



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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Συγγραφή οδηγού κατασκευής ενός Joomla
Component και ανάπτυξη εφαρμογής για την
ενοικίαση δωματίων βασισμένη σε αυτό.**



Της φοιτήτριας

Χατζηαναστασίου Μαγδαληνής

Αρ. Μητρώου: 04/2541

Επιβλέπων καθηγητής:

Δρ .Κλεφτούρης Δημήτριος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ	4
ΚΕΦΑΛΑΙΟ 1	6
1.1 Σύστημα Διαχείρισης Περιεχομένου	6
1.2 Τρόπος λειτουργίας.....	6
1.3 Στόχοι.....	6
1.4 Δυο σημαντικές ιδιότητες	7
1.5 Πρόσθετα χαρακτηριστικά.....	8
1.6 Πλεονεκτήματα ενός CMS.....	9
1.7 Μειονεκτήματα ενός CMS.....	11
1.8 Σύγκριση Συστημάτων Διαχείρισης Περιεχομένου	14
1.8.1 Το κοινό χαρακτηριστικό.....	15
1.8.2 Σύγκριση των δημοφιλέστερων CMSs	16
1.9 Ελεύθερο λογισμικό	17
1.9.1 Ορισμός Ελεύθερου λογισμικού	18
1.9.2 Άδειες ελεύθερου λογισμικού	18
1.9.3 Πλεονεκτήματα χρήσης.....	18
1.9.4 Μειονεκτήματα χρήσης.....	19
1.10 Επιλογή του CMS Joomla!	19
1.10.1 Τρόπος λειτουργίας.....	19
1.10.2 Πού χρησιμοποιείται.....	20
1.11 Επέκταση του Joomla!.....	20
1.11.1 Ορισμός	20
1.11.2 Οι λόγοι επέκτασης της Joomla!.....	21
1.11.3 Προσαρμογή εναντίον της επέκτασης	22
1.11.4 Τρόποι επέκτασης του Joomla!	23
ΚΕΦΑΛΑΙΟ 2	27
2.1 Ορισμοί.....	27
2.2 Εγκατάσταση	28
2.2.1 Προετοιμασία εγκατάστασης.....	28
2.2.2 Εγκατάσταση μέσω του περιηγητή	31
2.3 Δομή ενός Component	39
2.5 Εκτελώντας το Component	39
2.6 Διαχωρισμός μεταξύ frond-end και back-end	41

2.7 Καταχώρηση του component στη βάση δεδομένων (Registering).....	42
2.8 Δημιουργώντας Toolbars	45
2.9 Κουμπιά εργαλειοθήκης	47
2.10 Ανάπτυξη του Back-end.....	48
2.10.1 Δημιουργώντας μια κλάση Table	50
2.10.2 Δημιουργώντας τη φόρμα Review	51
2.10.3 Επεξεργασία δεδομένων	57
2.10.4 Δημιουργία λίστας με τα δεδομένα της βάσης	58
2.11 Ανάπτυξη του Front-end	62
2.11.1 Συγκέντρωση των reviews.....	62
2.11.2 Εμφάνιση των reviews.....	64
ΑΝΑΦΟΡΕΣ	68

ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία έχει ως σκοπό να συστήσει στους χρήστες τα Συστήματα Διαχείρισης Πληροφοριών, παρουσιάζοντας τα πλεονεκτήματά τους και τους τρόπους που μπορούν να χρησιμοποιηθούν ώστε να συμβάλουν στην προβολή επιχειρήσεων διεθνώς μέσω του διαδικτύου. Ταυτόχρονα, τα συστήματα αυτά διευκολύνουν τον προγραμματιστή και κάθε ανειδίκευτο χρήστη κατά την ανάπτυξη ενός ιστοχώρου, χάρη στην τυποποιημένη μορφή τους και τα ήδη υπάρχοντα στην αγορά, δωρεάν κομμάτια κώδικα για συγκεκριμένη χρήση, τα λεγόμενα συστατικά ή components, ανάλογα κάθε φορά με την απαίτηση της εφαρμογής του. Στη συνέχεια επιλέγεται για την εργασία μας το Σύστημα Διαχείρισης Πληροφοριών Joomla, καθώς ενδείκνυται για κατασκευή ιστοσελίδων που αφορούν προβολή, είναι εύχρηστο και παρέχει ικανοποιητική υποστήριξη στον προγραμματιστή και στο χρήστη, και τέλος έχει ωραία γραφικά περιβάλλοντα και κατανοητά μενού.

Η εργασία αποτελείται ουσιαστικά από δύο τμήματα, πέραν του θεωρητικού υπόβαθρου. Το πρώτο είναι ο οδηγός κατασκευής ενός component, ενώ στο δεύτερο αναπτύσσεται και περιγράφεται μια ενδεικτική εφαρμογή με βάση τον οδηγό κατασκευής.

Στο πρώτο τμήμα, περιγράφεται βήμα προς βήμα η κατασκευή και η ονομασία των αρχείων που θα δημιουργήσει ο χρήστης-προγραμματιστής, περιλαμβάνονται τα κομμάτια κώδικα που χρησιμοποιούνται σε κάθε περίπτωση, αφού μια τέτοια κατασκευή είναι πλέον τυποποιημένη, και γίνεται ξεκάθαρα ο διαχωρισμός των δύο μονοπατιών που αποτελούν τη διαδικασία ανάπτυξης ενός ιστοχώρου, αυτά του προγραμματισμού και του τρόπου εμφάνισης. Σκοπός αυτού του μέρους είναι να μπορέσει κάποιος προγραμματιστής, χωρίς απαραίτητα να έχει δημιουργήσει στο παρελθόν κάτι αντίστοιχο, να φτιάξει από το μηδέν ένα απλό

component, με τη χρήση του κώδικα, των οδηγιών, και των εντολών που περιγράφονται.

Το δεύτερο τμήμα αφορά την εφαρμογή που κατασκευάστηκε με βάση τον οδηγό. Δημιουργήθηκε ένα component, που πραγματεύεται την διαδικασία κράτησης δωματίων σε μια τουριστική επιχείρηση. Η εφαρμογή απεικονίζει στον επισκέπτη της σελίδας, μια φόρμα με επιλογές όπως η πιθανή ημερομηνία κράτησης και ο τύπος δωματίου, και σύμφωνα με τις εκάστοτε επιλογές το σύστημα πραγματοποιεί τη διαδικασία και επιστρέφει ενημέρωση με γραπτά μηνύματα.

Οι λόγοι που ωθούν στη δημιουργία μιας τέτοιας εφαρμογής είναι αρκετοί. Αρχικά η τεχνολογία έχει γίνει αναπόσπαστο κομμάτι της καθημερινότητάς μας και την επιζητούμε ώστε να κάνει την ζωή μας ευκολότερη. Επιπλέον, το διαδίκτυο χρησιμοποιείται για την παραμικρή διευκόλυνση ενός πολύ μεγάλου μέρους πολιτών αλλά και επιχειρήσεων και κατ' επέκταση η αγορά και χρήση των ψηφιακών τεχνολογιών και υπηρεσιών οδηγείται από το αυτό. Πολλές επιχειρήσεις θέλουν να αυξήσουν την προβολή τους ή να επεκταθούν διεθνώς και τη λύση έρχεται να δώσει το ίντερνετ. Για παράδειγμα, μια τουριστική μονάδα όπως είναι ένα ξενοδοχείο θα μπορούσε να έχει μεγάλο όφελος από ένα τέτοιο ηλεκτρονικό εγχείρημα. Αρχικά, απλά και μόνο με την προβολή της μέσω του διαδικτύου, αυτόματα συστήνεται σε περισσότερο κόσμο και σε παγκόσμια κλίμακα, γίνεται αναγνωρίσιμη, προβάλλοντας στοιχεία όπως την τοποθεσία της, τους εσωτερικούς χώρους, το περιβάλλον της και τα πλεονεκτήματά της. Στη συνέχεια, κάτι που θα ολοκλήρωνε μια τέτοια προσπάθεια της επιχείρησης για προβολή, θα ήταν να δημιουργήσει ένα ηλεκτρονικό σύστημα για κρατήσεις δωματίων, που θα μπορούσε να είναι ευέλικτο, ανάλογα με τις ανάγκες του ξενοδοχείου. Για να κάνει κάτι τέτοιο, οι τρόποι είναι είτε να επενδύσει στο να της φτιάξει ένας προγραμματιστής έναν ιστοχώρο, είτε να διαφημίζεται μέσω άλλων εμπορικών εταιριών που κάνουν κρατήσεις και προσφορές και να πληρώνει ένα ποσοστό, είτε να φτιάξει με ένα Σύστημα Διαχείρισης Περιεχομένου μια ιστοσελίδα και στη συνέχεια να προσθέσει μία έτοιμη επέκταση για ηλεκτρονικές κρατήσεις. Και στους τρεις παραπάνω τρόπους, είναι πολύ πιθανό να χρησιμοποιηθεί ένα τέτοιο έτοιμο extension, το οποίο υπάρχει στο διαδίκτυο και μπορεί να είναι commercial ή non-commercial. Ακριβώς για το λόγο αυτό, επειδή οι ανάγκες των ξενοδοχείων είναι τέτοιες που κάνουν επιτακτική πλέον την χρήση της ηλεκτρονικής κράτησης (e-booking), στην εργασία αυτή, επιλέξαμε να δημιουργήσουμε μία τέτοια κατασκευή.

Η υλοποίηση της εφαρμογής έγινε σε γλώσσα προγραμματισμού PHP, ο web server που χρησιμοποιήθηκε είναι ο Uniform Server, ενώ η βάση δεδομένων είναι η MySQL και τέλος, είναι συμβατή για χρήση σε όλες τις εκδόσεις 1.5 της πλατφόρμας του Joomla.

ΚΕΦΑΛΑΙΟ 1

1.1 Σύστημα Διαχείρισης Περιεχομένου

Το Σύστημα Διαχείρισης Περιεχομένου (Content Management System, CMS), είναι ένα πρόγραμμα ειδικά σχεδιασμένο για τη διαχείριση ιστότοπων. Δημιουργείται από τους σχεδιαστές ιστοσελίδων, αλλά προορίζεται για χρήση από τελικούς χρήστες. Αρχικά, προσφέρει έναν εύκολο και εύχρηστο τρόπο ενημέρωσης περιεχομένου. Αυτό συνήθως γίνεται με τη χρήση ενός συστήματος πλοήγησης (browser). Ο χρήστης απλά εισάγει το νέο κείμενο και το αποθηκεύει. Η ιστοσελίδα ενημερώνεται αμέσως. Το ίδιο απλό είναι να προστεθούν νέες σελίδες, να διαγραφούν παλιές, ή να αναδιαμορφωθεί μια ιστοσελίδα ώστε να συμβαδίζει με νέες απαιτήσεις ή προδιαγραφές.

