεται το component.



Εικόνα: Οθόνη Back-End με την παρουσία του νέου μας component.

«Δημιουργία ιστότοπου με την χρήση cms»

Αυτό που μένει τώρα είναι να δημιουργήσουμε ένα link στο ιστότοπο μας όπου εκεί θα εμφανίζονται τα περιεχόμενα του component. Και αυτό γίνεται ως εξής, από το administration panel κάνουμε κλικ στο menus/main menu και στη συνέχεια πατάμε το εικονίδιο new και στη σελίδα που εμφανίζεται επιλέγουμε Hotel Restaurant Reviews και στο πεδίο name γράφουμε Κριτικές στη συνέχεια πατάμε Save και βλέπουμε το αποτέλεσμα στην παρακάτω εικόνα.



Εικόνα: Το link με όνομα κριτικές υπάρχει τόσο στο κύριο μενού όσο και στο οριζόντιο.

12.3 Δημιουργία μπάρας εργαλείων.

Όλα τα components που συνδέονται με τον πυρήνα του Joomla! διαθέτουν κούμπια με τα οποία μπορείς να σώσεις, να διαγράψεις ή να δημοσιεύσεις στοιχεία. Έτσι και εμείς θα δημιουργήσουμε παρόμοια εργαλεία γράφοντας τον παρακάτω κώδικα σε ένα νέο αρχείο *toolbar.reviews.html.php* μέσα στον φάκελο *administrator/components/com_reviews*.

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
class TOOLBAR_reviews
{
    function _NEW()
    {
        JToolBarHelper::save();
        JToolBarHelper::apply();
        JToolBarHelper::cancel();
    }
    function _DEFAULT()
    {
        JToolBarHelper::title( JText::_ ( 'Hotel Restaurant Reviews' ), 'generic.png' );
        JToolBarHelper::publishList();
        JToolBarHelper::unpublishList();
        JToolBarHelper::editList();
        JToolBarHelper::deleteList();
        JToolBarHelper::addNew();
    }
}
}
```

«Δημιουργία ιστότοπου με την χρήση cms»

?>

Στην κλάση `TOOLBAR_reviews` δημιουργούμε δύο συναρτήσεις όπου η καθεμία δημιουργεί διαφορετικά εργαλεία με την βοήθεια της κλάσης `JToolBarHelper`. Η συγκεκριμένη κλάση περιέχει συναρτήσεις, ενσωματωμένες στο Joomla (`administrator/includes/toolbar.php`) που παράγουν όλων τον HTML κώδικα για την μπάρα εργαλείων. Αφού καθορίστηκαν οι συναρτήσεις θα πρέπει να προσθέσουμε επιπλέον τον παρακάτω κώδικα που θα αποφασίζει ποια συναρτηση θα εμφανίζεται. Έτσι δημιουργούμε ένα νέο αρχείο με όνομα `toolbar.reviews.php` στο φάκελο `administrator/components/com_reviews`.

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
require_once( JApplicationHelper::getPath( 'toolbar_html' ) );
switch( $task )
{
    case 'edit':
    case 'add':
        TOOLBAR_reviews::_NEW();
        break;
    default:
        TOOLBAR_reviews::_DEFAULT();
        break;
}
?>
```

Η γραμμή `require_once(JApplicationHelper::getPath('toolbar_html'));` χρησιμοποιεί την συνάρτηση `getPath()` της κλάσης `JApplicationHelper` (βοηθητική κλάση του Joomla η οποία περιέχει συναρτήσεις που περνούν στοιχεία του client όπως paths και XML αρχεία σε arrays) ώστε να φορτώθει το αρχείο `toolbar.reviews.html.php` χωρίς να αναφέρεται το όνομα του component. Η μεταβλητή `$task` είναι δηλωμένη σαν global από το Joomla και είναι υπεύθυνη για να κατευθύνει τη λογική ροή μέσα σε ένα component. Ο παραπάνω κώδικας δημιουργεί την επόμενη εικόνα



Εικόνα: Μπάρα εργαλείων

13 Ανάπτυξη κώδικα για το παρασκήνιο (back end) του component.

13.1 Δημιουργία πίνακα βάσης δεδομένων

Για να καταχωρούμε τις κριτικές για τα εστιατόρια των ξενοδοχείων είναι απαραίτητο να δημιουργήσουμε έναν πίνακα στη βάση μας όπου θα αποθηκεύονται. Σε αυτόν τον πίνακα η κάθε γραμμή θα αντιπροσωπεύει μια κριτική στο σύστημα. Όπως και προηγουμένως γράφουμε το sql query ή χρησιμοποιούμε το interface του php my admin.

```
CREATE TABLE jos_reviews
```

«Δημιουργία ιστότοπου με την χρήση cms»

```
(
    id int(11) NOT NULL auto_increment,
    restaurant_name varchar(255) NOT NULL,
    address varchar(255) NOT NULL,
    reservations varchar(40) NOT NULL,
    quick_note text NOT NULL,
    review text NOT NULL,
    notes text NOT NULL,
    smoking tinyint(1) NOT NULL default 0,
    credit_cards varchar(255) NOT NULL,
    cuisine varchar(40) NOT NULL,
    avg_dinner_price tinyint(4) NOT NULL default 0,
    review_date datetime NOT NULL,
    published tinyint(1) NOT NULL default 0,
    PRIMARY KEY (id)
);
```

Εάν χρησιμοποιούμε phpMy Admin βάζουμε τις συγκεκριμένες τιμές της παρακάτω εικόνας και προσέχουμε το id, που είναι και το primary key να είναι σε θέση auto_increment.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	int(11)			No	None	auto_increment	[Icons]
<input type="checkbox"/> restaurant_name	varchar(255)	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> address	varchar(255)	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> reservations	varchar(40)	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> quick_note	text	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> review	text	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> notes	text	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> smoking	tinyint(1)			No	0		[Icons]
<input type="checkbox"/> credit_cards	varchar(255)	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> cuisine	varchar(40)	utf8_general_ci		No	None		[Icons]
<input type="checkbox"/> avg_dinner_price	tinyint(4)			No	0		[Icons]
<input type="checkbox"/> review_date	datetime			No	None		[Icons]
<input type="checkbox"/> published	tinyint(1)			No	0		[Icons]

Εικόνα: Δημιουργία πίνακα μέσω phpMy Admin.

13.2 Δημιουργία κλάσης πίνακα.

Επόμενο βήμα είναι να γράψουμε τον κώδικα για συναρτήσεις με τις οποίες θα διαχειριζόμαστε τις κριτικές σε αυτό. Βέβαια σε αυτό μας βοηθάει το Joomla! το οποίο παρέχει μια κλάση την JTable που διαθέτει όλες τις απαραίτητες συναρτήσεις για να προσθέσουμε, να διαγράψουμε, να ανανεώσουμε και να διαβάσουμε τα δεδομένα που υπάρχουν στον πίνακα της βάσης μας.

Θα πρέπει λοιπόν να φτιάξουμε μια επέκταση της κλάσης JTable που θα αναφέρετε στον πίνακα jos_reviews. Μέσα στο φάκελο /administrator/components/com_reviews δημιουργούμε ένα νέο φάκελο tables ο οποίος περιέχει το αρχείο reviews.php και γράφουμε τον παρακάτω κώδικα.

```
<?php
```

«Δημιουργία ιστότοπου με την χρήση cms»

```

defined( '_JEXEC' ) or die( 'Restricted access' );
class TableReview extends JTable
{
    var $id = null;
    var $restaurant_name = null;
    var $address = null;
    var $reservations = null;
    var $quick_note = null;
    var $review = null;
    var $notes = null;
    var $smoking = null;
    var $credit_cards = null;
    var $cuisine = null;
    var $avg_dinner_price = null;
    var $review_date = null;
    var $published = null;
    function __construct( &$db )
    {
        parent::__construct( '#__reviews', 'id', $db );
    }
}
?>

```

Επεκτείνοντας την κλάση προσθέτουμε όλες τις στήλες του πίνακα σαν member variables (σχετίζονται με μια συγκεκριμένη κλάση αλλά είναι προσπελάσιμες από όλες τις μεθόδους της κλάσης) και τις μηδενίζουμε (null). Επίσης συμπληρώνουμε την κλάση constructor με την μέθοδο `__construct` η οποία παίρνει ένα αντικείμενο της βάσης σαν παράμετρο και καλεί τον εκάστοτε constructor χρησιμοποιώντας το όνομα του πίνακα της βάσης το βασικό κλειδί και το αντικείμενο της βάσης. Η κλάση `TableReview` κληρονομεί μεταξύ άλλων τις συναρτήσεις `bind()`, `store()`, `load()`, `delete()` που μας βοηθούν να χειριστούμε τα δεδομένα στη βάση χωρίς να γράφουμε επιπλέον κώδικα.

13.3 Δημιουργία της φόρμας του back end για την εισαγωγή κριτικών.

Αφού φτιάξαμε τον πίνακα στη βάση για να αποθηκεύει τα δεδομένα επόμενο βήμα είναι η δημιουργία μιας διεπαφής (interface) φιλικής προς το χρήστη για την καταχώρηση των κριτικών (reviews). Όπως κάναμε και στη γραμμή εργαλείων χωρίζουμε τον κώδικα σε δύο αρχεία ώστε το ένα να χειρίζεται τη λογική ροή, `admin.reviews.php` και το άλλο την παραγωγή του HTML κώδικα, `admin.reviews.html.php`. Στο αρχείο `admin.reviews.php` γράφουμε τον παρακάτω κώδικα.

```

<?php
    defined( '_JEXEC' ) or die( 'Restricted access' );
    require_once( JApplicationHelper::getPath( 'admin_html' ) );
    JTable::addIncludePath( JPATH_COMPONENT_DS . 'tables' );
    switch( $task )
    {
        case 'add':
            editReview( $option );

```



```

        break;
    }
    function editReview( $option )
    {
        $row =& JTable::getInstance('Review', 'Table');
        $lists = array();
        $reservations = array(
            '0' => array('value' => 'None Taken',
                'text' => 'None Taken'),
            '1' => array('value' => 'Accepted',
                'text' => 'Accepted'),
            '2' => array('value' => 'Suggested',
                'text' => 'Suggested'),
            '3' => array('value' => 'Required',
                'text' => 'Required'),
        );
        $lists['reservations'] = JHTML::_('select.genericList', $reservations, 'reservations',
            'class="inputbox" ', 'value', 'text', $row->reservations );
        $lists['smoking'] = JHTML::_('select.booleanlist', 'smoking', 'class="inputbox"', $row->smoking);
        $lists['published'] = JHTML::_('select.booleanlist', 'published', 'class="inputbox"', $row->published);
        HTML_reviews::editReview($row, $lists, $option);
    }
?>

```

Χρησιμοποιούμε την μέθοδο `require_once(JFactory::getPath('admin_html'))`; ώστε να συμπεριληφθεί το αρχείο `admin.reviews.html.php`. Η συνάρτηση `getPath()` παίρνει συγκεκριμένα strings και επιστρέφει τα απόλυτα paths για τα αντίστοιχα αρχεία του component. Η `require_once()` αναλαμβάνει να προσθέσει μια μόνο φορά το αρχείο. Στη συνέχεια η `addIncludePath()` (member function της κλάσης `JTable`) αυτόματα συμπεριλαμβάνει όλες τις κλάσεις που έχουμε καθορίσει στα αρχεία που βρίσκονται στον κατάλογο `tables`. Με την `JPATH_COMPONENT_DS` καθορίζεται το path του component και το DS είναι ο διαχωριστής των καταλόγων.