1.2 Τρόπος λειτουργίας

Ένα Σύστημα Διαχείρισης Περιεχομένου, λειτουργεί αποθηκεύοντας κείμενο και εικόνες μέσα σε μία βάση δεδομένων. Κάθε φορά που ζητείται μια ιστοσελίδα, το CMS εισχωρεί στα δεδομένα της βάσης και εμφανίζει την ιστοσελίδα. Επειδή τα δεδομένα είναι ξεχωριστά από τον κώδικα, αλλαγές στα δεδομένα μπορούν να γίνουν χρησιμοποιώντας απλά τη διεπιφάνεια χωρίς να απαιτούνται από το χρήστη γνώσεις html και ftp.

1.3 Στόχοι

Ένα Σύστημα Διαχείρισης Περιεχομένου είναι μια συλλογή από διαδικασίες που χρησιμοποιούνται προκειμένου να διαχειρίζονται τη ροή των εργασιών σε ένα συνεργατικό περιβάλλον. Αυτές οι διαδικασίες μπορούν να είναι χειροκίνητες ή βασισμένες στον υπολογιστή και σχεδιάστηκαν για να πραγματοποιούν τα παρακάτω:

- Να αφήνουν ένα μεγάλο αριθμό ανθρώπων να μοιράζονται αποθηκευμένα δεδομένα
- Να ελέγχουν την είσοδο στα δεδομένα, βασιζόμενες στο ρόλο των χρηστών και τα δικαιώματά τους (καθορίζοντας ποιες πληροφορίες ο χρήστης ή μια ομάδα χρηστών μπορεί να δει, να επέμβει σε αυτή, να δημοσιοποιήσει κτλ)
- Σκοπός τους είναι η εύκολη αποθήκευση και ανάκτηση δεδομένων
- Να μειώνουν επαναλαμβανόμενες και διπλότυπες πληροφορίες που μπορεί να έχουν εισαχθεί

- Να διατηρείται η ιστοσελίδα καλά οργανωμένη ώστε να γίνεται κατανοητό το περιεχόμενο που έχει να προσφέρει
- Να αναπτύξουν την επικοινωνία μεταξύ των χρηστών

1.4 Δυο σημαντικές ιδιότητες

Σε ένα CMS τα δεδομένα μπορούν να οριστούν σχεδόν με οποιονδήποτε τύπο: έγγραφα, ταινίες, εικόνες, τηλεφωνικοί αριθμοί, επιστημονικού υλικού δεδομένα κα. Τα CMS συχνά χρησιμοποιούνται για να αποθηκεύσουν, ελέγξουν, αναθεωρήσουν και διορθώσουν κάποιο παλιό περιεχόμενο, να εμπλουτίσουν σημασιολογικά, και να δημοσιεύσουν τεκμηριώσεις και έγγραφα. Λειτουργώντας σαν μια κεντρική αποθήκη σημαντικών και πλούσιων γνώσεων, το CMS με το σημαντικό έλεγχο Version control που διαθέτει έχει τη δυνατότητα, είτε να αφήσει το περιεχόμενο ενός ήδη υπάρχοντος αρχείου να ενημερωθεί σε μια καινούρια έκδοση, είτε να επιστρέψει σε μία παλαιότερη έκδοση. Επίσης μπορεί να εντοπίσει οποιαδήποτε αλλαγή έχει γίνει σε αρχεία από άτομα. Το Version control είναι ένα από τα κυρίαρχα πλεονεκτήματα ενός CMS. Μια ακόμη δυνατότητα που προσφέρει ένα CMS είναι το indexing-αποδελτίωση, κατά το οποίο όλα τα δεδομένα περνούν σε ένα κατάλογο (πχ σε έναν οργανισμό), ώστε οι χρήστες να ψάχνουν τα δεδομένα που τους ενδιαφέρουν χρησιμοποιώντας λέξεις κλειδιά τις οποίες το CMS γνωρίζει να ανακτά.

Ένα διαδικτυακό CMS (web CMS) είναι ένα CMS που σχεδιάστηκε για να απλοποιήσει την δημοσίευση διαδικτυακού υλικού σε ιστοσελίδες και φορητές συσκευές, μερικώς με τρόπο ώστε να δίνει τη δυνατότητα σε δημιουργούς να δημιουργούν, επιβεβαιώνουν και διαχειρίζονται περιεχόμενα, χωρίς να απαιτείται να έχουν οι ίδιοι τις τεχνικές γνώσεις καμιάς διαδικτυακής γλώσσας προγραμματισμού ή Hypertext Markup languages όπως είναι η HTML και χωρίς να ασχοληθούν με το uploading των αρχείων. Πολλά web-based CMS κατατάσσονται στα Ανοιχτά Λογισμικά ή υπάρχουν ως εμπορικά προϊόντα σε αντίστοιχες ιστοσελίδες.

1.5 Πρόσθετα χαρακτηριστικά

Εκτός από τα παραπάνω βασικά χαρακτηριστικά και ιδιότητες, το σύστημα επίσης θα ήταν καλό:

- να χρησιμοποιεί κάποιου είδους μηχανισμό ασφάλειας για να σταματά τους αναρμόδιους χρήστες από το να χειρίζονται και να τροποποιούν το περιεχόμενο
- να παρέχει χρονοδιάγραμμα για να ελέγχει πότε το περιεχόμενο εμφανίζεται
- να χρησιμοποιεί templates για τη δημοσιοποίηση του περιεχομένου της σελίδας σε ένα τυποποιημένο φορμάτ, ώστε να διατηρείται ένα συγκεκριμένο προφίλ και αίσθηση σε όλο τον ιστότοπο.
- να προσφέρει τη δυνατότητα της εξατομίκευσης στο χρήστη, ώστε να φέρνει στα μέτρα του το περιεχόμενο και συγκεκριμένα τον τρόπο που αυτό εμφανίζεται, ώστε να προσαρμόζεται στις απαιτήσεις και ανάγκες της εργασίας του.

Επίσης, ένα CMS μπορεί να παρέχει εργαλεία για εξατομικευμένο μαρκετινγκ. Αυτή η έννοια βασίζεται στην ικανότητα μιας ιστοσελίδας να διαφημίζει με βάση συγκεκριμένα χαρακτηριστικά του χρήστη, χρησιμοποιώντας πληροφορίες που παρέχονται από τον χρήστη ή συγκεντρώνονται από την ίδια την ιστοσελίδα και αποκαλύπτουν τις συνήθειες του χρήστη. Για παράδειγμα, όταν κάποιος επισκέπτης επισκεφθεί μια μηχανή αναζήτησης και ψάξει για ένα συγκεκριμένο αντικείμενο, τότε τα διαφημιστικά πλαίσια που βρίσκονται στη σελίδα θα εμφανίσουν επιχειρήσεις που πουλάνε αυτό το αντικείμενο και άλλα παρεμφερή προϊόντα (πχ αυτοκίνητα).

1.6 Πλεονεκτήματα ενός CMS

1. Διαχωρισμός σχεδίασης, δομή και περιεχομένου(design, structure, content)

Ο διαχωρισμός αυτών των τριών επιπέδων παρέχει πολλά πλεονεκτήματα καθ' όλη την εξέλιξη μιας ιστοσελίδας (life cycle). Κάθε περιοχή της σελίδας μπορεί να επαναδημιουργηθεί και να προσαρμοστεί ανεξάρτητα από τις υπόλοιπες περιοχές. Το σχεδιαστικό επίπεδο (design layer) μπορεί να είναι εντελώς αναθεωρημένο και ξαναδουλεμένο, με καινούρια εμφάνιση και χρώματα για μια νέα διεπιφάνεια χρήστη χωρίς την ανάγκη για καμία προσαρμογή στο επίπεδο της δομής (structure) ή του περιεχομένου (content). Η δομή μπορεί να ρυθμιστεί για επιπλέον λειτουργικότητα χωρίς και εδώ όπως και πριν να απαιτούνται αλλαγές στη σχεδίαση ή στο περιεχόμενο. Το περιεχόμενο μπορεί να αλλαχθεί δίχως την ανάγκη να προσαρμοστεί η front-end σχεδίαση ή η λειτουργική δομή. Ο διαχωρισμός των τριών αυτών επιπέδων δημιουργεί ένα δυνατό χαρακτηριστικό του CMS, την ευελιξία.

2. Παραγωγή περιεχομένου χωρίς προγραμματιστική ή σχεδιαστική εμπειρία

Επιτρέπει στον συντάκτη να αλληλεπιδρά με την ιστοσελίδα με διάφορους τρόπους. Παρέχει ένα γραφικό περιβάλλον χρήστη που επιτρέπει στον συντάκτη να δημιουργήσει περιεχόμενο, να προσθέσει κείμενο, εικόνες, multimedia αρχεία και πολλά ακόμη.

3. Μερικά ακόμη πλεονεκτήματα είναι:

- Συντήρηση
Βασίζεται σε ένα τυπικό φυλλομετρητή (web browser), έτσι το να μπορεί ο χρήστης να συντάξει, να διορθώσει και γενικά να έχει πρόσβαση από οπουδήποτε και οποτεδήποτε, παρέχει ευελιξία και μειώνει κάθε είδους καθυστερήσεις και συμφορήσεις.

- Διαμορφωμένα δικαιώματα και περιορισμοί πρόσβασης
Οι χρήστες έχουν προσδιορισμένους ρόλους και δικαιώματα που τους εμποδίζουν να αλλάξουν περιεχόμενα στα οποία δεν έχουν εξουσιοδότηση να το κάνουν.
- Διαφυλάσσεται η συνοχή της σχεδίασης
Επειδή το περιεχόμενο αποθηκεύεται ξεχωριστά από τη σχεδίαση, το περιεχόμενο όλων των συντακτών παρουσιάζεται με τον ίδιο τρόπο σχεδίασης.
- Η πλοήγηση γεννιέται και προσαρμόζεται αυτόματα
Τα μενού τυπικά δημιουργούνται αυτόματα με βάση το περιεχόμενο της βάσης δεδομένων και οι σύνδεσμοι (links) που τυχόν υπάρχουν δεν θα δείχνουν σε ιστοσελίδες που δεν υπάρχουν.
- Το περιεχόμενο είναι αποθηκευμένο σε μια βάση δεδομένων
Κεντρική αποθήκευση σημαίνει ότι το περιεχόμενο έχει τη δυνατότητα να επαναχρησιμοποιηθεί σε πολλά μέρη της ιστοσελίδας και να προσαρμοστεί ειδικά για κάθε είδους συσκευή (browser, κινητό τηλέφωνο, εκτύπωση).
- Δυναμικό περιεχόμενο
Επεκτάσεις όπως forums, ψηφοφορίες, έρευνες, αναζήτηση, ειδησεογραφικές εφαρμογές, ηλεκτρονικές αγορές είναι κάποια ενδεικτικά Modules.
- Συνεργασία
Ενθαρρύνεται η γρηγορότερη ενημέρωση, το αίσθημα υπευθυνότητας εφόσον το περιεχόμενο πρέπει να είναι εξουσιοδοτημένο, και τέλος, η συνεργασία μεταξύ των συντακτών.
- Χρονοπρογραμματισμός περιεχομένου
Η δημοσίευση του περιεχομένου μπορεί συχνά να γίνεται βάση χρόνου, δηλαδή το περιεχόμενο να είναι κρυμμένο και να γίνει χρήση του αργότερα, ή να απαιτείται η είσοδος του χρήστη δίνοντας το password του.