Με τη δήλωση `switch()` ελέγχεται η μεταβλητή `&task` και διαλέγει την απαραίτητη συνάρτηση. Τέλος η συνάρτηση `editReview()` προετοιμάζει κάποια HTML στοιχεία για την `display function HTML_reviews::editReview()`. Στο αρχείο `admin.reviews.html.php` προσθέτομε τον παρακάτω κώδικα.

```

<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
class HTML_reviews
{
    function editReview( $row, $lists, $option )
    {
        $editor =& JFactory::getEditor();
        JHTML::_('behavior.calendar');
        ?>
        <form action="index.php" method="post" name="adminForm" id="adminForm">

```

```

<fieldset class="adminform">
<legend>Details</legend>
<table class="admintable">
<tr>
  <td width="100" align="right" class="key">Restaurant Name:
  </td>
  <td>
    <input class="text_area" type="text" name="restaurant_name"
      id="name" size="50" maxlength="250"
      value="<?php echo $row->restaurant_name;?>" />
  </td>
</tr>
<tr>
  <td width="100" align="right" class="key">
    Address:
  </td>
  <td>
    <input class="text_area" type="text" name="address"
      id="address" size="50" maxlength="250"
      value="<?php echo $row->address;?>" />
  </td>
</tr>
<tr>
  <td width="100" align="right" class="key">
    Reservations:
  </td>
  <td>
    <?php
      echo $lists['reservations'];
    ?>
  </td>
</tr>
<tr>
  <td width="100" align="right" class="key">
    Quick Note:
  </td>
  <td>
    <?php
      echo $editor->display( 'quick_note', $row->quick_note ,
        '100%', '150', '40', '5' );
    ?>
  </td>
</tr>
<tr>

```

```
<td width="100" align="right" class="key">
  Review:
</td>
<td>
<?php
  echo $editor->display( 'review', $row->review ,
    '100%', '250', '40', '10' );
?>
</td>
</tr>
<tr>
<td width="100" align="right" class="key">
  Notes:
</td>
<td>
<textarea class="text_area" cols="20" rows="4"
  name="notes" id="notes" style="width:500px"><?php echo
  $row->notes; ?></textarea>
</td>
</tr>
<tr>
<td width="100" align="right" class="key">
  Smoking:
</td>
<td>
<?php
  echo $lists['smoking'];
?>
</td>
</tr>
<tr>
<td width="100" align="right" class="key">
  Credit Cards:
</td>
<td>
<input class="text_area" type="text" name="credit_cards"
  id="credit_cards" size="50" maxlength="250"
  value="<?php echo $row->credit_cards;?>" />
</td>
</tr>
<tr>
<td width="100" align="right" class="key">
  Cuisine:
</td>
```

```

        <td>
        <input class="text_area" type="text" name="cuisine"
        id="cuisine" size="31" maxlength="31"
        value="<?php echo $row->cuisine;?>" />
        </td>
    </tr>
    <tr>
        <td width="100" align="right" class="key">
            Average Dinner Price:
        </td>
        <td>
            <strong>Euro</strong><input class="text_area" type="text"
            name="avg_dinner_price" id="avg_dinner_price" size="5" maxlength="3"
            value="<?php echo $row->avg_dinner_price;?>" />
        </td>
    </tr>
    <tr>
        <td width="100" align="right" class="key">
            Review Date:
        </td>
        <td>
            <input class="inputbox" type="text" name="review_date"
            id="review_date" size="25" maxlength="19"
            value="<?php echo $row->review_date; ?>" />
            <input type="reset" class="button" value="..."
            onclick="return showCalendar('review_date', 'y-mm-dd');" />
        </td>
    </tr>
    <tr>
        <td width="100" align="right" class="key">
            Published:
        </td>
        <td>
            <?php
            echo $lists['published'];
            ?>
        </td>
    </tr>
</table>
</fieldset>
<input type="hidden" name="id"
value="<?php echo $row->id; ?>" />
<input type="hidden" name="option"
value="<?php echo $option;?>" />

```

```

<input type="hidden" name="task"
value="" />
</form>
<?php
}
}
?>

```

Η συνάρτηση editReview() παίρνει σαν αντικείμενα μια σειρά του πίνακα της βάσης, μια μήτρα (array) από HTML snippets (κομμάτια κώδικα που επαναλαμβάνονται) και το όνομα του component ώστε να παράγει την παρακάτω εικόνα. Πριν δώσει το τελικό αποτέλεσμα συμπεριλαμβάνει και μερικά κομμάτια βοηθητικού κώδικα για επιπλέον στοιχεία στο interface με τη βοήθεια της JFactory::getEditor() παίρνει σαν αντικείμενο τον text editor που έχουμε διαλέξει από τις γενικές ρυθμίσεις του administration panel. Ακριβώς από κάτω υπάρχει η JHTML::_('behavior.calendar') η οποία προσθέτει στο header τα javascript και τα CSS που είναι απαραίτητα για το ημερολόγιο που χρησιμοποιείται στο Review Date.

The screenshot shows the Joomla! Admin interface for editing a review. The form is organized into several sections:

- Restaurant Information:** Fields for 'Restaurant Name', 'Address', and 'Reservations' (set to 'None Taken').
- Review Editor:** A WYSIWYG editor for the review text, featuring a rich toolbar with options for bold, italic, underline, text color, background color, font family, font size, bulleted list, numbered list, link, unlink, image, and media. Below the editor is a 'Path: p' field and buttons for 'Image', 'Pagebreak', 'More...', 'JComments ON', 'JComments OFF', and 'Toggle editor'.
- Notes:** A simple text input field.
- Smoking:** Radio buttons for 'No' (selected) and 'Yes'.
- Credit Cards:** A text input field.
- Cuisine:** A text input field.
- Average Dinner Price:** A text input field with a 'Euro' symbol.
- Review Date:** A date selection field with a calendar icon.

Εικόνα: Admin interface του component.

13.4 Επεξεργασία των δεδομένων.

Αφότου συμπληρώσουμε την φόρμα με δεδομένα και στη συνέχεια ο Admin πατήσει το εικονίδιο save θα πρέπει να σωθούν τα δεδομένα στη βάση μας. Γράφουμε τον κώδικα της συνάρτησης saveReview() στο αρχείο admin_reviews.php

```
function saveReview( $option )
{
    global $mainframe;
    $row =& JTable::getInstance('review', 'Table');
    if (!$row->bind(JRequest::get('post')))
    {
        echo "<script> alert('".$row->getError()."'); window.history.go(-1); </script>\n";
        exit();
    }
    $row->quick_note = JRequest::getVar( 'quick_note', "", 'post',
    ' string', JREQUEST_ALLOWRAW );
    $row->review = JRequest::getVar( 'review', "", 'post',
    'string', JREQUEST_ALLOWRAW );
    if (!$row->review_date)
    $row->review_date = date( 'Y-m-d H:i:s' );
    if (!$row->store())
    {
        echo "<script> alert('".$row->getError()."');
        window.history.go(-1); </script>\n";
        exit();
    }
    $mainframe->redirect('index.php?option=' . $option, 'Review Saved');
}
```

Δηλώνουμε την μεταβλητή &mainframe που είναι τύπου global η οποία ενσωματώνει πολλές μεθόδους για να ελέγχουμε session μεταβλητές. Επίσης η &row αποτελεί ένα αντικείμενο της κλάσης TableReview το όνομα της οποίας τοποθετείται στις παραμέτρους της getInstance() με τη δεύτερη παράμετρο να ενεργεί σαν πρόθεμα για την πρώτη. Στη συνέχεια καλούμε την συνάρτηση bind() για να φορτώσει όλες τις μεταβλητές από την φόρμα στο αντικείμενο. Η bind() παίρνει σαν παράμετρο μια μήτρα και προσπαθεί να ταιριάξει όλα τα στοιχεία σε member μεταβλητές του αντικειμένου και καλώντας την JRequest::get() απομονώνει τις τιμές που παίρνει από την &_POST. Σε περίπτωση που για άγνωστο λόγο αποτύχει η bind() εμφανίζεται ένα παράθυρο (Javascript Alert) και επιστρέφει ο χρήστης στην προηγούμενη εικόνα.

Από τη στιγμή που τα πεδία quick_note και review δέχονται περιεχόμενο HTML η bind() χρειάζεται ειδική μεταχείριση διότι αυτόματα αποδίδει τον HTML κώδικα. Γι αυτό χρησιμοποιούμε την συνάρτηση getVar() της κλάσης JRequest περνώντας της το όνομα της φόρμας την default τιμή τη ζητούμενη μήτρα από την οποία θα τραβήξουμε δεδομένα, τη αναμενόμενη διαμόρφωση (format) και το JREQUEST_ALLOWRAW σημάδι αντίστοιχα.

Τέλος καλούμε την συνάρτηση store() η οποία αγνοεί όλες μεταβλητές ,του αντικειμένου jTable είναι μηδενικές από προεπιλογή και τις υπόλοιπες η τις ανανεώνει (update) ή κάνει insert και αυτό εξαρτάται από την τιμή του id. Δηλαδή αν γράφουμε ένα αρχείο για πρώτη φορά δεν θα υπάρχει τιμή για το id έτσι θα κατασκευαστεί ένα insert query για να καταχωρηθούν τα δεδομένα. Εάν δεν υπάρξει πρόβλημα με τον κώδικα της sql αναλαμβάνει η συνάρτηση redirect() της &mainframe και στέλνει τον χρήστη στην κεντρική σελίδα του component έχοντας και ένα μήνυμα επιβεβαίωσης των ρυθμίσεων.

Στο αρχείο admin.reviews.php προθέτουμε τον παρακάτω κώδικα κάτω από το switch() για να ολοκληρωθεί η διαδικασία της επεξεργασίας.

```
switch (&task)
{
    "
    "
Case 'save':
saveReview( $option );
break;
}
```

Εφόσον όλα έχουν κυλήσει σωστά θα πρέπει μέσα στο πίνακα jos_reviews να υπάρχουν καταχωρημένα τα στοιχεία όπως φαίνονται μέσω του phpMyAdmin στην επόμενη εικόνα.

ptions	id	restaurant_name	address	reservations	quick_note	review	notes	smoking	credit_cards	cuisine
	11	ZEPHYROS Ledra Marriott	Λεωφ. Συγγρού 115 (Ledra Marriott Hotel)	Accepted	<p>Address: Λεωφ. Συγγρού 115 (Le...	<p>setQuery($query);				
 $rows = $db->loadObjectList();
 if ($db->getErrorNum())
 {
 echo $db->stderr();
 return false;
 }
}
```

```
HTML_reviews::showReviews($option, $rows);
}
```

Η συνάρτηση showReviews φορτώνει τα δεδομένα που πρέπει να παρουσιαστούν. Πρώτα αναφέρουμε τη σύνδεση με την βάση και στη συνέχεια καλούμε την συνάρτηση τύπου member setQuery() η οποία παίρνει μια ακολουθία χαρακτήρων και την αποθηκεύει για να χρησιμοποιηθεί αργότερα. Στη συνέχεια καλείται η loadObjectList() το προηγούμενο setquery εκτελείται και οι ξεχωριστές γραμμές (rows) στο αποτέλεσμα φορτώνονται σε μια μήτρα (array) σαν αντικείμενο. Εάν το query αποτύχει επιστρέφει την τιμή false και παρουσιάζεται η λέξη λάθος (error). Όταν το query εκτελεσθεί χωρίς λάθη περνάμε την μήτρα με τα αποτελέσματα στην παρακάτω συνάρτηση και την γράφουμε στο αρχείο admin.reviews.html.php:

```
function showReviews($option, &$rows)
{
 ?>
 <form action="index.php" method="post" name="adminForm">
 <table class="adminlist">
 <thead>
 <tr>
 <th width="20">
 <input type="checkbox" name="toggle"
 value="" onclick="checkAll(<?php echo
 count($rows); ?>);" />
 </th>
 <th class="title">Restaurant Name</th>
 <th width="15%">Address</th>
 <th width="10%">Reservations</th>
 <th width="10%">Cuisine</th>
 <th width="10%">Credit Cards</th>
 <th width="5%" nowrap="nowrap">Published</th>
 </tr>
 </thead>
 <?php
 jimport('joomla.filter.output');
 $k = 0;
 for ($i=0, $n=count($rows); $i < $n; $i++)
 {
 $row = &$rows[$i];
 $checked = JHTML::_('grid.id', $i, $row->id);
 $published = JHTML::_('grid.published', $row, $i);
 $link = JFilterOutput::ampReplace('index.php?option=' .
 $option . '&task=edit&cid[]=' . $row->id);
 }
 ?>
 <tr class="<?php echo "row$k"; ?>">
 <td>
 <?php echo $checked; ?>

```