1.7 Μειονεκτήματα ενός CMS

1. Τα CMS απαιτούν κατανάλωση πολλών πόρων(resource hungry)

Το μειονέκτημα των συστημάτων CMS είναι ότι απαιτούν για τη λειτουργία τη χρήση πολλών πόρων και πηγών πληροφοριών. Οι servers λειτουργούν με την ανάγκη πάντοτε να υποστηρίζονται από τεχνικούς, και επιπλέον απαιτείται ακόμη περισσότερη μνήμη, επεξεργαστική ικανότητα και υποστήριξη λογισμικού. Ακόμη και μικρές αλλαγές στις ρυθμίσεις ενός εξυπηρετητή μπορούν να κάνουν ολόκληρο το σύστημα του CMS να αποτύχει και σαν επακόλουθο αυτού η ιστοσελίδα να μην μπορεί να είναι ορατή ή να προκληθούν λάθη σε αποτελέσματα.

CMS με κακογραμμένο κώδικα ή CMS που έχουν αναπτυχθεί με Access αντί για SQL θα επιβραδυνθούν με την αύξηση του αριθμού των χρηστών. Ακόμη και καλογραμμένες και δυναμικές ιστοσελίδες θα απασχολούν μεγαλύτερο μέρος της CPU και περισσότερη μνήμη σε σύγκριση με την ίδια την ιστοσελίδα ανεπτυγμένη όμως με HTML κώδικα.

2. Το hosting για ένα CMS κοστίζει ακριβά

Αν πραγματικά ένας πελάτης, ένας χρήστης ή μια εταιρία χρειάζονται ένα αξιόπιστο CMS, τότε θα πρέπει να περιμένουν να τους κοστίσει αρκετά χρήματα για να το έχουν. Χωρίς τεχνική υποστήριξη, κατάλληλη διαχείριση των κοινών χώρων όπου φιλοξενούνται και υποστήριξη λογισμικού, θα πρέπει κάποιος να περιμένει ότι θα αντιμετωπίσει προβλήματα. Ο συνδυασμός του γεγονότος ότι ένα αξιόπιστο CMS απαιτεί μεγάλη κατανάλωση σε πόρους με την ανάγκη να το εξυπηρετούν ικανοί και ειδήμονες τεχνικοί διατηρώντας και αναβαθμίζοντας το λογισμικό και το υλικό, σημαίνει ότι το hosting ενός CMS είναι μια ακριβή διαδικασία.

3. Τα CMS απαιτούν να αναβαθμίζονται ως προς το λογισμικό

Όπως και οποιοδήποτε προϊόν, έτσι και τα CMS κάποια στιγμή μπορούν να υποστούν υποτίμηση ως προς τις δυνατότητες τους. Δεν υπάρχουν εγγυήσεις ότι το CMS που θα αγοράσει κάποιος πελάτης σήμερα θα δουλεύει σε λίγα χρόνια από τώρα(ακόμη και σε 5 ή 10), χωρίς έναν προγραμματιστή να αφιερώσει χρόνο σε αναβαθμίσεις ώστε να το κάνει συμβατό με το υπάρχον αναβαθμισμένο και αλλαγμένο κάθε φορά λογισμικό. Νέες εκδόσεις των Windows, server software και των office ακόμη θα έχουν αντίκτυπο στη λειτουργία του CMS.

4. Οι μηχανές αναζήτησης δε τα χειρίζονται κατάλληλα (limited SEO - Search Engine Optimization)

Το πλεονέκτημα του να δημοσιεύσεις μια ιστοσελίδα είναι να αποκτήσεις το επιπλέον πλεονέκτημα του να εμφανίζεται αυτή στις μηχανές αναζήτησης, όμως δυστυχώς ένα μεγάλο μειονέκτημα πολλών CMS συστημάτων είναι ότι δεν δε τα χειρίζονται κατάλληλα οι μηχανές αναζήτησης, όσον αφορά τον τρόπο της κατάταξής τους σε κάποιο κατάλογο ή θέμα.

Οι ιστοσελίδες που περιέχει ένας CMS ιστότοπος παράγονται δυναμικά. Αυτό συχνά σημαίνει ότι οι URLS των ιστοσελίδων περιέχουν μεγάλες συμβολοσειρές που συνδιάζουν λέξεις, αριθμούς και σύμβολα. Οι μηχανές αναζήτησης έχουν σημαντικό πρόβλημα στο να αναλύσουν τις διευθύνσεις τέτοιου είδους σελίδων. Λίγοι διαδικτυακοί προγραμματιστές θα ενδιαφερθούν να μορφοποιήσουν τον κώδικα έτσι ώστε να κάνουν τη μηχανή αναζήτησης πιο συμβατή. Τα Meta tags που είναι αναγκαία ώστε να διασφαλίζεται μια καλή κατάταξη στα αποτελέσματα μιας αναζήτησης, πολύ συχνά και αυτά μένουν έξω από τέτοιου είδους σελίδες που δημιουργούνται δυναμικά.

5. Συντήρηση των αποτελεσμάτων των μηχανών αναζήτησης (SEO maintenance)

Μιας και οι σελίδες που έχουν δημιουργηθεί με CMS περιέχουν εκατοντάδες άλλων σελίδων μπορεί να είναι δύσκολη υπόθεση το να βελτιστοποιηθούν. Για παράδειγμα θα πάρει αρκετό χρόνο η αλλαγή των λέξεων κλειδιών για τόσες πολλές ηλεκτρονικές σελίδες. Οι λέξεις κλειδιά για μια ηλεκτρονική σελίδα πρέπει να βελτιώνονται με ακρίβεια και να μεταβάλλονται, ώστε να είναι πάντα συμβατές και σύγχρονες ως προς τα σημερινά κάθε φορά δεδομένα και να ακολουθούν το ρυθμό αλλαγής των αναζητήσεων που κάνουν οι άνθρωποι στο πέρασμα των χρόνων.

6. Καθυστέρηση στη φόρτωση

Εξαιτίας του γεγονότος ότι ένας ιστότοπος σε CMS είναι βασισμένος και καθοδηγούμενος από την βάση δεδομένων, θα πρέπει πάντα να περιμένουμε τον εξυπηρετητή να επεξεργαστεί τις ιστοσελίδες. Από τη στιγμή που ο αριθμός τους ανέρχεται σε εκατοντάδες από αυτές, είναι πολύ πιθανό να αργούν να φορτωθούν. Αυτό θα προκαλέσει το να χαθούν επισκέπτες και πελάτες, αφού όπως οι περισσότεροι χρήστες του διαδικτύου έχουν σύντομο εύρος προσοχής. Γι' αυτό το λόγο μία ιστοσελίδα θα έπρεπε να χρειάζεται

μόνο μερικά δευτερόλεπτα για να φορτώσει και να αναδείξει το περιεχόμενό της στον επισκέπτη.

7. Περιέχει εκατοντάδες αρχεία

Ένας ιστότοπος που περιέχει πολλά αρχεία αφήνει τον εαυτό του εκτεθειμένο σε σφάλματα. Για παράδειγμα, ένας πελάτης ανανεώνοντας τον ιστότοπο μπορεί να δημιουργήσει σφάλματα (όπως παραδείγματος χάρη κατά την εισαγωγή εικόνων που το μέγεθός τους είναι πολύ μεγάλο, κατά την εισαγωγή αρχείων flash, javascript ή βίντεο) το αποτέλεσμα των οποίων είναι μια ουσιαστική μεταβολή του τρόπου σχεδίασης του template .

Μπορεί να κοστίσει στον σχεδιαστή πολλές ώρες να βρει την αιτία του σφάλματος που προέκυψε με τέτοιο τρόπο. Υπάρχει πιθανότητα ακόμη και να πάει πίσω να ξανασχεδιάσει τη σελίδα όπως ήταν αρχικά, ειδικά στην περίπτωση όπου δεν είχε κρατηθεί αντίγραφο των αρχείων ως back up. Κάτι τέτοιο θα προκαλούσε απογοήτευση στον ιδιοκτήτη του ιστότοπου αφού αυτός δεν θα μπορεί για κάποιο χρονικό διάστημα να είναι ανεβασμένος στο διαδίκτυο. Μπορεί ακόμη να χάσει όλα τα αρχεία τα οποία είχε δημιουργήσει και δημοσιεύσει κι έτσι να χάσει και τους πελάτες του και επιπλέον να πρέπει να καταναλώσει πολύτιμο χρόνο και χρήμα πάλι από την αρχή ώστε να διορθώσει τα σφάλματα που προέκυψαν.

Συμπερασματικά, καταλαβαίνουμε ότι πριν αποφασίσει κάποιος να δημιουργήσει ένα τέτοιο σύστημα για την επιχείρησή του, θα πρέπει να προσδιορίσει προηγουμένως τις ανάγκες της. Αν η επιχείρηση είναι μικρή, όπου θα χρειαστούν λιγότερες ιστοσελίδες και όχι μεγάλος αριθμός, τότε συστήνεται ένας στατικός ιστοχώρος (static site) αντί για CMS. Αντίθετα στην περίπτωση που η επιχείρηση είναι μεγάλη και έχει εκατοντάδες ηλεκτρονικές σελίδες, οι οποίες θα χρειαστεί να προστεθούν ή να ανανεώνονται συχνά από πολλαπλούς συντάκτες, θα ήταν καλύτερο και περισσότερο εξυπηρετικό να δημιουργήσει ένα CMS site. Γενικότερα προτείνεται σε άτομα τα οποία ενδιαφέρονται να δημιουργήσουν μια ηλεκτρονική σελίδα την οποία θα εμπλουτίζουν συχνά και θα έχει πολλά περιεχόμενα, αλλά δεν θα ήθελαν να ξοδέψουν χρήματα σε προγραμματιστές και σχεδιαστές ιστοσελίδων.

Κάποια από τα δημοφιλέστερα Content Management Systems είναι τα εξής: Joomla!, Drupal, Wordpress, Mambo, Typo3, Moodle, PHP-BB, PHP-nuke.

1.8 Σύγκριση Συστημάτων Διαχείρισης Περιεχομένου

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, κάποια από τα δημοφιλέστερα αυτή την εποχή συστήματα CMS της αγοράς είναι τα Joomla!, Drupal, Mambo, PHP-nuke, Wordpress.

Ένα CMS, όπως είναι τα Drupal, Joomla, Wordpress, αφαιρεί από το προσκήνιο την ανάγκη για έναν σχεδιαστή σελίδων και βάζει τον χρήστη στη θέση του καθοδηγητή. Αυτό που κάνει το CMS λειτουργικό είναι να παράγει ένα σταθερό σεντ κώδικα, δηλαδή μια προκαθορισμένη βάση, το οποίο το περιβάλλει ένα template-πρότυπο, ώστε να είναι ευκολότερο και πιο εύχρηστο από τον χρήστη και τον επισκέπτη, και στη συνέχεια όλα αυτά τα μεταφράζει ως μια ιστοσελίδα περισσότερο φιλική και ευανάγνωστη, που όμως έχει δημιουργηθεί εύκολα και είναι εύκολα διαχειρίσιμη από τον χρήστη.