```

</td>
<td>
 <a href="<?php echo $link; ?>">
 <?php echo $row->restaurant_name; ?>
 </td>
<td>
 <?php echo $row->address; ?>
</td>
<td>
 <?php echo $row->reservations; ?>
</td>
<td>
 <?php echo $row->cuisine; ?>
</td>
<td>
 <?php echo $row->credit_cards; ?>
</td>
<td align="center">
 <?php echo $published;?>
</td>
</tr>
<?php
$k = 1 - $k;
}
?>
</table>
<input type="hidden" name="option"
 value="<?php echo $option;?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="checked" value="0" />
</form>
<?php
}

```

Στην αρχή του κώδικα αυτής της συνάρτησης καθορίζεται μια φόρμα που δείχνει στο αρχείο index.php και έχει το όνομα adminform ( για αναφορές σε Javascript). Στη συνέχεια δημιουργούμε ένα πίνακα, ο οποίος σχετίζεται με την κλάση adminlist που βρίσκεται στο αρχείο general.css ( γραμμή 401 ) και προστίθενται οι επικεφαλίδες σε συγκεκριμένα κελία. Εκτός από την πρώτη επικεφαλίδα, που είναι ένα κουτάκι (checkbox) το οποίο αν το κλικάρουμε αυτομάτως επιλέγει όλες τις καταγραφές που έχουμε κάνει, όλες οι άλλες είναι τυπικές.

Αμέσως μετά αρχίζει ένας βρόχος ( loop ) γύρω από την &row, Οι μεταβλητές &i και &n μηδενίζονται, αντίστοιχα και οι αριθμοί των γραμμών, ο βρόχος συνεχίζεται μέχρι τη στιγμή που δεν υπάρχουν άλλες καταγραφές στη &row για να παρουσιαστούν. Εναλλάσσουμε την τιμή της &k μεταξύ 0 και 1 όπου αφορούν δύο κλάσεις που έχουν ελάχιστες διαφορές την row0 και row1 που βρίσκονται στο general.css ( γραμμή 418 ) αρχείο.

Η JHTML::('grid.id') ενσωματώνει τον HTML κώδικα για τα checkboxes και η συνάρτηση JHTML::(:grid.published;) παράγει μια εικόνα που βασίζεται στην τιμή της μεταβλητής «Δημιουργία ιστότοπου με την χρήση cms»

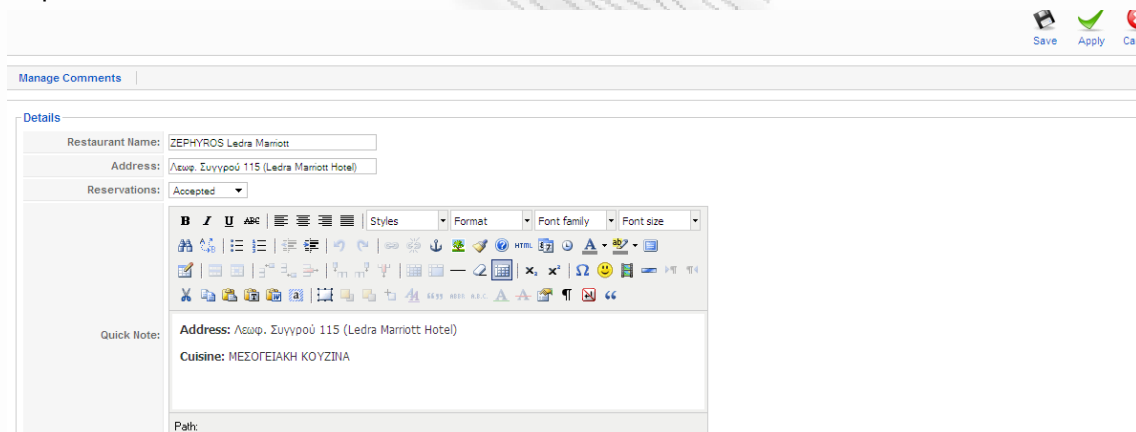
&published στο αντικείμενο row. Αν η τιμή είναι 1 τότε απεικονίζει μια εικόνα με ένα check και αν είναι 0 ένα "x". Χρειάζεται επίσης να παρέχουμε link στα οποία ο χρήστης θα μπορεί να κάνει κλικ και να τροποποιεί όσες καταγραφές υπάρχουν. Όταν έχουμε URL στον κώδικα μας που περιέχει ampersands(&) πολλοί browsers καταλήγουν σε error. Γι αυτό και εμείς για να μην έχουμε προβλήματα και για να υπάρχει συμβατότητα με την XHTML χρησιμοποιούμε την συνάρτηση ampReplace(). Ανήκει στην κλάση JFilterOutput και φορτώνεται όταν καλούμε την jimport('joomla.filter.output').

Στο τέλος του κώδικα της συνάρτησης υπάρχουν τέσσερις κρυφές ( hidden ) μεταβλητές. Η μια διατηρεί την τιμή της &option ώστε να είμαστε στο σωστό component. Η task γίνεται διαθέσιμη για να μπορεί το Javascript στην γραμμή εργαλείων να ενεργοποιηθεί πριν την κατοχύρωση της φόρμας. Όταν όλα τα checkboxes εναλλάσσονται τότε η boxchecked παίρνει την τιμή 1. Μηδενίζεται όταν όλα τα checkboxes είναι κενά.

Για να επεξεργαστούμε τις κριτικές καλύτερα γραφούμε 3 επιπλέον γραμμές κώδικα στην συνάρτηση editReview που βρίσκεται στο αρχείο admin.reviews.php

```
$row =& JTable::getInstance('review', 'Table');
$cid = JRequest::getVar('cid', array(0), '', 'array');
$id = $cid[0];
$row->load($id);
```

Παίρνουμε ένα αντικείμενο από την κλάση TableReview για να χειρίζεται τα δεδομένα της καταγραφής. Επίσης τραβάμε την μεταβλητή cid από την φόρμα, η οποία είναι μια μήτρα με τα ID των καταγραφών. Από την στιγμή που θέλουμε να επεξεργαστούμε μια καταγραφή την φορά επιλέγουμε το πρώτο ID και φορτώνουμε το αντίστοιχο αντικείμενο. Οι καταγραφές μας τώρα θα είναι σε μορφή link αν κλικάρουμε πάνω σε μία από αυτές θα παρουσιαστεί η παρακάτω εικόνα:



Εικόνα: Οθόνη επεξεργασίας δεδομένων (edit screen).

Όπως παρατηρούμε στην παραπάνω εικόνα υπάρχει ένα κουμπί με την ονομασία apply. Πατώντας το μπορούμε να σώσουμε την δουλειά μας και να συνεχίζουμε την επεξεργασία βρισκόμενοι πάλι στην ίδια οθόνη. Για να γίνει αυτό τροποποιούμε την switch () στο αρχείο admin.review.php

```
case 'apply':
case 'save':
saveReview($option, $task);
break;
```

Επιπλέον προσθέτουμε τον παρακάτω κώδικα στην συνάρτηση SaveReview ( \$option, \$task ) που ελεγχει την μεταβλητή \$task:

```
switch ($task)
```

```

{
case 'apply':
$msg = 'Changes to Review saved';
$link = 'index.php?option=' . $option .
'&task=edit&cid]=' . $row->id;
break;
case 'save':
default:
$msg = 'Review Saved';
$link = 'index.php?option=' . $option;
break;
}
$mainframe->redirect($link, $msg);

```

### 13.6 Κατάργηση των δεδομένων μας.

Στο αρχείο admin.reviews.php προσθέτουμε την συνάρτηση removeReviews.

```

function removeReviews($option)
{
global $mainframe;
$cid = JRequest::getVar('cid', array(), "", 'array');
$db =& JFactory::getDBO();
if(count($cid))
{
$cids = implode(',', $cid);
$query = "DELETE FROM #__reviews WHERE id IN ($cids)";
$db->setQuery($query);
if (!$db->query())
{
echo "<script> alert('".$db->getErrorMessage()."');
window.history.go(-1); </script>\n";
}
}
$mainframe->redirect('index.php?option=' . $option);
}

```

Εξάγουμε την μεταβλητή cid από τη φόρμα και ελέγχουμε εάν υπάρχει κάποιο id στην μήτρα. Εάν υπάρχει φτιάχνουμε ένα string που να χωρίζει τα id με κόμματα και το χρησιμοποιούμε για την δημιουργία ενός delete query. Εφόσον όλα πάνε καλά κατευθύνεται ο χρήστης πίσω στην οθόνη με τις καταγραφές των δεδομένων. Στην switch () προσθέτουμε

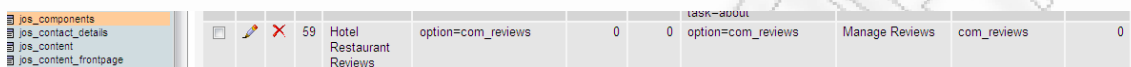
```

case 'remove':
removeReviews($option);
break;

```

### 13.7 Δημιουργία και διαχείριση σχολίων.

Για να προσφέρουμε την δυνατότητα στους χρήστες του ιστότοπου μας να γράφουν τα σχόλια τους για τα εστιατόρια των ξενοδοχείων θα πρέπει να στήσουμε στο παρασκήνιο ( Back end ) μια οθόνη διαχείρισης, όπου θα μπορούμε να επεξεργαζόμαστε και να αφαιρούμε τα σχόλια. Το πρώτο που πρέπει να κάνουμε είναι να πάμε στη βάση και να προσθέσουμε ένα link κάτω από το Hotel Restaurant Reviews στο μενού component. Για να δημιουργήσουμε το insert query θα πρέπει να βρούμε το id για το link του component Hotel Restaurant Reviews που είχαμε εγγράψει στη βάση μας σε προηγούμενο κεφάλαιο. Αυτό το βρίσκουμε αν πάμε στο PhpMy Admin στο όνομα της βάσης μας και ψάχνουμε στο jos\_components. Στη δική μας περίπτωση το id έχει την τιμή 59.



| jos_components | option             | parent                   | admin_menu_link    | admin_menu_alt | ordering    |
|----------------|--------------------|--------------------------|--------------------|----------------|-------------|
| 59             | option=com_reviews | Hotel Restaurant Reviews | option=com_reviews | Manage Reviews | com_reviews |

**Εικόνα: PhpMy Admin id number.**

Στην συνέχεια είτε γράφουμε τον κώδικα για το query είτε χρησιμοποιούμε το interface του PhpMy Admin.

```
INSERT INTO jos_components (name, parent, admin_menu_link,
 admin_menu_alt, ordering)
VALUES (Manage Comments, 59,
 option=com_reviews&task=comments, Manage Comments, 1);
```

Επίσης γράφουμε και το query για την εισαγωγή του πίνακα στη βάση μας όπου εκεί θα αποθηκεύονται τα σχόλια των χρηστών.

```
CREATE TABLE jos_reviews_comments(
 id int(11) NOT NULL auto_increment,
 review_id int(11) NOT NULL,
 user_id int(11) NOT NULL,
 full_name varchar(50) NOT NULL,
 comment_date datetime NOT NULL,
 comment_text text NOT NULL,
 PRIMARY KEY (id)
);
```

Αφού προσθέσαμε το link πρέπει τώρα να το κατευθύνουμε για να δείχνει σε μια οθόνη που θα επεξεργαζόμαστε τα σχόλια. Γράφουμε λοιπόν την παρακάτω συνάρτηση στο αρχείο admin.reviews.php

```
function comments()
{
 global $option, $mainframe;
 $db =& JFactory::getDBO();
 $query = "SELECT count(*) FROM #__reviews_comments";
 $db->setQuery($query);
 $total = $db->loadResult();
 $query = "SELECT c.*, r.restaurant_name FROM #__reviews_comments
 AS c LEFT JOIN #__reviews AS r ON r.id = c.review_id ";
```

```

$rows = $db->loadObjectList();
if ($db->getErrorNum())
{
 echo $db->stderr();
 return false;
}
HTML_reviews::showComments($option, $rows,);
}

```

Στη συνάρτηση αυτή συνδυάζουμε τον πίνακα των comments και τον πίνακα των reviews. Στο τέλος της συνάρτησης καλούμε την HTML\_reviews::showComments(); τον κώδικα της οποίας τον γράφουμε στο αρχείο admin.reviews.html.php

```

function showComments($option, &$rows,)
{
 ?>
 <form action="index.php" method="post" name="adminForm">
 <table class="adminlist">
 <thead>
 <tr>
 <th width="20">
 <input type="checkbox" name="toggle"
 value="" onclick="checkAll(<?php echo
 count($rows); ?>);" />
 </th>
 <th class="title">Review Name</th>
 <th width="15%">Commenter</th>
 <th width="20%">Comment Date</th>
 <th width="30%">Comment</th>
 </tr>
 </thead>
 <?php
 jimport('joomla.filter.output');
 $k = 0;
 for ($i=0, $n=count($rows); $i < $n; $i++) {
 $row = &$rows[$i];
 $checked = JHTML::_('grid.id', $i, $row->id);
 $link = JFilterOutput::ampReplace('index.php?option=' .
 $option . '&task=editComment&cid[]=' . $row->id);
 ?>
 <tr class="<?php echo "row$k"; ?>">
 <td><?php echo $checked; ?></td>
 <td><a href="<?php echo $link; ?>"><?php echo $row->restaurant_name;
 ?></td>
 <td><?php echo $row->full_name; ?></td>
 }
}

```

```

 <td><?php echo JHTML::Date($row->comment_date); ?></td>
 <td><?php echo substr($row->comment_text, 0, 149); ?></td>
 </tr>
<?php
 $k = 1 - $k;
}
?>
</table>
<input type="hidden" name="option"
 value="<?php echo $option;?>" />
<input type="hidden" name="task" value="comments" />
<input type="hidden" name="checked" value="0" />
</form>
<?php
}

```

Ο κώδικας της συνάρτησης αυτής είναι παρόμοιος με της συνάρτησης showReviews που γράψαμε παραπάνω. Καθορίζουμε μια φόρμα που δείχνει στο αρχείο index.php και έχει το όνομα adminform ( για αλληλεπίδραση με Javascript). Ο χώρος για τις επικεφαλίδες του πίνακα περιλαμβάνει ένα τετραγωνάκι το οποίο αν το κλικάρουμε ενεργοποιεί όλα τα τετραγωνάκια που αντιπροσωπεύουν την κάθε καταγραφή, επίσης υπάρχει το όνομα του review το όνομα του ατόμου που έκανε το σχόλιο, η ημερομηνία και το σχόλιο.

Στη συνέχεια με τη βοήθεια των κλάσεων του Joomla που τις εισάγουμε με την jimport('joomla.filter.output') βλέπουμε το αποτέλεσμα στην οθόνη μας ώστε να μετατρέψουμε τις όποιες κριτικές σε links για την επεξεργασία των σχολίων. . Εναλλάσσουμε την τιμή της &k μεταξύ 0 και 1 όπου αφορούν δύο κλάσεις που έχουν ελάχιστες διαφορές την row0 και row1 που βρίσκονται στο general.css αρχείο. Ακολουθείται από έναν βρόχο που διατρέχει όλες τις γραμμές του πίνακα comment της βάσης μας. Με την substr() περιορίζουμε το κείμενο των σχολίων ώστε ο admin να βλέπει μόνο ένα μέρος αυτού στην οθόνη. Το αποτέλεσμα αυτού του κώδικα φαίνεται στην παρακάτω εικόνα.

| Review Name             | Commenter     | Comment Date               | Comment        |
|-------------------------|---------------|----------------------------|----------------|
| ZEPHYROS Ledra Marriott | Administrator | Tuesday, 28 September 2010 | Πολύ καλό!     |
| ZEPHYROS Ledra Marriott | Nick          | Tuesday, 28 September 2010 | Nice cuisine!! |

**Εικόνα: Οθόνη διαχείρισης σχολίων.**

Αυτό που απομένει είναι να αναπτύξουμε τον κώδικα των συναρτήσεων οι οποίες θα είναι υπεύθυνες για την επεξεργασία, την αποθήκευση και την διαγραφή των σχολίων, Ο κώδικας είναι ο εξής:

```

function editComment()
{
 global $option;

```

```
$row =& jTable::getInstance('comment', 'Table');
$cid = JRequest::getVar('cid', array(0), "", 'array');
$id = $cid[0];
$row->load($id);
HTML_reviews::editComment($row, $option);
}
```

```
function saveComment()
{
 global $option;
 $row =& jTable::getInstance('comment', 'Table');
 if (!$row->bind(JRequest::get('post')))
 {
 echo "<script> alert('".$row->getError()."");
 window.history.go(-1);
 </script>\n";
 exit();
 }
 if (!$row->store()) {
 echo "<script> alert('".$row->getError()."");
 window.history.go(-1);
 </script>\n";
 exit();
 }
 $this->setRedirect('index.php?option=' . $option . '&task=comments',
 'Comment changes saved');
}
```

```
function removeComment()
{
 global $option;
 $cid = JRequest::getVar('cid', array(), "", 'array');
 $db =& JFactory::getDBO();
 if(count($cid))
 {
 $cids = implode(',', $cid);
 $query = "DELETE FROM #__reviews_comments WHERE id IN ($cids)";
 $db->setQuery($query);
 if (!$db->query())
 {
 echo "<script> alert('".$db->getErrorMessage()."");
 window.history.go(-1);
 }
 }
}
```

```

 </script>\n";
 }
}
$this->setRedirect('index.php?option=' . $option . '&task=comments');
}

```

Όπως παρατηρούμε στο τέλος της συνάρτησης editComment() καλείται η HTML\_reviews::editComment() ο κώδικας της οποίας τοποθετείται στο αρχείο admin.reviews.html.php

```

function editComment ($row, $option)
{
 JHTML::_('behavior.calendar');
 ?>
 <form action="index.php" method="post" name="adminForm" id="adminForm">
 <fieldset class="adminform">
 <legend>Comment</legend>
 <table>
 <tr>
 <td width="100" align="right" class="key">
 Name:
 </td>
 <td>
 <input class="text_area" type="text" name="full_name"
 id="full_name" size="50" maxlength="250" value="<?php echo
 $row->full_name;?>" />
 </td>
 </tr>
 <tr>
 <td width="100" align="right" class="key">
 Comment:
 </td>
 <td>
 <textarea class="text_area" cols="20" rows="4"
 name="comment_text" id="comment_text"
 style="width:500px"><?php echo $row->comment_text;?>
 </textarea>
 </td>
 </tr>
 <tr>
 <td width="100" align="right" class="key">
 Comment Date:
 </td>
 <td>
 <input class="inputbox" type="text" name="comment_date"

```



```

 id="comment_date" size="25" maxlength="19" value="<?php echo
 $row->comment_date; ?>" />
 <input type="reset" class="button" value="..."
 onclick="return showCalendar('comment_date', 'y-mm-dd');" />
 </td>
</tr>
</table>
</fieldset>
<input type="hidden" name="id" value="<?php echo $row->id; ?>" />
<input type="hidden" name="option" value="<?php echo $option; ?>" />
<input type="hidden" name="task" value="" />
</form>
<?php
}

```

Φτιάχνουμε μια φόρμα(form) με όνομα adminform και ένα πεδίο το οποίο θα διαμορφωθεί από τα CSS που χρησιμοποιούνται για το Administrator Back End. Χρησιμοποιούμε μια απλή εισαγωγή (input) για το πεδίο name και την κλάση text\_area από τα CSS. Η ίδια κλάση ευθύνεται και για το πεδίο που γράφεται το σχόλιο. Τέλος η ημερομηνία ανήκει στο πεδίο με την inputbox ενώ το κουμπι που εμφανίζει το ημερολόγιο ανήκει στην κλάση button. Πατώντας το κουμπί καλείται η συνάρτηση Javascript showCalendar() που αλληλεπιδρά με το πεδίο comment\_date και κρατάει την ημερομηνία με τη παρακάτω μορφή y-mm-dd.

Αυτό που μένει τώρα είναι να φτιάξουμε τα κουμπιά που θα αντιπροσωπεύουν τις συναρτήσεις για την επεξεργασία αποθήκευση και κατάργηση των σχολίων. Επομένως στο αρχείο toolbar.reviews.html.php γράφουμε την ακόλουθη κλάση.

```

class TOOLBAR_reviews_comments
{
 function _EDIT()
 {
 JToolBarHelper::save('saveComment');
 JToolBarHelper::cancel('comments');
 }
 function _DEFAULT()
 {
 JToolBarHelper::title(JText::_('Comments'), 'generic.png');
 JToolBarHelper::editList('editComment');
 JToolBarHelper::deleteList("Are you sure you want to remove these comments?",
 "removeComment", "Delete");
 }
}

```

Η πρώτη παράμετρος στις save(), cancel() editlist() που είναι μέλη της JToolBarHelper διαγράφει την προεπιλογή και μας επιτρέπει να την καθορίσουμε με ότι θέλουμε. Με την title εμφανίζεται ο τίτλος και μια φωτογραφία στα αριστερά του μενού comment. Η deleteList έχει σαν πρώτη παράμετρο ένα ολόκληρο μήνυμα επιβεβαίωσης και σαν δεύτερη την διεργασία που πρέπει να εκτελεσθεί. Το μόνο που απομένει είναι να προσθέσουμε ορισμένες εντολές στην switch () που ελέγχει όλες αυτές τις διεργασίες και βρίσκεται στο αρχείο toolbars.reviews.php

```

switch ($task)
{
 “

```

```

“
“
case 'comments':
case 'saveComment':
case 'removeComment':
TOOLBAR_reviews_comments::_DEFAULT();
break;
case 'editComment':
TOOLBAR_reviews_comments::_EDIT();
break;
“
“
}

```



Εικόνα: comments toolbars.

## 14 Ανάπτυξη κώδικα για το προσκήνιο (front end) του component.

Έχοντας τελειώσει με τη δουλεία που κάναμε στο παρασκήνιο ( Back end ) έφτασε η στιγμή να δούμε πως θα φαίνονται στο χρήστη όλα αυτά που κάναμε. Για να γίνει αυτό βέβαια θα πρέπει να γράψουμε ακόμη ορισμένα κομμάτια κώδικα. Δημιουργούμε ένα αρχείο με το όνομα reviews.php και το τοποθετούμε στον συγκεκριμένο κατάλογο /components/com\_reviews .

```

<?php
defined('_JEXEC') or die('Restricted access');
JHTML::stylesheet('styles.css','components/com_reviews/css/');
echo '<div class="componentheading">Hotel Restaurant Reviews</div>';
jimport('joomla.application.helper');
require_once(JApplicationHelper::getPath('html'));
JTable::addIncludePath(JPATH_ADMINISTRATOR.DS.
'components'.DS.$option.DS.'tables');
switch($task)
{
case 'view':
viewReview($option);
break;
case 'comment':
addComment($option);
break;

```

```

default:
showPublishedReviews($option);
break;
}
function showPublishedReviews($option)
{
$db = & JFactory::getDBO();
$query = "SELECT * FROM #__reviews WHERE
published = '1' ORDER BY review_date DESC";
$db->setQuery($query);
$rows = $db->loadObjectList();
if ($db->getErrorNum())
{
echo $db->stderr();
return false;
}
HTML_reviews::showReviews($rows, $option);
}

```

Η `require_once( JApplicationHelper::getPath( 'html' ) );` χρειάζεται για να συμπεριλάβει το αρχείο `reviews.html.php` που θα δημιουργήσουμε και θα εξηγήσουμε τον κώδικά του στη συνέχεια, Η `JTable::addIncludePath(JPATH_ADMINISTRATOR.DS.`

`'components'.DS.$option.DS.'tables');` τραβάει τα δεδομένα της κλάσης `table` από το back end. Η συνάρτηση `switch()` έχει τρεις λειτουργίες, για τις δύο πρώτες θα μιλήσουμε στη συνέχεια ενώ η τρίτη που είναι και προεπιλογή (`default`) καλεί την συνάρτηση `showPublishedReviews` που μας παρουσιάζει στην οθόνη μας όλες τις δημοσιοποιημένες κριτικές (`reviews`) με βάση την ημερομηνία ανάρτησης (από την νέα στην πιο παλιά) και αυτό οφείλεται στο `sql query` που τρέχει μέσα σε αυτή. Στο τέλος καλείται η κλάση `HTML_reviews` την οποία προσθέτουμε σε ξεχωριστό αρχείο `review.html.php` που είναι υπεύθυνο για τον HTML κώδικα.

```

<?php
class HTML_reviews
{
function showReviews($rows, $option)
{
?>
<table>
<?php
foreach($rows as $row)
{
$link = 'index.php?option=' .
$option . '&id=' . $row->id . '&task=view';
echo
'

' . $row->restaurant_name . '


```

```

 ';
 }
 ?>
</table>
<?php
}
}
?>

```

Η showReviews παίρνει σαν παραμέτρους δεδομένα της βάσης και το όνομα του component. Ακολουθεί ένας πίνακας και στη συνέχεια ένας βρόχος δημιουργεί link στα δεδομένα που έχουν βρεθεί. Στην παρακάτω εικόνα βλέπουμε πως φαίνονται οι κριτικές:



Εικόνα: Οι λίστες με τις κριτικές στο front end .

Κάνοντας κλικ πάνω σε κάποιο από τα link θα πρέπει να μεταφερθούμε σε μια άλλη σελίδα για να μπορέσουμε να δούμε τις κριτικές που για το συγκεκριμένο εστιατόριο. Θα πρέπει να προσθέσουμε στον κώδικα του αρχείου reviews.php την παρακάτω συνάρτηση.