Σε αυτό το σημείο, η πρόσθεση από τον χρήστη ηλεκτρονικών σελίδων και πληροφοριών σε αυτή την προκαθορισμένη βάση, είναι τόσο εύκολο που γίνεται απλά συμπληρώνοντας κάποια textboxes και τσεκάροντας checkboxes, το καθένα από τα οποία είναι σε οργανωμένη, εμφωλευμένη ανάλογα με το θέμα και ευκόλως ακολουθούμενη από τον χρήστη δομή. Αρχίζοντας με μια βασική διαδικασία, αυτή της εγκατάστασης, ένα CMS κάνει απλό για οποιονδήποτε να δημιουργήσει και να διαχειριστεί έναν ιστότοπο.

Παρακάτω ακολουθεί μια σύγκριση ορισμένων από τα προαναφερθέντα CMSs, ώστε ο ενδιαφερόμενος χρήστης ανάλογα με τις ανάγκες που έχει να επιλέξει αυτό που θα τον βοηθήσει να πραγματοποιήσει τον στόχο του, χωρίς περιττές λειτουργίες.

1.8.1 Το κοινό χαρακτηριστικό

Καθένα από αυτά τα εκπληκτικά συστήματα είναι διαθέσιμα για χρήση από οποιαδήποτε επιχείρηση ή ατομική χρήση δωρεάν. Είναι όλα open source συστήματα, που αναπτύσσονται και ανανεώνονται σταθερά από ομάδες εθελοντών παγκοσμίως. Με αυτή την ιδιαιτερότητα, το λειτουργικό δεν θα είναι ποτέ ξεπερασμένο και έξω από τα «σημερινά» δεδομένα (out of date), καθώς οποιαδήποτε νέα ενημέρωση και πρόσθετη λειτουργικότητα θα είναι αυτό ακριβώς που θέλει και χρειάζεται ο ίδιος ο κόσμος. Η υποστήριξη για το λειτουργικό έρχεται στον χρήστη υπό τη μορφή μιας συνολικής-κοινοτικής προσπάθειας. Το σημαντικότερο είναι ότι το καθένα από αυτά τα συστήματα παρέχει στον χρήστη ένα εύχρηστο και εύκολα προσβάσιμο περιβάλλον, που κάνει το «χτίσιμο» ενός ιστότοπου αιφνιδιαστικά απλή υπόθεση. Εφόσον τα συστήματα αυτά συνδιάζουν α έτοιμα θέματα (φόντο) και πρόσθετα (add-on) χαρακτηριστικά, δεν είναι απαραίτητη η ανάγκη σχεδιαστή ιστοσελίδας ή γραφίστα, αφού ο καθένας μπορεί να δημιουργήσει μόνος του.



1.8.2 Σύγκριση των δημοφιλέστερων CMSs

CMS Comparison Chart

	Joomla	Drupal	Mambo	PHP-Nuke
Details	An award-winning CMS used to create websites and powerful applications. It is open-source and currently the most popular CMS available	An open-source CMS based on the powerful LAMP server model.	An open-source CMS used to create and administer data-driven websites	An open-source auto-mated news publishing CMS.
Advantages	- easy to install and set up - requires no programming skills - fully customizable - extensive support	- provides a rock-solid platform - easy to use multi-site feature - supports user groups and user permissions - great for membership sites	- automatic installation - exclusive features - supports large websites	- comprehensive interface - multi-user support - exclusive features
Disadvantages	- limited themes - heavy on server resources - manages a limited amount of sites - insufficient for membership sites	- different versions vary significantly - complex user interface - more difficult to manage	- more complex - heavy on server resources - known for bugs	- known for security issues - generated pages can be overlooked by search engines - requires some coding skills
This type of CMS Platform is right for you if...	lack web designing skills and need an easy to use solution	your project requires extreme flexibility	plan to create a large business site	if staff members need to manage areas of the site

Πίνακας 1 Σύγκριση των δημοφιλέστερων ΣΔΠ

Η Joomla! έχει ένα εντυπωσιακό βαθμό λειτουργικότητας και είναι πολύ ικανοποιητική για εταιρικές, εμπορικές, ακόμη και για προσωπικές ηλεκτρονικές σελίδες.

Το Drupal λέγεται ότι έχει τη μεγαλύτερη καμπύλη μάθησης από τα 3 γνωστότερα Συστήματα Διαχείρισης Περιεχομένου (βλ. Joomla!, Wordpress, Drupal), και οι χρήστες του υποστηρίζουν τη συνολική του δύναμη. Χρησιμοποιείται κυρίως για community sites, forums, weblogs και εταιρικές ιστοσελίδες.

Το Wordpress έχει γίνει η επιλογή για bloggers ανά τον κόσμο. Έχοντας δημιουργηθεί από χρήστες για χρήστες, το Wordpress κάνει μια εξαιρετική δουλειά παίρνοντας τη σύγχυση και δημιουργώντας καθ' όλα λειτουργικές ιστοσελίδες και weblogs. Η πλοήγηση είναι πάντοτε εύκολη και αμέτρητες ιδιότητες μπορούν να προστεθούν σε μία τέτοια ιστοσελίδα σαν πρόσθετα (plugins).

Το Moodle είναι ένα CMS που χρησιμοποιείται για να δημιουργεί και να δημοσιεύει online μαθήματα, όπως επίσης προσφέρεται και για ιστοσελίδες που έχουν ως θέμα την ηλεκτρονική εκμάθηση (web-based learning websites). Το όνομά του το έχει πάρει από τα αρχικά των λέξεων Modular Object-Oriented Dynamic Learning Environment, όμως και η λέξη Moodle από μόνη της σημαίνει 'περνώ την ώρα μου'. Αυτό το CMS είναι κατάλληλο για πανεπιστήμια, σχολεία και οργανισμούς οι οποίοι χρειάζονται να δουλεύουν με μονάδες ηλεκτρονικής μάθησης στο διαδίκτυο.

1.9 Ελεύθερο λογισμικό

Οι ιστοσελίδες που είναι δημιουργημένες με Συστήματα Διαχείρισης Περιεχομένου βασίζονται σε προγραμματιστικό επίπεδο σε κομμάτια κώδικα τα οποία επαναλαμβάνονται σε διάφορα σημεία μιας τέτοιας σελίδας, όπως επίσης και σε επεκτάσεις που φτιάχνονται από προγραμματιστές, δηλαδή κώδικα ο οποίος μπορεί να προστεθεί στην ιστοσελίδα για να βελτιώσει τη λειτουργικότητα ή να επεκτείνει το περιεχόμενο της, όπως θα δούμε σε επόμενο κεφάλαιο. Τέτοιες επεκτάσεις επαγγελματικές ή ερασιτεχνικές, διανέμονται στο διαδίκτυο και μπορεί να είναι εμπορεύσιμες όπου ο χρήστης τις αγοράζει ή να διαμοιράζονται για δωρεάν χρήση. Για να δημιουργηθούν νέες και βελτιωμένες εκδόσεις τέτοιων επεκτάσεων, είναι ευκολότερο για τους προγραμματιστές να γράψουν κώδικα ακολουθώντας κάποιο προϋπάρχον πρότυπο κώδικα, ώστε να τους ελαχιστοποιήσουν το χρόνο συγγραφής καθώς θα βελτιώνουν την αποτελεσματικότητά του. Όλες οι επεκτάσεις που υπάρχουν στο διαδίκτυο είναι ανοιχτού κώδικα και επιτρέπεται να χρησιμοποιούνται κατ' αυτό τον τρόπο, κάτι που στη συνέχεια βοηθάει όλη την κοινότητα χρηστών και προγραμματιστών που ασχολούνται με τα Συστήματα Διαχείρισης Περιεχομένου.

1.9.1 Ορισμός Ελεύθερου λογισμικού

Το ελεύθερο λογισμικό όπως ορίζεται από το Ίδρυμα Ελευθέρου Λογισμικού (Free Software Foundation), είναι λογισμικό που μπορεί να χρησιμοποιηθεί, αντιγραφεί, μελετηθεί, τροποποιηθεί και αναδιανεμηθεί χωρίς περιορισμό και χωρίς να απαιτείται η εξουσιοδότηση ή καταβολή κάποιου αντιτίμου για απόκτηση της σχετικής άδειας. Η ελευθερία από τέτοιους περιορισμούς είναι βασικό στοιχείο στην ιδέα του "ελεύθερου λογισμικού", έτσι ώστε το αντίθετο του ελεύθερου λογισμικού είναι το ιδιόκτητο λογισμικό, και όχι το λογισμικό που πωλείται για κέρδος, όπως το εμπορικό.

1.9.2 Άδειες ελεύθερου λογισμικού

Σύμφωνα με την ισχύουσα νομοθεσία περί πνευματικής ιδιοκτησίας, η ελεύθερη αντιγραφή, διανομή και τροποποίηση του λογισμικού δεν επιτρέπεται. Για το λόγο αυτό, οι εκδόσεις ελεύθερου λογισμικού κάνουν χρήση ειδικής άδειας σύμφωνα με την οποία, παραχωρείται το δικαίωμα αντιγραφής, τροποποίησης και αναδιανομής του λογισμικού, στους χρήστες.

Οι άδειες χρήσης ελεύθερου λογισμικού περιλαμβάνουν τις εξής ελευθερίες:

- Ελευθερία χρήσης του προγράμματος για οποιονδήποτε σκοπό.
- Ελευθερία μελέτης και τροποποίησής του.
- Ελευθερία αντιγραφής του.
- Ελευθερία βελτίωσης και επανέκδοσής του, προς το συμφέρον της κοινότητας των χρηστών.

Με βάση αυτή τη φιλοσοφία δημιουργήθηκε μια μεγάλη κοινότητα χρηστών και προγραμματιστών, οι οποίοι συνεργάζονται για τη συνεχή βελτίωση του λογισμικού, παρέχοντας γνώσεις και εργασία. Σήμερα υπάρχει ένα παγκόσμιο δίκτυο προγραμματιστών που κάνουν χρήση ανοικτού κώδικα, οι οποίοι παράλληλα αναπτύσσουν και διορθώνουν τον κώδικα των προγραμμάτων, κυκλοφορώντας νέες βελτιωμένες εκδόσεις λογισμικού.

1.9.3 Πλεονεκτήματα χρήσης

- Μείωση κόστους πληροφοριακών συστημάτων
- Δεν αγοράζονται άδειες χρήσης
- Δωρεάν ανανεώσεις
- Απεριόριστος αριθμός εγκαταστάσεων

- Δυνατότητα ίδιας παραμετροποίησης και επέκτασης
- Δυνατότητα ίδιας υποστήριξης
- Ύπαρξη πληθώρας επιλογών και δοκιμασμένων λύσεων
- Προώθηση του ανταγωνισμού των εταιρειών λογισμικού με αποτέλεσμα καλύτερες προσφερόμενες υπηρεσίες
- Αξιοπιστία, αφού το λογισμικό είναι δοκιμασμένο από πολλούς και αποφεύγονται αρνητικές εκπλήξεις και σφάλματα
- Ασφάλεια, αφού ο κώδικας μελετάται από πλήθος ανθρώπων, άρα τα όποια κενά ασφαλείας εντοπίζονται και διορθώνονται με μεγάλη ταχύτητα.

1.9.4 Μειονεκτήματα χρήσης

- Λιγότερες επιλογές επαγγελματικής υποστήριξης.
- Ασυμβατότητες με κάποια διαδεδομένα κλειστά πρότυπα αρχείων.
- Έλλειψη ορισμένων εξειδικευμένων εφαρμογών.

1.10 Επιλογή του CMS Joomla!

Σύμφωνα με προηγούμενη ενότητα όπου έγινε μια σύγκρισή των δημοφιλέστερων Συστημάτων Διαχείρισης Περιεχομένου, θα επιλέξουμε το Joomla για την κατασκευή του δικού μας ιστότοπου, αφού μας καλύπτουν τα χαρακτηριστικά του, προτείνεται για αρχάριους και είναι εύκολο στην εγκατάσταση.

1.10.1 Τρόπος λειτουργίας

Το Joomla εγκαθίσταται σε έναν κεντρικό υπολογιστή, τον web server. Ο διαχειριστής έχει πρόσβαση στο περιβάλλον διαχείρισης μέσω ενός browser. Στη συνέχεια έχει τη δυνατότητα να προσθέσει κείμενο, φωτογραφίες ή επεκτάσεις, και έτσι να διαμορφώσει τις ιστοσελίδες. Κάθε χρήστης του διαδικτύου έχει τη δυνατότητα να κατεβάσει το λογισμικό και να το εγκαταστήσει. Υπάρχουν διαθέσιμοι ιστότοποι υποστήριξης, που παρέχουν οδηγίες εγκατάστασης, extensions ανοιχτού κώδικα και υποστήριξη μέσα από τις οργανωμένες δικτυακές κοινότητες, και με αυτό τον τρόπο αυξάνεται η λειτουργικότητα της ιστοσελίδας και μπορούν να δημιουργηθούν ακόμα κι εφαρμογές πολύ αξιόπιστες για την ανάπτυξη ακόμα και εταιρικών δικτυακών τόπων. Το Joomla μπορεί να χρησιμοποιηθεί και από χρήστες που δεν κατέχουν εξειδικευμένες γνώσεις. Ωστόσο, ανάλογα με τις απαιτήσεις του

ιστότοπου που επιθυμεί να δημιουργήσει, ίσως απαιτηθεί και η συνδρομή πιο έμπειρου προσωπικού.

1.10.2 Πού χρησιμοποιείται

Το Joomla αποτελεί το κατάλληλο εργαλείο για την ανάπτυξη διαφόρων κατηγοριών ιστοσελίδων, παρέχοντας στο διαχειριστή εύκολη παραμετροποίηση με τη χρήση επεκτάσεων(extensions) όπως:

- Παρουσιάσεις επιχειρήσεων/συλλόγων
- Υπηρεσίες καταλόγων
- Επίδειξη εικόνων (photo gallery)
- Συνδρομητικές υπηρεσίες
- Ηλεκτρονικά καταστήματα
- Προβολή προϊόντων και υπηρεσιών
- Ξενοδοχεία, online κρατήσεις
- Προσωπικά ημερολόγια και βιογραφικά
- Blogs, forums
- Ηλεκτρονικές εκδόσεις περιοδικών και εφημερίδων
- Μεσιτικά γραφεία
- Αγγελίες

1.11 Επέκταση του Joomla!

1.11.1 Ορισμός

Joomla! είναι ένα ανοιχτού κώδικα Σύστημα Διαχείρισης Περιεχομένου που σχεδιάστηκε για να βοηθήσει τους χρήστες στην οικοδόμηση ιστοχώρων και άλλων online εφαρμογών. Το Joomla είναι μια συνέχεια της εργασίας της ομάδας ανάπτυξης που παραιτήθηκε ομόφωνα από το πρόγραμμα Mambo τον Αυγούστο του 2005. Η έκδοση Joomla 1.0 προέρχεται από Mambo 4.5.2.3 αλλά περιλαμβάνει πολλές επιπλέον διορθώσεις και μεγαλύτερο επίπεδο ασφάλειας. Το όνομα Joomla προέρχεται από τη σουαχίλι λέξη «Jumla», η οποία σημαίνει «όλα μαζί» ή «συνολικά» (all together as a whole).

Το Joomla μπορεί να εγκατασταθεί χειροκίνητα από τον πηγαίο κώδικα σε ένα σύστημα που τρέχει έναν κεντρικό υπολογιστή δικτύου (web server) που υποστηρίζει τις εφαρμογές PHP, από ένα σύστημα διαχείρισης συσκευασίας ή χρησιμοποιώντας μία συσκευή TurnKey Joomla που περιλαμβάνει την εφαρμογή και τις εξαρτήσεις της ως ένα έτοιμο προς χρήση σύστημα.

Το Joomla είναι γραμμένο σε PHP, χρησιμοποιεί αντικειμενοστρεφείς τεχνικές προγραμματισμού (Object-Oriented Programming) και πρότυπα σχεδίασης λογισμικού, αποθηκεύει τα δεδομένα σε μια MySQL βάση δεδομένων, και περιλαμβάνει χαρακτηριστικά όπως το page caching, οι εκτυπώσιμες εκδόσεις των σελίδων, οι ειδήσεις, οι ψηφοφορίες, η αναζήτηση, το ηλεκτρονικό εμπόριο, blogs, wikis, η διαχείριση εγγράφων, οι γκαλερί για την προβολή και επίδειξη εικόνων και τέλος παρέχει υποστήριξη για γλωσσική διεθνοποίηση.

Το Joomla! είναι λογισμικό που βοηθά την εύκολη δημιουργία και ενημέρωση μιας ιστοσελίδας.

Μπορούμε να σκεφτούμε ένα ιστοχώρο Joomla! ως συγκέντρωση τριών στοιχείων:

1. Το περιεχόμενο, το οποίο αποθηκεύεται κυρίως σε μια βάση δεδομένων
2. Το template, το οποίο ελέγχει τη διάταξη και την παρουσίαση του περιεχομένου (όπως το φόντο, τα χρώματα και το layout).
3. Το λογισμικό Joomla!, το οποίο είναι αυτό που συνδιάζει το περιεχόμενο και το template μαζί, για να παράγει ιστοσελίδες.

1.11.2 Οι λόγοι επέκτασης της Joomla

Η Joomla!, όχι μόνο σχεδιάστηκε για να χειρίζεται περιεχόμενο και κείμενο, αλλά επίσης για να επιτρέψει σε διάφορες σύνθετες εφαρμογές να ενσωματωθούν και συνενωθούν σαν ένα ενιαίο σύνολο. Τα καλάθια αγορών, τα φόρουμ, τα προφίλ των μέσων κοινωνικής δικτύωσης, κτηματομεσιτικά προγράμματα, είναι κάποια παραδείγματα των επεκτάσεων που οι προγραμματιστές έχουν γράψει για τη Joomla!. Όλα αυτά μπορούν να τρέξουν σε μία σελίδα Joomla, και μόνο μια βάση δεδομένων, ένα πρότυπο-template, και ο πυρήνας που θα τα συσχετίσει, χρειάζονται για να διατηρηθεί. Όταν χτίζεται μια επέκταση σε Joomla! , θα κληρονομήσει την εμφάνιση και την αίσθηση του ιστοχώρου γενικότερα. Οποιοσδήποτε τύπος προγράμματος που μπορεί να κωδικοποιηθεί PHP είναι ένα πιθανό component που μπορεί να γραφεί.

Οι επεκτάσεις μπορούν επίσης να είναι φορητές. Όταν κωδικοποιούνται σωστά, ο χρήστης είναι σε θέση εύκολα να εγκαταστήσει τον κώδικά σε ένα άλλο αντίγραφο Joomla! χωρίς να πρέπει να γίνει configuration στη βάση δεδομένων, να εισαχθούν logins και άλλες βασικές πληροφορίες

διαμόρφωσης πάλι από την αρχή. Επιπλέον, θα είναι σε θέση να διανείμει τις επεκτάσεις του σε άλλους, έτσι ώστε μπορούν να τις χρησιμοποιήσουν, χωρίς οποιαδήποτε γνώση προγραμματισμού ή βάσεων δεδομένων.

Συμπερασματικά, η Joomla μπορεί να επεκταθεί μέσω των components, των modules και των plug-ins. Αυτό επιτρέπει στον χρήστη να προσθέσει λειτουργίες σε μία τέτοια σελίδα, κι έπειτα να τις διατηρήσει ή να τις ενημερώσει, χωρίς να αλλάξει τον βασικό κώδικα.

1.11.3 Προσαρμογή εναντίον της επέκτασης

Ο κώδικας του Joomla! έχει δημιουργηθεί με σκοπό να επεκταθεί παρά να τροποποιηθεί. Είναι προτιμητέο να γραφτεί μια επέκταση, παρά να γίνουν αλλαγές στον κώδικα του πυρήνα. Όταν εκδίδονται αναπροσαρμογές και μπαλώματα για το ίδιο το Joomla!, ο κώδικας του πυρήνα θα ενημερωθεί, αλλά οι επεκτάσεις του χρήστη θα μείνουν ανέπαφες. Αυτές οι επεκτάσεις δημιουργούνται ώστε να επεξεργάζονται κατά τρόπο ανεξάρτητο, που επιτρέπει στο χρήστη να αναπτύξει ελεύθερα τον κώδικά του χωρίς τη διατάραξη άλλων στοιχείων παρόντων στην εγκατάσταση του Joomla!.

Αν και είναι ανεξάρτητες, οι επεκτάσεις δεν λειτουργούν σε ένα εντελώς σφραγισμένο περιβάλλον. Μπορεί ο χρήστης να αναμίξει διαφορετικά είδη για να πάρει τις λειτουργίες που επιθυμεί. Ο κώδικας του Joomla! επιτρέπει στις επεκτάσεις να μοιράζονται τους πόρους και να εκτελούν ενέργειες ή μια στην άλλη. Δεδομένου ότι μπορούμε να γράψουμε επεκτάσεις, κάνουμε αυτό αντί της προσαρμογής του πυρήνα.

1.11.4 Τρόποι επέκτασης του Joomla!

Υπάρχουν τρεις τύποι επεκτάσεων Joomla!, κάθε μια για μια συγκεκριμένη χρήση.