```

function viewReview($option)
{
 $id = JRequest::getVar('id', 0);
 $row =& JTable::getInstance('review', 'Table');
 $row->load($id);
 if(!$row->published)
 {
 JError::raiseError(404, JText::_('Invalid
 ID provided'));
 }
 HTML_reviews::showReview($row, $option);
}

```

Η getVar() φέρνει το id που ζητείται επιπλέον εξασφαλίζει το γεγονός πως η μεταβλητή μας δεν θα δεχθεί διαφόρων ειδών επιθέσεις. Αυτό έχει σημασία όταν έχουμε το site μας ανεβασμένο στο δίκτυο και δεν θέλουμε τα δεδομένα από τη βάση μας να γίνουν στόχος από κακόβουλα άτομα. Εάν η τιμή του id λείπει θα χρησιμοποιηθεί η προεπιλογή του 0 που είναι η δεύτερη παράμετρος. Η getInstance () φέρνει ένα αντικείμενο της κλάσης table από το back end. Αφού φορτώνεται το αντικείμενο που αντιστοιχεί στο συγκεκριμένο id γίνεται ένας γρήγορος έλεγχος για να σιγουρευτούμε πως το id έχει δημοσιευτεί. Εάν δεν συμβεί αυτό η συνάρτηση raiseError () της κλάσης JError φανερώνει το μήνυμα **404 - Page could not be found**



**Εικόνα: 404 - Page could not be found.**

Για να δουλέψει σωστά η συνάρτηση vieReview() είναι απαραίτητος και ο κώδικας που έχουμε γράψει παραπάνω για την συνάρτηση switch(). Επίσης θα προσθέσουμε στην κλάση που βρίσκεται στο αρχείο reviews.html.php και την συνάρτηση showReview() που αναλαμβάνει τον HTML κώδικα.

```
function showReview($row, $option)
{
 ?>
 <p class="contentheading"><?php echo $row->restaurant_name; ?></p>
 <p class="createdate"><?php echo JHTML::Date
 ($row->review_date); ?></p>
 <p><?php echo $row->quick_note; ?></p>
 <p>Address: <?php echo $row->address; ?></p>
 <p>Cuisine: <?php echo $row->cuisine; ?></p>
 <p>Average dinner price: <?php echo
 $row->avg_dinner_price; ?> Euro</p>
 <p>Credit cards: <?php echo
 $row->credit_cards; ?></p>
 <p>Reservations: <?php echo
 $row->reservations; ?></p>
 <p>Smoking: <?php
 if($row->smoking == 0)
 {
 echo "No";
 }
 else
 {
 echo "Yes";
 }
 ?></p>
 <p><?php echo $row->review; ?></p>
 <p>Notes: <?php echo $row->notes; ?></p>
 <?php $link = 'index.php?option=' . $option ; ?>
 <a href="<?php echo $link; ?>">< return to the reviews
 «Δημιουργία ιστότοπου με την χρήση cms»
```

```
<?php
}
```

Η `showReview()` παίρνει σαν παραμέτρους μια σειρά του πίνακα της βάσης σαν αντικείμενο και το όνομα του component. Οι κλάσεις `contentheading` και `createdate` αναφέρονται στον τρόπο που θα εμφανιστούν αυτά που γράφουμε στα HTML tags και παίρνουν τις πληροφορίες από το αρχείο `template_css.css`. Με αυτό τον τρόπο καταφέρνουμε το component που φτιάχνουμε να έχει ομοιόμορφο τρόπο εμφάνισης με το υπόλοιπο site. Η στήλη `smoking` ελέγχεται και τοποθετείται κατάλληλα σε `yes` ή `no` ανάλογα με την επιλογή που έχουμε κάνει.

The screenshot shows a web page for 'The BYZANTINO Restaurant Hilton'. On the left, there is a navigation menu with items like 'Αρχική', 'Ενοδοχεία', 'Δωμάτια', 'Ενοικιάσεις', 'Μέσα Μεταφοράς', 'Δελτίο Καιρού', 'Web Links', and 'Κριτικές'. Below the menu is a section titled 'Hotel Restaurant Reviews' which lists 'The BYZANTINO Restaurant Hilton' with its address, cuisine, phone number, and another restaurant 'ZEPHYROS Ledra Marriott'. The main content area displays details for 'The BYZANTINO Restaurant Hilton' including the date 'Tuesday, 12 October 2010', address 'Βασιλίσσης Σοφίας 46 (Hilton Hotel)', cuisine 'ΜΕΣΟΓΕΙΑΚΗ ΚΟΥΖΙΝΑ', phone '+30(0)210 7281400', and other information like 'Average dinner price: 32 Euro', 'Credit cards: Visa, Diners', 'Reservations: Accepted', and 'Smoking: No'. On the right, there is a review text: 'Βρεθείτε ανάμεσα σε διάσημους Αθηναίους, αλλά και σε πελάτες του ξενοδοχείου, σ' αυτό το ιστορικό εστιατόριο της Αθήνας. Σημείο αναφοράς στο γευστικό χάρτη της πόλης, το εστιατόριο Byzantino του Hilton Αθηνών προσφέρει πλούσιο μπουφέ με αυθεντικές Μεσογειακές γεύσεις, μενού αλά καρτ και το πιο πλούσιο και δημοφιλέ Κυριακάτικο Brunch της Αθήνας (σεβίρεται από τις 12:30 έως τις 16:00).'. Below the text is a photo of the restaurant interior and a caption: 'Χαλαρώστε σ'ένα κομψό περιβάλλον με λεπτομέρειες από ξύλο και γυαλί, απολαύστε τα αρώματα που αναδύονται από την ανοιχτή κουζίνα κι επιλέξτε νόστιμες Ελληνικές σπεσιαλιτέ και άλλες Μεσογειακές γεύσεις.'

Εικόνα: Εμφάνιση μιας κριτικής για ένα εστιατόριο.

#### 14.1 Κώδικας για τον τρόπο που θα παρουσιάζονται τα σχόλια.

Για να μπορέσουμε να διαχειριστούμε τα δεδομένα που καταχωρούνται στον πίνακα για τα σχόλια θα πρέπει να φτιάξουμε μια κλάση που θα έχει την ευθύνη όλων των βασικών συναρτήσεων, που θα γράψουμε για τα σχόλια. Έτσι λοιπόν στο κατάλογο `administrator/components/com_reviews/tables` δημιουργούμε το αρχείο `comments.php` και προσθέτουμε τον παρακάτω κώδικα:

```
<?php
defined('_JEXEC') or die('Restricted access');
class TableComment extends JTable
{
 var $id = null;
 var $review_id = null;
 var $user_id = null;
 var $full_name = null;
 var $comment_date = null;
 var $comment_text = null;
function __construct(&$db)
```

«Δημιουργία ιστότοπου με την χρήση cms»

```

{
 parent::__construct('#__reviews_comments', 'id', $db);
}
}
?>

```

Η TableComment είναι μια επέκταση της κλάσης JTable του Joomla. Όλες οι μεταβλητές θα πρέπει να αντιστοιχούν με τα ονόματα των στηλών του πίνακα. Τέλος δημιουργούμε ένα constructor που έχει σαν αναφορά ένα στιγμιότυπο της βάσης το οποίο καλεί ένα parent\_constructor που δέχεται σαν όρισμα το όνομα του πίνακα, το όνομα του βασικού κλειδιού και το στιγμιότυπο της βάσης. Απομένει να φτιάξουμε μια φόρμα στην οποία οι χρήστες του site να μπορούν να γράφουν τα σχόλια τους. Ο παρακάτω κώδικας μπαίνει στο αρχείο reviews.html.php.

```

function showCommentForm($option, $review_id, $name)
{
 ?>

 <form action="index.php" method="post">
 <table><h4>Add comment</h4>
 <tr>
 <td>
 Name:
 </td>
 <td>
 <input class="text_area" type="text" name="full_name"
 id="full_name" value="<?php echo $name; ?>" />
 </td>
 </tr>
 <tr>
 <td>
 Comment:
 </td>
 <td>
 <textarea class="text_area" cols="20" rows="4"
 name="comment_text" id="comment_text"
 style="width:500px">
 </textarea>
 </td>
 </tr>
 </table>
 <input type="hidden" name="review_id"
 value="<?php echo $review_id; ?>" />
 <input type="hidden" name="task"
 value="comment" />

```

```

<input type="hidden" name="option"
 value="<?php echo $option; ?>" />
<input type="submit" class="button" id="button"
 value="Submit" />
</form>
<?php
}

```

Η συνάρτηση `showCommentForm()` παίρνει σαν παραμέτρους το όνομα του component το `Id` του review που δείχνουμε και ένα όνομα. Η μεταβλητή `$name` παρέχει το όνομα του χρήστη που είναι εγγεγραμμένος στο site μας κατά αυτόν τον τρόπο κάποιος που έχει κάνει `log in` στο σύστημα δεν είναι απαραίτητο να γράψει το όνομα του στο πεδίο `name` αφού αυτό θα εμφανίζεται αυτόματα. Η μεταβλητή `task` παίρνει την τιμή `comment` ώστε να καταγράψει το σχόλιο στο component και παίρνουμε και το `review_id` για να ξέρουμε ότι αναφερόμαστε στο συγκεκριμένο review. Στο αρχείο `reviews.php` και συγκεκριμένα στην συνάρτηση `viewReview()` γράφουμε επιπλέον κώδικα ώστε να τοποθετήσουμε την φόρμα για τα σχόλια ακριβώς κάτω από το άρθρο.

```

$user =& JFactory::getUser();
if($user->name)
{
 $name = $user->name;
}
else
{
 $name = "";
}
HTML_reviews::showCommentForm($option, $id, $name);

```

Εάν ο χρήστης έχει κάνει `login` παίρνουμε το όνομα του με την γραμμή του κώδικα `$user =& JFactory::getUser();` και εν συνεχεία το μεταβιβάζουμε στη μεταβλητή `$name` αν δεν υπάρχει τότε η `&name` παραμένει κενή. Τέλος καλείται η `showCommentForm()`. Για να δουλέψει η φόρμα και να γίνει η αποθήκευση στη βάση, του σχολίου που έχουμε κάνει, θα πρέπει πατώντας το κουμπί `submit` να τρέχει κάποιος κώδικας που θα κάνει αυτή την δουλειά. Αυτό το καταφέρνουμε με την συνάρτηση `addComment()` που είναι η εξής:

```

function addComment($option)
{
 global $mainframe;
 $row =& JTable::getInstance('comment', 'Table');
 if (!$row->bind(JRequest::get('post')))
 {
 echo "<script> alert('".$row->getError()."');
 window.history.go(-1); </script>\n";
 exit();
 }
}

```



```

$row->comment_date = date('Y-m-d H:i:s');
$user =& JFactory::getUser();
if($user->id)
{
 $row->user_id = $user->id;
}
if (!$row->store())
{
 echo "<script> alert('".$row->getError()."');
 window.history.go(-1); </script>\n";
 exit();
}
$mainframe->redirect('index.php?option=' .
 $option . '&id=' . $row->review_id .
 '&task=view', 'Comment Added.');
```

Ο παραπάνω κώδικας είναι παρόμοιος με αυτόν που είχαμε γράψει για την προσθήκη δεδομένων στο πίνακα με τις κρητικές για το back end,

Message

- Comment Added.

#### Add comment

Name:

Comment:

Εικόνα: Εμφάνιση φόρμας και μηνύματος προσθήκης για τα σχόλια.

## 14.2 Εμφάνιση σχολίων.

Τα σχόλια που γράφει ο κάθε χρήστης θα φαίνονται ακριβώς πάνω από τη φόρμα προσθήκης. Σε αυτό θα φαίνεται το όνομα του χρήστη η ημερομηνία που γράφτηκε και το κείμενο. Στο αρχείο reviews.php προσθέτουμε τις παρακάτω γραμμές κώδικα:

```

$db =& JFactory::getDBO();
$db->setQuery("SELECT * FROM #__reviews_comments
WHERE review_id = '$id'");
$rows = $db->loadObjectList();
foreach($rows as $row)
{
 HTML_reviews::showComment($row);
```

}

Παίρνουμε όλα τα δεδομένα του πίνακα comment που αντιστοιχούν σε συγκεκριμένο review\_id και στη συνέχεια εμφανίζουμε το καθένα από αυτά. Τέλος καλείται η showComment () από το αρχείο reviews.html.php.

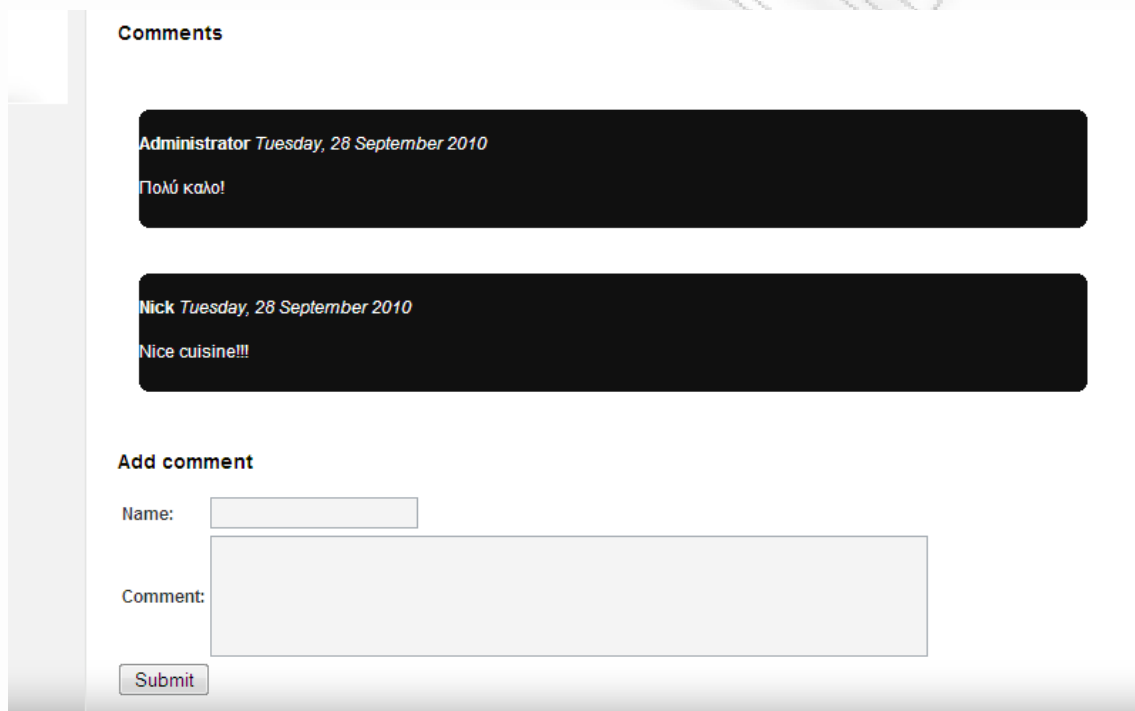
```
function showComment($row)
{
 ?>

 <div class="box">
 <b class="rtop"><b class="r1"> <b class="r2"> <b class="r3"> <b
class="r4">
 <p><?php echo $row->full_name;
 ?> <?php
 echo JHTML::Date($row->comment_date);
 ?></p>
 <p><?php echo $row->comment_text; ?></p>
 <b class="rbottom"><b class="r4"> <b class="r3"> <b class="r2"> <b
class="r1">
 </div>
 <?php
}
```

Στην αρχή του κώδικα στο αρχείο reviews.php έχει τοποθετηθεί η εντολή JHTML::stylesheet('styles.css','components/com\_reviews/css/'); που ενσωματώνει κανόνες css που βρίσκονται στον κατάλογο components/com\_reviews/css/. Με αυτόν τον τρόπο εμφανίζουμε τα σχόλια σε κυκλικά κουτιά. Το αρχείο styles.css περιλαμβάνει τον εξής κώδικα:

```
.box {
background: #101010 ;
color:#FFFFFF;
margin:0 15px;
text-align: left;
}
.rtop, .rbottom{
display:block;
background:#FFFFFF;
}
.rtop *, .rbottom *{
display: block;
height: 1px;
overflow: hidden;
background:#101010 ;
}
.r1{
```

```
margin: 0 5px
}
.r2{
margin: 0 3px
}
.r3{
margin: 0 2px
}
.r4{
margin: 0 1px;
height: 2px
}
```



Εικόνα: Παρουσίαση σχολίων.

### 14.3 Κώδικας Γεωκωδικοποίησης

Στο component που δημιουργήσαμε προσθέτουμε στο αρχείο reviews.html.php τον παρακάτω κώδικα με τον οποίο καταφέρνουμε να γεωκωδικοποιούμε την διεύθυνση του εν λόγω εστιατορίου που υπάρχει στις κριτικές. Δηλαδή με το που σώσουμε το άρθρο που γράφουμε για κάποιο εστιατόριο η διεύθυνση που έχουμε γράψει στο πεδίο address μετατρέπεται σε συντεταγμένες και εμφανίζεται μετά τις σημειώσεις (notes) και πριν τα σχόλια με ένα marker σε ένα χάρτη της Google. Ακολουθεί ο κώδικας και εν συνεχεία η ανάλυσή του.

```
<script
src=http://maps.google.com/maps?file=api&v=2.x&key=ABQIAAAAzr2EBOXUKnm_j
```

«Δημιουργία ιστότοπου με την χρήση cms»

```
Vnk0OJI7xSosDVG8KKPE1-m51RBrvYughuyMxQ-i1QfUnH94QxWla6N4U6MouMmBA
type="text/javascript">
```

```
</script>
```

```
<script type="text/javascript">
```

```
var map;
var address_com;
```

```
function initialize() {
 if (GBrowserIsCompatible()) {
 map = new GMap2(document.getElementById("map_canvas"));
 map.setCenter(new GLatLng(37.58, 23.43), 16);
 map.setUIToDefault();
 map.setMapType(G_SATELLITE_MAP);
 geocoder = new GClientGeocoder();
 address_com = "<?php echo $row->address; ?>";
 showAddress(address_com);
 }
}
```

```
function showAddress(address_var) {
 if (geocoder) {
 geocoder.getLatLng(address_var,function(point) {
 if (!point) {
 alert("Η " + address_var + " δεν υπάρχει στο Google Maps.");
 } else {
 map.panTo(point);
 var myHtml = address_var;
 var mypoint = point;
 map.openInfoWindow(point, myHtml);
 map.addOverlay(createMarker(point, myHtml));
 }
 });
 }
}
```

```
function createMarker(point, message) {
 var marker = new GMarker(point);

 GEvent.addListener(marker, "click", function() {
 var myHtml = "" + message + "
";
 map.openInfoWindowHtml(point, myHtml);
 });
}
```

```
«Δημιουργία ιστότοπου με την χρήση cms»
```

```
return marker;
}

</script>
<body onload="initialize()" onunload="GUnload()">
 <div id="map_canvas" style="width: 650px; height: 400px"></div>
</body>
```

Στην αρχή ανάμεσα στα script tags καλείται ο χάρτης και ο API κώδικας της Google με όλες του τις βιβλιοθήκες και δηλώνουμε ότι ο κώδικας είναι ένα javascript. Ορίζουμε δύο global μεταβλητές την map και την address\_com.

Στο πρόγραμμα έχουμε τρεις βασικές συναρτήσεις. Η initialize() αρχικά βάζει στο αντικείμενο map τον χάρτη με id όνομα map\_canvas και ακολούθως ορίζουμε το κέντρο του χάρτη σε ένα σημείο και με ένα συγκεκριμένο ζουμ. Στην δική μας περίπτωση έχουμε βάλει σαν κέντρο του χάρτη την περιοχή της Αθήνας με συντεταγμένες (37.58, 23.43) και επίπεδο ζουμ 16. Στη συνέχεια καλείται η μέθοδος setUIToDefault() που ρυθμίζει την διεπαφή του χάρτη (interface) σε default κατάσταση και με τη βοήθεια της setMapType(G\_SATELLITE\_MAP); βάζουμε σαν αρχική επιλογή για τον χάρτη μας την δορυφορική εικόνα. Ενεργοποιούμε την υπηρεσία γεωκωδικοποίησης του API της Google μέσω του αντικειμένου της κλάσης GClientGeocoder() που ονομάζεται geocoder. Η κλάση αυτή επικοινωνεί απ ευθείας με τους server της Google και αποκτά την γεωκωδικοποίηση για την συγκεκριμένη διεύθυνση που έχει δώσει ο χρήστης.

Στην μεταβλητή address\_com ορίζουμε την τιμή της php μεταβλητής address στην οποία είναι καταχωρημένη η διεύθυνση που έχουμε ορίσει και έχει αποθηκευτεί στην mySql βάση του component. Ακολούθως καλούμε την συνάρτηση showAddress() η οποία παίρνει σαν όρισμα την μεταβλητή που μας φέρνει την διεύθυνση που έχουμε βάλει στο component.

Η συνάρτηση showAddress() μέσω του αντικειμένου geocoder γεωκωδικοποιεί ( κάνει μετατροπή της διεύθυνσης σε συντεταγμένες ) και δημιουργεί ένα marker στο εν λόγω σημείο. Η συνάρτηση createMarker() δέχεται σαν όρισμα τις συντεταγμένες και μια ακολουθία χαρακτήρων (string) το οποίο στην περίπτωσή μας είναι η διεύθυνση. Η createMarker() όπως αναφέρθη δημιουργεί το marker στο σημείο όπου έχει επιστρέψει η γεωκωδικοποίηση, ενώ δημιουργεί και ένα γεγονός ( event ) πάνω στον χάρτη έτσι ώστε όταν πατήσουμε πάνω στο marker εμφανίζει μέσα σε ένα balloon την ακολουθία χαρακτήρων ( string ) με την διεύθυνση που έχει οριστεί.

Τέλος ανάμεσα στα tags <body> υπάρχει ο html κώδικας για τον χάρτη όπου με το που φορτώνει η σελίδα καλείται η initialize() και εμφανίζεται ο χάρτης με τις συγκεκριμένες διαστάσεις και id name map\_canvas. Στην παρακάτω εικόνα βλέπουμε πως εμφανίζεται ο χάρτης στην σελίδα μας.

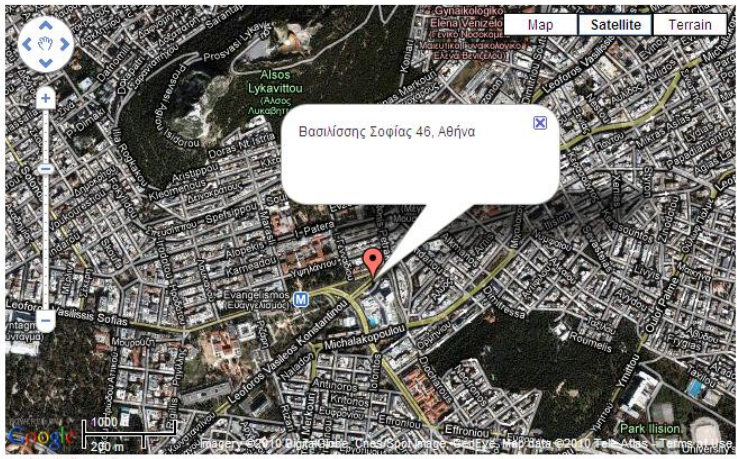
**Address:** Λεωφ. Συγγρού 115 (Ledra Marriott Hotel)

**Cuisine:** ΜΕΣΟΓΕΙΑΚΗ ΚΟΥΖΙΝΑ

Χαλαρώστε σ' ένα κομψό περιβάλλον με λεπτομέρειες από ξύλο και γυαλί, απολαύστε τα αρώματα που αναδύονται από την ανοιχτή κουζίνα κι επιλέξτε νόστιμες Ελληνικές σπεσιαλιτέ και άλλες Μεσογειακές γεύσεις.

Το εστιατόριο Βυζαντινό του Hilton Αθηνών είναι ανοιχτό όλο το χρόνο για πρωινό, μεσημεριανό και βραδινό και αποτελεί την ιδανική επιλογή για όσους αναζητούν αγές, αυθεντικές γεύσεις σε σύγχρονο περιβάλλον.

Notes:



< return to the reviews

Εικόνα: Παρουσίαση χάρτη.

## 15 Προετοιμασία και εγκατάσταση του component.

Για να μπορεί ο οποιοσδήποτε χρήστης που χρησιμοποιεί το Joomla, για να φτιάξει τις ιστοσελίδες του, να εκμεταλλευτεί το component που φτιάξαμε θα πρέπει να κάνουμε δύο πράγματα. Πρώτα θα πρέπει να δημιουργήσουμε ένα xml αρχείο στον κατάλογο που έχουμε με όνομα reviews.xml και να εισάγουμε τα παρακάτω:

```
<?xml version="1.0" encoding="utf-8"?>
<install type="component" version="1.5.0">
<name>Hotel Restaurant Reviews</name>
<author>Nick Zaxariadis</author>
<creationDate>September 2010</creationDate>
<copyright>(C) 2010</copyright>
<authorEmail>nick_zaxariadis@yahoo.com</authorEmail>
<authorUrl>http://localhost/my_project</authorUrl>
<version>1.5.0</version>
<description>A component for writing and managing
hotel restaurant reviews.</description>
<installfile>install.reviews.php</installfile>
<uninstallfile>uninstall.reviews.php</uninstallfile>
<install>
<sql>
<file driver="mysql" charset="utf8">install.mysql.sql</file>
</sql>
</install>
<uninstall>
<sql>
<file driver="mysql" charset="utf8">uninstall.mysql.sql</file>
</sql>
</uninstall>
</files>
```

«Δημιουργία ιστότοπου με την χρήση cms»

```

<filename>reviews.html.php</filename>
<filename>reviews.php</filename>
<filename>css/styles.css</filename>
</files>
<administration>
<menu>Hotel Restaurant Reviews</menu>
<submenu>
<menu link="option=com_reviews">Manage Reviews</menu>
<menu task="comments">Manage Comments</menu>
</submenu>
<files folder="admin">
<filename>install.mysql.sql</filename>
<filename>uninstall.mysql.sql</filename>
<filename>admin.reviews.html.php</filename>
<filename>admin.reviews.php</filename>
<filename>controller.php</filename>
<filename>tables/comment.php</filename>
<filename>tables/review.php</filename>
<filename>toolbar.reviews.html.php</filename>
<filename>toolbar.reviews.php</filename>
</files>
</administration>
</install>

```

Σε αυτό το xml αρχείο περιέχονται στοιχεία μέσα σε αντίστοιχα tags όπως το όνομα του module το πότε και από ποιόν δημιουργήθηκε ποια άλλα αρχεία περιλαμβάνει και τις ρυθμίσεις που προσφέρει. Στο component υπάρχει μια ιδιαιτερότητα η οποία είναι τα αρχεία που υπάρχουν και αφορούν το administration panel. Γι αυτό το λόγο ανάμεσα στα administration tags περιλαμβάνονται όλα τα αρχεία που πρόκειται να εγκατασταθούν και σχετίζονται με το φάκελο admin. Κάτω από το tag <install> υπάρχει το tag <sql> στο οποίο εσωκλείουμε το αρχείο install.mysql.sql το οποίο περιέχει sql queries για να εγκατασταθούν οι πίνακες στη βάση μας κατά την διάρκεια της εγκατάστασης. Τα αντίστοιχα αρχεία και tags υπάρχουν και για την διαδικασία της απεγκατάστασης.

Δεύτερον θα πρέπει με τη βοήθεια ενός προγράμματος συμπίεσης αρχείων να συμπίεσουμε όλα τα αρχεία που έχουμε δημιουργήσει μαζί με τα αρχεία του administration panel ώστε να προκύψει ένα αρχείο με όνομα com\_hotel\_restaurant\_reviews.zip. Έτσι θα μπορεί να γίνει η εγκατάσταση του component μέσω του administration panel του Joomla σε οποιαδήποτε ιστοσελίδα έχει δημιουργηθεί με το συγκεκριμένο cms

Το αρχείο install.mysql.sql περιέχει τον παρακάτω κώδικα για την δημιουργία των πινάκων στην Mysql βάση μας σε αυτά τα queries χρησιμοποιούμε την # η οποία αντικαθιστά το πρόθεμα της βάσης που χρησιμοποιεί το κάθε σύστημα Joomla, ώστε να αποφεύγονται τα οποιαδήποτε προβλήματα κατά τις διαδικασίες της εγκατάστασης αλλά και της απεγκατάστασης.

```

CREATE TABLE IF NOT EXISTS #__reviews (
id int(11) NOT NULL auto_increment,
restaurant_name varchar(255) NOT NULL,
address varchar(255) NOT NULL,
reservations varchar(40) NOT NULL,
quick_note text NOT NULL,
review text NOT NULL,
notes text NOT NULL,
smoking tinyint(1) NOT NULL default 0,

```

```

credit_cards varchar(255) NOT NULL,
cuisine varchar(40) NOT NULL,
avg_dinner_price tinyint(4) NOT NULL default 0,
review_date datetime NOT NULL,
published tinyint(1) NOT NULL default 0,
PRIMARY KEY (id)
);
CREATE TABLE IF NOT EXISTS #__reviews_comments(
id int(11) NOT NULL auto_increment,
review_id int(11) NOT NULL,
user_id int(11) NOT NULL,
full_name varchar(50) NOT NULL,
comment_date datetime NOT NULL,
comment_text text NOT NULL,
PRIMARY KEY (id)
);

```

Το αντίστοιχο αρχείο για την απεγκατάσταση των πινάκων με όνομα `uninstall.mysql.sql` περιέχει τον παρακάτω κώδικα.

```

DROP TABLE #__reviews;
DROP TABLE #__reviews_comments;

```

## 16 Δημιουργία Module για τις κριτικές

Σε αυτό το κεφάλαιο θα δημιουργήσουμε ένα module για τις κριτικές των εστιατορίων ώστε να μπορούν οι επισκέπτες της ιστοσελίδας μας να τις παρακολουθούν στην αρχική σελίδα. Αυτή ακριβώς είναι και η δουλεία που κάνει ένα module δηλαδή να φέρνει και να δείχνει τα δεδομένα οπουδήποτε θέλουμε στην σελίδα μας.

### 16.1 Εγγραφή του module στη βάση μας.

Όπως είχαμε κάνει και για το component έτσι και για το module θα πρέπει να το εγγράψουμε στη βάση δεδομένων ώστε να μπορούμε να το χειριζόμαστε από τον πίνακα ελέγχου (administration panel). Έτσι δημιουργούμε στον πίνακα `jos_module` μια καταγραφή είτε γράφοντας τον παρακάτω sql κώδικα είτε χρησιμοποιώντας το interface του phpMyAdmin.

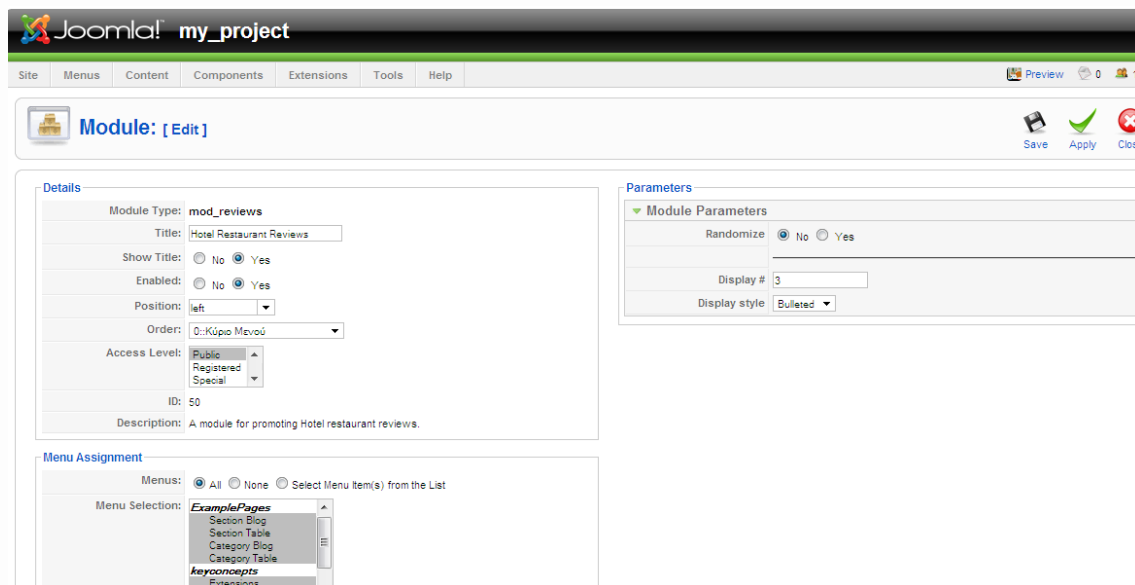
```

INSERT INTO jos_modules (title, ordering, position,
 published, module, showtitle, params)
VALUES (Restaurant Reviews, 1, left, 1,
 mod_reviews, 1, style=simple\nitems=3\nrandom=1);

```

Εάν γίνουν όλα σωστά τότε πηγαίνοντας στο administration panel πατάμε `extensions/module manager` και βρίσκουμε το link με το όνομα `hotel restaurant reviews` κλικάροντας μας μεταφέρει στην σελίδα(της παρακάτω φωτογραφίας) που μπορούμε να κάνουμε ρυθμίσεις για το module όπως το που και πώς θα εμφανίζεται. Συγκεκριμένα στη δική μας υπόθεση θα εμφανίζεται στην αριστερή μεριά της ιστοσελίδας και σε όλες τις υποσελίδες.





Εικόνα: Ρυθμίσεις module.

## 16.2 Κώδικας για το module.

Πρώτα θα γράψουμε τον κώδικα μιας κλάσης η οποία μας επιτρέπει να παρουσιάζουμε τα δεδομένα από τις κριτικές με είτε με σειρά δημοσίευσης είτε με τυχαία σειρά. Σε αυτό η δύο βασικές συναρτήσεις της κλάσης. Μέσα στο κατάλογο modules δημιουργούμε έναν καινούριο φάκελο με όνομα mod\_reviews και ένα αρχείο με όνομα helper.php και γράφουμε τα παρακάτω.

```
<?php
defined('_JEXEC') or die('Restricted access');
class modReviewsHelper
{
 function getReviews(&$params)
 {
 $items = $params->get('items', 1);
 $db =& JFactory::getDBO();
 $query = "SELECT id, restaurant_name, quick_note FROM #__reviews
 WHERE published = '1' ORDER BY review_date DESC";
 $db->setQuery($query, 0, $items);
 $rows = $db->loadObjectList();
 return $rows;
 }
 function renderReview(&$review, &$params)
 {
 $link = JRoute::_
 ("index.php?option=com_reviews&task=view&id=" .
 $review->id);
 require(JModuleHelper::getLayoutPath('mod_reviews', '_review'));
 }
}
```

```

function getRandomReview()
{
 $db =& JFactory::getDBO();
 $query = "SELECT id, restaurant_name, quick_note FROM #__reviews";
 $db->setQuery($query);
 $rows = $db->loadObjectList();
 $i = rand(0, count($rows) - 1);
 $row = array($rows[$i]);
 return $row;
}
}
?>

```

Η κλάση ονομάζεται `modReviewsHelper` όταν φορτώνεται το `module` στην συνάρτηση `getReviews` το αντικείμενο `$params` χρησιμοποιείται για να πάρει και να ρυθμίσει τις παραμέτρους. Στον πίνακα `jos_modules` η στήλη με τις παραμέτρους περιέχει τρεις τιμές μία για τον αριθμό των κριτικών που θα περιέχει το `module` και είναι ρυθμισμένο στο 3 μια για το `style` που θα παρουσιάζονται με ρύθμιση στο απλό (`Simple`) και μια για το `random` με ρύθμιση 1. Αυτά με τη βοήθεια της συνάρτησης `get()` μεταβιβάζονται στην μεταβλητή `$items`. Στη συνέχεια αφού πάρουμε μια αναφορά αντικειμένου από τη βάση μας γράφουμε ένα `query` και επιλέγουμε το `id`, το `restaurant_name` και το `quick_note` από τον πίνακα `jos_reviews` με αντίστροφη χρονολογική σειρά από την σύμφωνα με την ημέρα δημοσίευσής τους. Στην `setQuery()` χρησιμοποιούμε την δεύτερη και τρίτη παράμετρο για να δημιουργηθεί ένα όριο σε αυτά που φορτώνονται στο `query`. Εφόσον το `query` είναι έτοιμο φορτώνονται οι κατάλληλες γραμμές του πίνακα.

Με την `renderReview()` εμφανίζουμε το αποτέλεσμα των κριτικών και με την `JRoute` το μετατρέπουμε σε μορφή συνδέσμου (`link`) με αποτέλεσμα να γίνονται και πιο φιλικά ως προς τις μηχανές αναζήτησης. Τέλος χρησιμοποιούμε τις συναρτήσεις `require()` και `getLayoutPath()` που είναι μέλη της κλάσης `JModuleHelper` για να συμπεριλάβουμε το `template _review` που θα δημιουργήσουμε αργότερα.

Η συνάρτηση `getRandomReview()` αρχικά συνδέεται με την συγκεκριμένη βάση. Το `query` φορτώνει τις στήλες που περιλαμβάνουν το `id` το `restaurant_name` και το `quick_note` από τις γραμμές του πίνακα `jos_reviews` στην μεταβλητή `$row` αφού γίνει αυτό η PHP συνάρτηση `rand()` χρησιμοποιείται για να πάρει μια τυχαία τιμή μεταξύ του 0 και του αριθμού των σειρών μείον ένα. Κατόπιν η μεταβλητή `$row` τοποθετείται σε ένα πίνακα που περιέχει ένα στοιχείο με το δείκτη του αντικειμένου που επιλέχθηκε τυχαία.

Για να λειτουργήσει η κλάση που δημιουργήσαμε θα πρέπει να φτιάξουμε και άλλο ένα αρχείο στο φάκελο `mod_reviews` με όνομα `mod_reviews.php` και να συμπεριλάβουμε τον παρακάτω κώδικα.