- Components

Από τις επεκτάσεις διαθέσιμες, τα components είναι τα πιο ουσιώδη. Είναι ουσιαστικά αυτό που είναι ορατό στο «κύριο» μέρος της σελίδας. Αυτό εξαρτάται από το σχέδιο και τη διάταξη του module που έχει επιλεγθεί να είναι σε λειτουργία. Τα components είναι πολύ βασικά στοιχεία για τη λειτουργικότητα του Joomla!, το οποίο σχεδιάστηκε να φορτώνει και να τρέχει ακριβώς ένα συστατικό για κάθε σελίδα που παράγεται. Αυτά τα βασικά στοιχεία περιλαμβάνουν banners, επικοινωνία, τροφοδοσία ειδήσεων, ψηφοφορίες και web links και επίσης μπορούν να ποικίλουν από μηχανές αναζήτησης μέχρι ολοκληρωμένες λύσεις ηλεκτρονικών αγορών. Κάθε component οδηγείται από ένα μενού και κάθε στοιχείο του μενού τρέχει ένα συστατικό. Η διαχείριση των components γίνεται μέσω των επιλογών του μενού Components της πλατφόρμας διαχείρισης του Joomla. Συνεπώς, καταλαβαίνουμε ότι η λειτουργία διαχείρισης περιεχομένου του Joomla!, είναι και αυτή η ίδια ένα component.

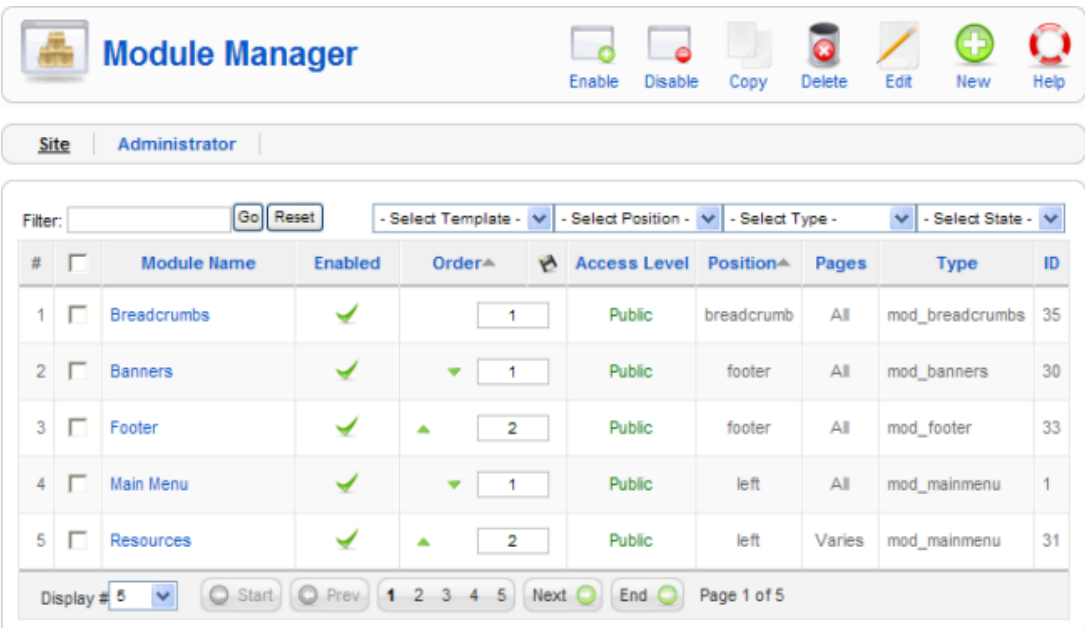
Κάθε συστατικό έχει δύο μέρη, το back-end που επιτρέπει στο διαχειριστή να διαμορφώσει και να διαχειριστεί το περιεχόμενό του και το front-end που επιδεικνύει τις πληροφορίες που έχουν επιλεγεί να φαίνονται στην ιστοσελίδα με τον επιλεγμένο τρόπο. Τα components συχνά έχουν σύνθετα back-end controls. Το back-end χρησιμοποιείται συνήθως για να δημιουργήσει και να ενημερώσει εγγραφές σε πίνακες της βάσης δεδομένων, όμως μπορεί να κάνει τυπικά τα πάντα, αρκεί να είναι προγραμματισμένο σε PHP. Για παράδειγμα, μπορεί να έχουμε μια εργασία που τρέχει από τη γραμμή εντολών του UNIX, αλλά μπορούμε να χρησιμοποιήσετε το back end για να παρέχουμε μια σύνδεση, link, όπου οι μη-προγραμματιστές μπορούν να την καλέσουν. Μπορεί επίσης να χρησιμοποιηθεί για να επιτρέψει στους διαχειριστές της σελίδας να ανεβάσουν εικόνες ή βίντεο.

- Modules

Σε αντίθεση με τα Components , οσαδήποτε Modules μπορούν να υπάρχουν σε μία σελίδα. Συμπληρώνουν το περιεχόμενο που περιέχεται σε ένα Component και δεν έχουν στόχο να είναι στην κύρια δομή της σελίδας.

Τα modules είναι ελαφριές και ευέλικτες επεκτάσεις που χρησιμοποιούνται για την απόδοση σελίδων. Μερικές φορές τα modules συνδέονται με τα components, για παράδειγμα το module «latest news» συνδέεται με το component που έχει το βασικό περιεχόμενο ειδήσεων και εμφανίζει links με τα νεότερα γεγονότα. Τα modules έχουν συνήθως τη μορφή «κουτιών» που διατάσσονται γύρω από ένα component σε μια ιστοσελίδα, ένα τέτοιο είναι το login module. Τα modules ορίζονται ανά menu item, έτσι, παραδείγματος χάριν, ο διαχειριστής μπορεί να αποφασίσει να παρουσιάσει ή να κρύψει το login module ανάλογα με το ποιο component (menu item) χρησιμοποιείται. Εντούτοις, αυτές οι επεκτάσεις δε χρειάζεται να συνδεθούν με components, όπως επίσης δε χρειάζεται να συνδεθούν με τίποτα, αφού μπορούν να είναι απλώς στατική HTML ή κείμενο.

Τα modules μπορούν να επιδείξουν τις πληροφορίες ακόμα κι αν δεν είναι συνδεδεμένες με ένα component. Στην πραγματικότητα μπορούν να επιδείξουν τα πάντα, από στατική HTML μέχρι σύνθετα slide shows φωτογραφιών. Ο χρήστης μπορεί να χειριστεί τα modules μέσω του μενού “Extensions” -> “Module Manager” στο back-end μέρος του Joomla (στο διαχειριστή) όπως και φαίνεται στη παρακάτω εικόνα:



Module Manager

Enable Disable Copy Delete Edit New Help

Site Administrator

Filter: [Go] [Reset] - Select Template - - Select Position - - Select Type - - Select State -

#	Module Name	Enabled	Order	Access Level	Position	Pages	Type	ID
1	Breadcrumbs	✓	1	Public	breadcrumb	All	mod_breadcrumbs	35
2	Banners	✓	1	Public	footer	All	mod_banners	30
3	Footer	✓	2	Public	footer	All	mod_footer	33
4	Main Menu	✓	1	Public	left	All	mod_mainmenu	1
5	Resources	✓	2	Public	left	Varies	mod_mainmenu	31

Display # 5 [Start] [Prev] 1 2 3 4 5 [Next] [End] Page 1 of 5

- Plug-Ins

Όταν ένα κομμάτι του κώδικα απαιτείται να υπάρχει σε όλη την ιστοσελίδα, είναι καλύτερο να εφαρμόζεται ως plug-in. Το plug-in χρησιμοποιείται συνήθως για να δώσει σχήμα και διάταξη στην εμφάνιση ενός component ή module όταν κατασκευάζεται μια σελίδα και προσθέτει επιπλέον λειτουργικότητα στον ιστοχώρο. Είναι σύνθετες επεκτάσεις που χειρίζονται τα διαφορετικά events.

Μερικά παραδείγματα plug-in περιλαμβάνουν τον τονισμό λέξης κλειδιού και τα πλαίσια σχολιασμού άρθρων. Ο χειρισμός τους γίνεται από το back-end, με παρόμοιο τρόπο με εκείνο των modules. Από το μενού “Extensions” -> “Plugin Manager” της πλατφόρμας διαχείρισης.

Οι τρεις προαναφερθέντες τύποι επέκτασης είναι οι βασικότεροι για την τελειοποίηση της λειτουργικότητας μιας ιστοσελίδας Joomla και τοποθετούνται σε επιλεγμένες θέσεις από τον διαχειριστή, ώστε να εξυπηρετούν τις ανάγκες του λόγου ύπαρξης της σελίδας και επιπλέον να την κάνουν ελκυστική για τον επισκέπτη. Όμως εκτός από τα παραπάνω υπάρχουν και κάποιοι ακόμη τύποι που η προσφορά τους είναι συνολική στη σελίδα και στον τρόπο εμφάνισής της. Αφού ορίζουν τη δομή και αρχιτεκτονική της και τον τρόπο εμφάνισης των στοιχείων, όπως η γλώσσα για παραδειγμα.

- Templates

Τα templates-πρότυπα καθορίζουν τον τρόπο ο ιστοχώρος φαίνεται, δηλαδή τη μορφή του. Ένα πρότυπο Joomla χειρίζεται εξ' ολοκλήρου τη διάταξη και το σχεδιασμό των σελίδων. Επιπλέον, το πρότυπο διευκρινίζει σε ποια μέρη της σελίδας εμφανίζονται τα διαφορετικά Modules που υπάρχουν. Ο διαχειριστής με τρόπο όπως εκείνο των υπόλοιπων προσθέτων που αναφέρθηκαν, μπορεί να αλλάξει το προεπιλεγμένο template του ιστοχώρου ως εξής: από το μενού “Extensions” -> “Template Manager”, που βρίσκεται στην περιοχή διαχείρισης.

- Languages

Οι γλώσσες είναι βασικά πακέτα που μεταφράζουν την ολόκληρη διεπαφή Joomla της πλατφόρμας διαχείρισης (back-end) ή/και το front-end του ιστοχώρου. Η διαχείριση των ήδη εγκατεστημένων γλωσσών στην εφαρμογή της Joomla γίνεται μέσω του μενού “Extensions” -> “Language Manager”.

ΚΕΦΑΛΑΙΟ 2

ΑΝΑΠΤΥΞΗ ΕΝΟΣ JOOMLA COMPONENT

2.1 Ορισμοί

Πριν προχωρήσουμε στο τεχνικό μέρος της εργασίας που ξεκινάει με την εγκατάσταση το πακέτου αρχείων του Joomla στον υπολογιστή, θα ήταν καλό να αναφέρουμε κάποιους ορισμούς που έχουν άμεση σχέση με τη διαδικασία ανάπτυξης μιας επέκτασης- extension. Πιο συγκεκριμένα θα αναφερθούμε στη γλώσσα προγραμματισμού που χρησιμοποιείται, στη διαχείριση βάσης δεδομένων και στον τρόπο μορφοποίησης της επέκτασης που θα συγγράψουμε.

Τι είναι η PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών και διαδραστικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor. Είναι μια server-side scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Είναι server-side επειδή εκτελείται στον διακομιστή, δηλαδή ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετάφορτώνεται σαν ένα κανονικό html έγγραφο ώστε να φανεί το παραγόμενο αποτέλεσμα, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

Η PHP δημιουργήθηκε το 1994 από τον Rasmus Lerdorf ο οποίος χρησιμοποίησε τη γλώσσα προγραμματισμού C προσπαθώντας να αντικαταστήσει ένα σύνολο από διάφορα μικρά προγράμματα (scripts) που είχε κατασκευάσει σε γλώσσα Perl με σκοπό τη συντήρηση της ιστοσελίδας του.

Τι είναι η MySql

Η MySQL είναι ένα πολύ γρήγορο και δυνατόσύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, που λειτουργεί σαν server και το οποίο μας επιτρέπει την αποθήκευση, αναζήτηση, ταξινόμηση και αποτελεσματική ανάκτηση των δεδομένων μας.

Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα, για να επιτρέψει την χρήση από πολλούς χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Η MySQL χρησιμοποιείται και εφαρμογές όπως γνωστά συστήματα διαχείρισης περιεχομένου(CMS, στα οποία θα αναφερθούμε στα επόμενα κεφάλαια) και πιο συγκεκριμένα τα Joomla, Wordpress, Drupal, αλλά και σε ιστοσελίδες όπως της Google, του youtube και του facebook.

Τι είναι το CSS

Τα αρχικά CSS προέρχονται από το Cascading Style Sheets δηλαδή αλληλουχία φύλλων στυλ. Είναι μια γλώσσα υπολογιστή που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης HTML ή XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας. Τα CSS έχουν σχεδιαστεί για να επιτρέπουν να διαχωρισμό του περιεχομένου της ιστοσελίδας σας από το σχεδιαστικό κομμάτι και αυτό της παρουσίασης. Αυτό είναι πολύ σημαντικό αφού τα στοιχεία σχεδίασης της ιστοσελίδας θα είναι σε ένα ξεχωριστό αρχείο το οποίο θα τροφοδοτεί τις υπόλοιπες σελίδες και κατ' αυτόν τον τρόπο βελτιώνεται η προσβασιμότητα του περιεχομένου. Επίσης, αφαιρώντας τα στοιχεία που μορφοποιούν μια σελίδα από τον κυρίως κώδικα, το αποτέλεσμα είναι να έχουμε πιο καθαρό και ευανάγνωστο κώδικα HTML, καλύτερη θέση εμφάνισης στις μηχανές αναζήτησης, και τέλος ο ελαχιστοποίηση του χρόνου που απαιτείται για αλλαγές στο κομμάτι της σχεδίασης αφού όλα τα στοιχεία βρίσκονται σε ένα αρχείο. Το CSS περιέχει στοιχεία όπως χρώματα, φόντο, layouts. Έτσι, όταν χρειαστεί να κάνουμε μία αλλαγή, όπως να αλλάξουμε το φόντο των σελίδων μας, μπορούμε να επεμβούμε στο αρχείο CSS και αυτομάτως οι αλλαγές θα επηρεάσουν και όλα τα αρχεία τα οποία συνδέονται με αυτό. Η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο ενώ τα CSS για να το μορφοποιήσουν. Στην HTML θα γράψουμε <h1>Επικεφαλίδα</h1> ενώ η μορφοποίησή της θα έρθει από το CSS:h1{color:red} που σημαίνει ότι το χρώμα της επικεφαλίδας θα είναι κόκκινο.

2.2 Εγκατάσταση

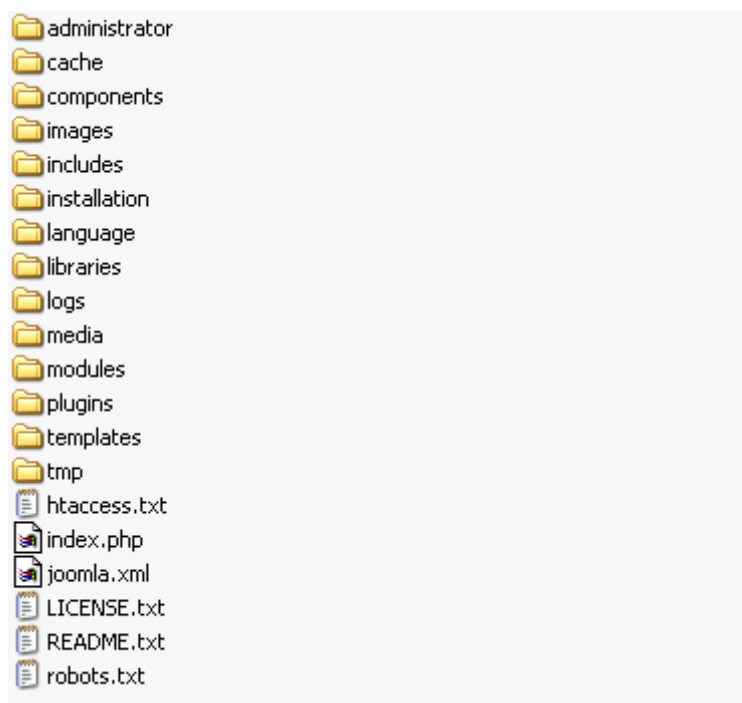
2.2.1 Προετοιμασία εγκατάστασης

Αρχικά, για την επιτυχημένη εγκατάσταση και χρήση του Joomla χρειάζεται ένας διακομιστής Web (Web server, εδώ θα χρησιμοποιηθεί το

Uniform Server) μία βάση δεδομένων (MySQL) και η γλώσσα προγραμματισμού PHP μαζί με ορισμένα modules που ενεργοποιούνται μέσω αυτής.

Βήμα 1: Το πρώτο πράγμα που θα κάνουμε είναι να αποκτήσουμε τον Uniform Server κατεβάζοντας το απαραίτητο αρχείο από την ηλεκτρονική διεύθυνση <http://www.uniformserver.com/>. Στη συνέχεια τον τοποθετούμε μέσα στον υπολογιστή μας στο δίσκο C, ώστε να μπορέσουμε αργότερα να προσθέσουμε μέσα στο φάκελο του σέρβερ τα αρχεία της Joomla και να ανεβάζουμε την ιστοσελίδα που θα δημιουργήσουμε, τοπικά, μέσω του browser που χρησιμοποιούμε.

Βήμα 2: Η δεύτερη κίνηση είναι να προμηθευτούμε ένα αντίγραφο του Joomla στη μορφή ενός αρχείου zip που θα βρούμε στην ηλεκτρονική διεύθυνση <http://www.joomla.org>. Η τελευταία έκδοση που χρησιμοποιήσαμε εδώ είναι η 1.6.1, όμως πλέον υπάρχει και η νεότερη και πιο πρόσφατη έκδοση του πακέτου 1.7. Στη συνέχεια κάνουμε αποσυμπίεση (extract) του αρχείου σε κάποια τοποθεσία της επιλογής μας, ώσπου αργότερα να χρειαστεί να το μεταφέρουμε τα αρχεία στο φάκελο ρίζα (root folder) του διακομιστή Web. Η λίστα των φακέλων που υπάρχουν μέσα στο αρχείο zip είναι η εξής:



Εικόνα 1 Αρχεία που περιλαμβάνονται στο φάκελο joomla

Βήμα 3: Στο τελευταίο βήμα της προετοιμασίας, πρέπει να "ανεβάσουμε" τα αποσυμπίεσμένα αρχεία στο διακομιστή Web. Στην περίπτωση τοπικής εγκατάστασης, πρόκειται για μια απλή διαδικασία αντιγραφής και επικόλλησης των αρχείων που είχαμε κάνει extract στο προηγούμενο βήμα, στο φάκελο ρίζα (root folder) του διακομιστή Web. Εδώ ο εν λόγω φάκελος είναι ο www.

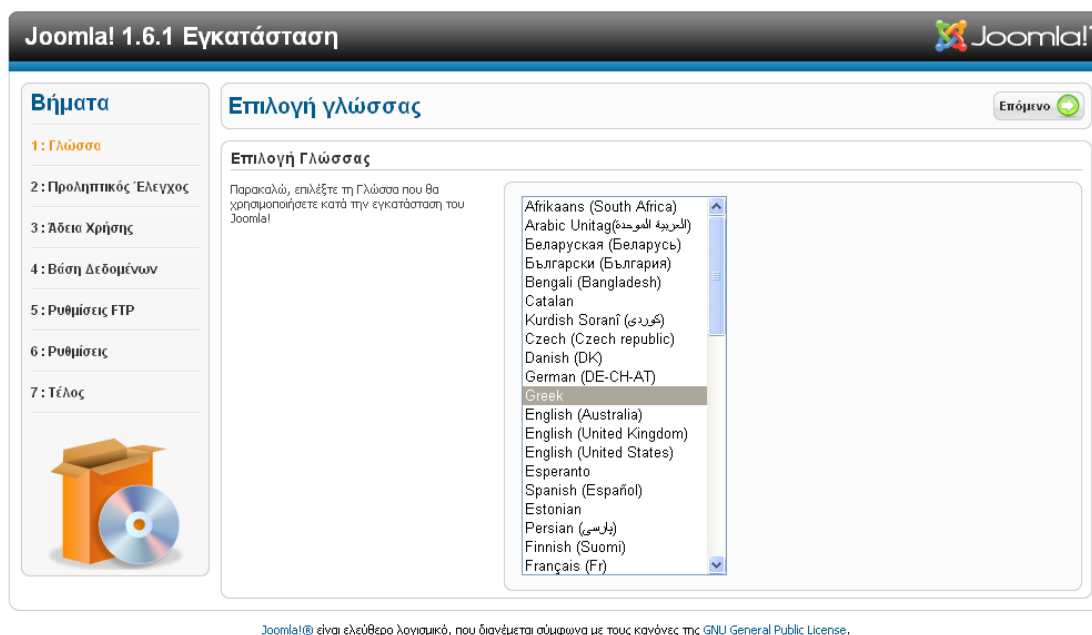
Βήμα 4: Σε αυτό το σημείο είμαστε έτοιμοι να προχωρήσουμε με την εγκατάσταση του προγράμματος Joomla στο διακομιστή Web. Υπάρχουν δύο τρόποι εγκατάστασης: εγκατάσταση μέσω φυλλομετρητή Web (Web Browser Installation) και χειροκίνητη εγκατάσταση (Manual Installation). Από τις δύο αυτές μεθόδους εγκατάστασης η πρώτη θεωρείται πιο εύκολη, λόγω του μικρότερου αριθμού βημάτων που απαιτούνται. Επιλέγουμε αυτή, που στην ουσιαστικά είναι μία ημιαυτόματη μέθοδος η οποία σε κάθε βήματος διαδικασίας ζητάει από τον χρήστη την εισαγωγή ορισμένων στοιχείων. Δηλαδή, μεγαλύτερο μέρος της δουλειάς γίνεται από το λογισμικό εγκατάστασης.

Υποθέτοντας ότι έχουμε ήδη "ανεβάσει" το πακέτο των αρχείων και φακέλων του Joomla στο φάκελο ρίζα του διακομιστή Web, είτε σε ένα τοπικό είτε σε ένα απομακρυσμένο οικοδεσπότη, το επόμενο βήμα είναι να εισάγουμε στη γραμμή διευθύνσεων του browser τη διεύθυνση URL της ιστοσελίδας μας. Για μια τοπική εγκατάσταση, όπως η δική μας, η διεύθυνση URL θα είναι της μορφής: `http://localhost/joomla`. Δηλαδή μετά το φάκελο του localhost δεν παρεμβάλουμε τον φάκελο www, αλλά απευθείας το φάκελο με τα αρχεία της joomla με το όνομα που επιλέξαμε να έχει. Το όνομα αυτό θα είναι αυτόματα και το όνομα της ιστοσελίδας μας.