```

<?php
defined('_JEXEC') or die('Restricted access');
require(dirname(__FILE__).DS.'helper.php');
$random = $params->get('random', 0);
$style = $params->get('style', 'default');
if($random)
{
 $list = modReviewsHelper::getRandomReview();
}

```

```

else
{
$list = modReviewsHelper::getReviews($params);
}
require(JModuleHelper::getLayoutPath('mod_reviews', $style));
?>

```

Η συνάρτηση require() ενσωματώνει το αρχείο helper.php στο οποίο βρίσκετε η βοηθητική κλάση που γράψαμε παραπάνω. Στην μεταβλητή \$random δίνεται η τιμή του αντικειμένου \$params στο οποίο έχουν φορτωθεί οι παράμετροι του module μέσω τις get(). Στη συνέχεια ελέγχουμε την τιμή της μεταβλητής \$random και αν έχει μη μηδενική τιμή καλείται η συνάρτηση getRandomReview() αλλιώς καλείται η getReviews(). Με αυτόν τον τρόπο δίνεται η δυνατότητα δύο διαφορετικών τρόπων παρουσίασης των κριτικών μας. Με την βοήθεια της getLayoutPath('mod\_reviews', \$style)); διαλέγουμε το πώς θα εμφανιστούν οι κριτικές είτε δηλαδή με τελείες (bullets) είτε όχι. Για να γίνει αυτό πρέπει να δημιουργήσουμε στον κατάλογο modules/mod\_reviews έναν φάκελο tmpl ο οποίος θα περιέχει 3 αρχεία το \_review.php το default.php και το bulleted.php.

Στο πρώτο αρχείο εισάγουμε τον παρακάτω κώδικα ο οποίος μετατρέπει σε link το όνομα του εστιατορίου και το quick note. Όπως παρατηρούμε το όνομα του αρχείου αρχίζει με underscore ώστε να προσφέρετε μόνο για εσωτερική χρήση και όχι σαν επιλογή στον admin.

```

<?php defined('_JEXEC') or die('Restricted access'); ?>
<a href="<?php echo $link ?>"><?php echo $review->restaurant_name; ?>

<p><?php echo $review->quick_note?></p>


```

Στο δεύτερο αρχείο εισάγουμε τον παρακάτω κώδικα με τον οποίο δημιουργούμε ένα βρόχο στη λίστα με τις κριτικές και στη συνέχεια καλείται η βοηθητική κλάση που ετοιμάζει να εμφανίσει τις κριτικές.

```

<?php
defined('_JEXEC') or die('Restricted access');
foreach ($list as $review){
modReviewsHelper::renderReview($review, $params);
}
?>

```

Κατά τον ίδιο τρόπο με παραπάνω δημιουργούμε και το τρίτο αρχείο με τον εξής κώδικα. Η μόνη διαφορά είναι ότι το αποτέλεσμα περικλείεται από τελείες. Και στις δύο περιπτώσεις μέσω της κλάσης φορτώνεται το \_review layout.

```

<?php defined('_JEXEC') or die('Restricted access'); ?>

<?php
foreach ($list as $review)
{
echo "";
modReviewsHelper::renderReview($review, $params);
echo "";
}
?>


```

Αφού έχουμε τελειώσει με το γράψιμο του κώδικα το αποτέλεσμα φαίνεται στην παρακάτω εικόνα. Στην περίπτωση μας το module θα εμφανίζεται στη δεξιά μεριά της ιστοσελίδας ακριβώς κάτω από τις δημοσκοπήσεις.

The screenshot shows a Joomla! website interface. On the left, there's a sidebar with a 'Hotel Restaurant Reviews' module. The main content area displays several hotel reviews, including 'The BYZANTINO Restaurant Hilton' and 'ZEPHYROS Ledra Marriott'. A red arrow points to the 'ZEPHYROS Ledra Marriott' review. On the right, there's a 'TA NEA RSS FEED' module displaying a list of news items. The background features a large, faint watermark of the Greek Parliament building.

Εικόνα: Εμφάνιση module.

### 16.3 Προετοιμασία και εγκατάσταση του module.

Για να μπορεί ο οποιοσδήποτε χρήστης που χρησιμοποιεί το Joomla, για να φτιάξει τις ιστοσελίδες του, να εκμεταλλευτεί το module που φτιάξαμε θα πρέπει να κάνουμε δύο πράγματα. Πρώτα θα πρέπει να δημιουργήσουμε ένα xml αρχείο στον κατάλογο που έχουμε με όνομα mod\_review.xml και να εισάγουμε τα παρακάτω:

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5">
<name>Hotel Restaurant Reviews</name>
<author>Nick Zaxariadis</author>
<creationDate>September 2010</creationDate>
<copyright>(C) 2010</copyright>
<authorEmail>nick_zaxariadis@yahoo.com</authorEmail>
<authorUrl>http://localhost/my_project</authorUrl>
<version>1.0</version>
<description>A module for promoting Hotel restaurant
reviews.</description>
<files>
<filename module="mod_reviews">mod_reviews.php</filename>
<filename>helper.php</filename>
<filename>tmpl/_review.php</filename>
<filename>tmpl/bulleted.php</filename>
<filename>tmpl/default.php</filename>
</files>
<params>
<param name="random" type="radio" default="0" label="Randomize"
description="Show random reviews">
<option value="0">No</option>
<option value="1">Yes</option>
«Δημιουργία ιστότοπου με την χρήση cms»
```

```
</param>
<param name="@spacer" type="spacer"
default="" label="" description="" />
<param name="items" type="text" default="1" label="Display #"
description="Number of reviews to display" />
<param name="style" type="list" default="default" label="Display
style" description="The style to use for displaying the reviews.">
<option value="default">Flat</option>
<option value="bulleted">Bulleted</option>
</param>
</params>
</install>
```

Σε αυτό το xml αρχείο περιέχονται στοιχεία μέσα σε αντίστοιχα tags όπως το όνομα του module το πότε και από ποιόν δημιουργήθηκε ποια άλλα αρχεία περιλαμβάνει και τις ρυθμίσεις που προσφέρει. Δεύτερον θα πρέπει με τη βοήθεια ενός προγράμματος συμπίεσης αρχείων να συμπίεσουμε όλα τα αρχεία που έχουμε δημιουργήσει ώστε να προκύψει ένα αρχείο με όνομα mod\_reviews.zip. Έτσι θα μπορεί να γίνει η εγκατάσταση του module μέσω του administration panel του Joomla σε οποιαδήποτε ιστοσελίδα έχει δημιουργηθεί με το συγκεκριμένο cms

## 17 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.

Για να μπορέσει η συγκεκριμένη εφαρμογή που αφορά εύρεση ξενοδοχείων στην Ελλάδα να γίνει ακόμα πιο ελκυστική για τον χρήστη στο απώτερο μέλλον θα μπορέσουν να προστεθούν ορισμένες επεκτάσεις.

Οι πιο σημαντικές από αυτές θα μπορούσαν να ήταν οι εξής:

- Μεγαλύτερο εύρος ξενοδοχείων και δωματίων.
- Δυνατότητα online παρακολούθησης διαθεσιμότητας δωματίων.
- Κλείσιμο δωματίων μέσω πιστωτικών καρτών.( Απαραίτητη προϋπόθεση η ύπαρξη συστημάτων ασφαλείας για τις συναλλαγές.)
- Να προτείνει το σύστημα ανάλογα ξενοδοχεία η δωμάτια σύμφωνα με τις προτιμήσεις του κάθε χρήστη.
- Δυνατότητα απ ευθείας ενοικίασης αυτοκινήτων η σκαφών.

## 18 Συμπεράσματα

Η συγκεκριμένη διπλωματική εργασία αναφέρεται στα συστήματα διαχείρισης περιεχομένου (CMS- Content Management Systems) τα οποία παρέχουν στους διαχειριστές ένα ολοκληρωμένο και σωστά δομημένο περιβάλλον επεξεργασίας περιεχομένων. Αυτό έχει σαν αποτέλεσμα να μπορούμε να δημιουργήσουμε ιστοσελίδες υψηλού επιπέδου. Παρέχουν μεγάλο εύρος επεκτάσεων που μπορούμε να αγοράσουμε είτε να βρούμε δωρεάν στο διαδίκτυο. Επιπλέον οι κοινότητες υποστήριξης των CMS φροντίζουν ώστε να υπάρχει η λύση σε οποιοδήποτε πρόβλημα δημιουργηθεί.

Πιο συγκεκριμένα φτιάξαμε μια ιστοσελίδα για αναζήτηση ξενοδοχείων έχοντας δημιουργήσει νέες επεκτάσεις του cms Joomla και παραμετροποιήσει είδη υπάρχουσες λειτουργίες. Με αυτό τον τρόπο δώσαμε την δυνατότητα στους χρήστες της ιστοσελίδας να ανανεώνουν εύκολα και γρήγορα το περιεχόμενο, να αποφασίζουν για το πώς θα παρουσιάζεται (design) και με τις επιπλέον εφαρμογές να απαιτούν το καλύτερο δυνατό αποτέλεσμα.

## 19 Βιβλιογραφία & Πηγές στο Διαδίκτυο

- [1] Davenport, T., Prusak, L., Working Knowledge: How Organizations Manage What They Know, - Harvard Business School Press (1998).
- [2] Verna A., The Future of Knowledge: Increasing Prosperity through ValueNetworks, USA:Butterworth-Heinemann (2003).
- [3] Beginning Joomla From Novice to Professional - Dan Rahmel
- [4] Learning Joomla! 1.5 Extension Development- Joseph LeBlanc- Packt Publishing
- [5] Mastering Joomla! 1.5 Extension and Framework Development- James Kennard
- [6] Professional Joomla! (Programmer to Programmer), Dan Rahmel, John Wiley & Sons
- [7] Call D., Knowledge Mangement – Not Rocket Science, Journal of Knowledge Management, Vol. 9, Issue 2, p. 19-30.
- [8] <http://www.joomla.org>
- [9] <http://www.joomla.gr>
- [10] <http://en.wikipedia.org/wiki/Joomla!>
- [11] <http://extensions.joomla.org>
- [12] <http://help.joomla.org>
- [13] <http://forum.joomla.gr>
- [14] <http://www.w3schools.com/>
- [15] <http://code.google.com/apis/maps/documentation/geocoding/>
- [16] <http://www.rss-specifications.com/locate-jobs-with-rss.htm>
- [17] <http://maps.google.com/>
- [18] <http://download.oracle.com/javase/tutorial/index.html>
- [19] <http://www.tanea.gr/>
- [20] <http://www.java2s.com/Tutorial/Java/CatalogJava.htm>
- [21] <http://rss.softwaregarden.com/aboutrss.html>
- [22] [http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)
- [23] <http://www.turismo.eu/it/geo.html>
- [24] [http://www.docs.fitiitnt.org/joomla16alpha2/Joomla-Administrator/mod\\_menu/JMenuNode.html](http://www.docs.fitiitnt.org/joomla16alpha2/Joomla-Administrator/mod_menu/JMenuNode.html)
- [25] <http://trends.builtwith.com/cms>