2.2.2 Εγκατάσταση μέσω του περιηγητή

Επιλογή γλώσσας

Βήμα 5: Από τη στιγμή που θα εισάγουμε τη διεύθυνση URL το πρώτο βήμα για την εγκατάσταση του Joomla είναι η επιλογή της γλώσσας εγκατάστασης. Στο βήμα αυτό απλά επιλέγουμε τη γλώσσα προτίμησής μας για την εγκατάσταση.

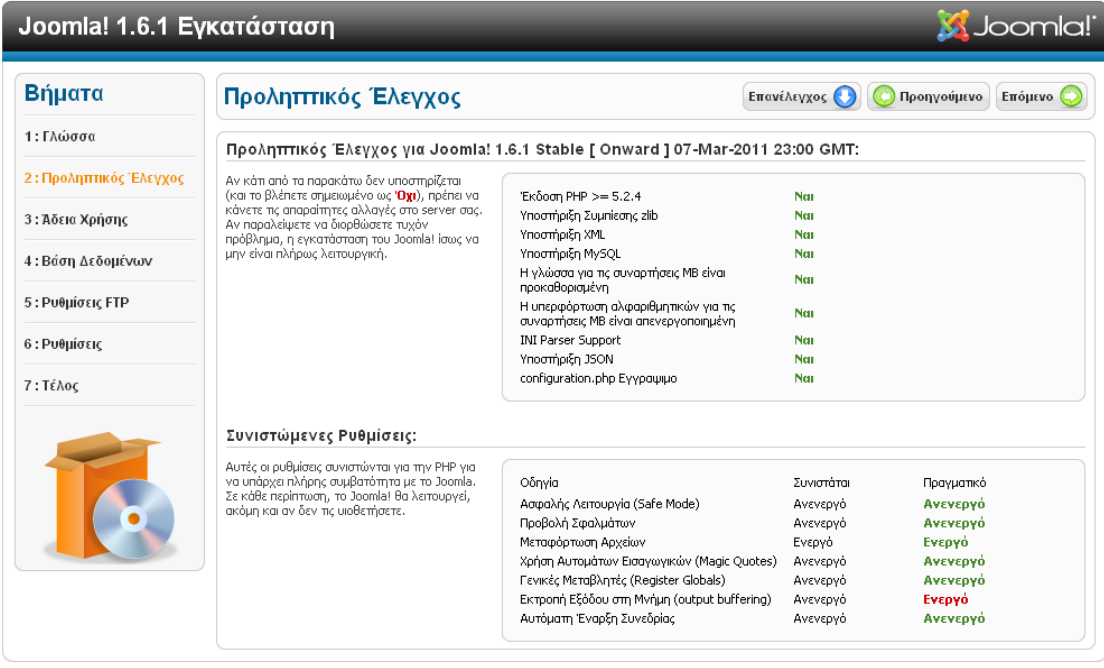


Εικόνα 2 Επιλογή γλώσσας

Η γλώσσα που είναι επιλεγμένη είναι η Αγγλική (en-GB -English(United Kingdom)). Αυτή είναι η εξ ορισμού γλώσσα που χρησιμοποιείται για την όλη ανάπτυξη του 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. Οι μεταφρασμένες αυτές εκδόσεις είναι διαθέσιμες στην ηλ. διεύθυνση <http://www.joomlancode.org>.

Προληπτικός έλεγχος

Βήμα 6: Η επόμενη οθόνη που εμφανίζεται στο φυλλομετρητή δείχνει αν ο διακομιστής μας πληροί τα απαραίτητα κριτήρια για την επιτυχημένη ολοκλήρωση της εγκατάστασης του Joomla!



Joomla! 1.6.1 Εγκατάσταση

Βήματα

- 1: Γλώσσα
- 2: Προληπτικός Έλεγχος**
- 3: Άδεια Χρήσης
- 4: Βάση Δεδομένων
- 5: Ρυθμίσεις FTP
- 6: Ρυθμίσεις
- 7: Τέλος

Προληπτικός Έλεγχος [Επενέλεγχος] [Προηγούμενο] [Επόμενο]

Προληπτικός Έλεγχος για Joomla! 1.6.1 Stable [Onward] 07-Mar-2011 23:00 GMT:

Αν κάτι από τα παρακάτω δεν υποστηρίζεται (και το βλέπετε σημειωμένο ως **Όχι**), πρέπει να κάνετε τις απαραίτητες αλλαγές στο server σας. Αν παραλείψετε να διορθώσετε τυχόν πρόβλημα, η εγκατάσταση του Joomla! ίσως να μην είναι πλήρως λειτουργική.

Έκδοση PHP >= 5.2.4	Ναι
Υποστήριξη Συμπίεσης zlib	Ναι
Υποστήριξη XML	Ναι
Υποστήριξη MySQL	Ναι
Η γλώσσα για τις συναρτήσεις MB είναι προκαθορισμένη	Ναι
Η υπερφόρτωση αλφαριθμητικών για τις συναρτήσεις MB είναι απενεργοποιημένη	Ναι
INI Parser Support	Ναι
Υποστήριξη JSON	Ναι
configuration.php Εγγράμιμο	Ναι

Συνιστώμενες Ρυθμίσεις:

Αυτές οι ρυθμίσεις συνιστώνται για την PHP για να υπάρχει πλήρης συμβατότητα με το Joomla!. Σε κάθε περίπτωση, το Joomla! θα λειτουργεί, ακόμη και αν δεν τις υιοθετήσετε.

Οδηγία	Συνιστάται	Πραγματικό
Ασφαλής Λειτουργία (Safe Mode)	Ανεργό	Ανεργό
Προβολή Σφαλμάτων	Ανεργό	Ανεργό
Μεταφόρτωση Αρχείων	Ενεργό	Ενεργό
Χρήση Αυτομάτων Εισαγωγικών (Magic Quotes)	Ανεργό	Ανεργό
Γενικές Μεταβλητές (Register Globals)	Ανεργό	Ανεργό
Εκτροπή Εξόδου στη Μνήμη (output buffering)	Ανεργό	Ενεργό
Αυτόματη Ένταξη Συνεδρίας	Ανεργό	Ανεργό

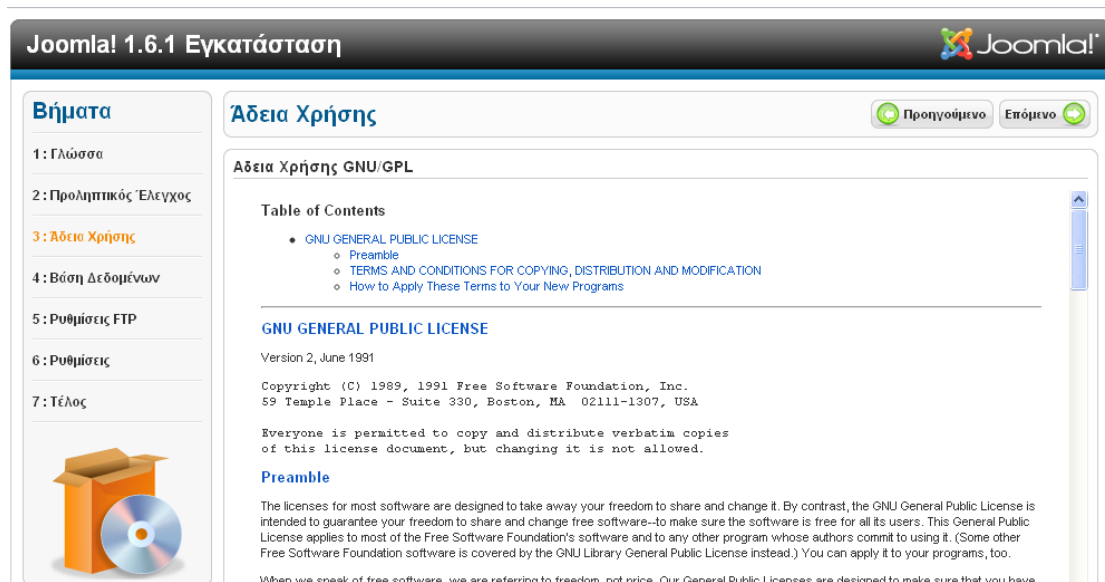
Joomla!® είναι ελεύθερο λογισμικό, που διανέμεται σύμφωνα με τους κανόνες της GNU General Public License.

Εικόνα 3 Προληπτικός έλεγχος

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, επειδή αυτά τα κριτήρια είναι βασικές προϋποθέσεις ώστε να επιτευχθεί η εγκατάσταση.

Άδεια

Βήμα 7: Στο βήμα αυτό δε χρειάζεται να συμπληρώσουμε ή να τροποποιήσουμε κάτι. Ωστόσο είναι καλό να διαβάσουμε, να κατανοήσουμε και να συμφωνήσουμε με τους όρους και τις προϋποθέσεις κάτω από τις οποίες εκδίδεται το Joomla.



Εικόνα 4 Άδεια χρήσης

Ρύθμιση βάσης δεδομένων

Βήμα 8: Κάπου εδώ ξεκινούν πλέον στα πιο σημαντικά στάδια της διαδικασίας εγκατάστασης. Στο βήμα αυτό δίνουμε της πληροφορίες που απαιτούνται, έτσι ώστε το Joomla να μπορεί να επικοινωνήσει με τη βάση δεδομένων.

Στο στάδιο αυτό συμπληρώνουμε τις πληροφορίες που εισάγαμε κατά τη διαδικασία εγκατάστασης και ρύθμισης της βάσης δεδομένων μας. Πιο συγκεκριμένα, επιλέγουμε πρώτα τον τύπο της βάσης δεδομένων από την αντίστοιχη λίστα, η οποία περιλαμβάνει τους τύπους mysql και mysqli εξ ορισμού. Εμείς θα επιλέξουμε τον τύπο mysql.

Συμπληρώνουμε τα στοιχεία Host Name (Όνομα Οικοδεσπότη), User Name (Όνομα Χρήστη), Password (Κωδικός Πρόσβασης) και Database Name (Όνομα Βάσης Δεδομένων), τα οποία είχαμε εισάγει κατά τη διαδικασία εγκατάστασης και ρύθμισης της βάσης δεδομένων. Καλό είναι να ονομάσουμε τη βάση ακριβώς όπως και την σελίδα μας.

Τα πεδία Advanced Settings επιτρέπουν περαιτέρω διαμόρφωση της βάσης δεδομένων. Παραδείγματος χάριν, αν υπάρχει μία παλαιότερη βάση δεδομένων με το ίδιο όνομα με αυτό που βάλαμε στο πεδίο Database Name, την οποία όμως δε χρειαζόμαστε πια, τότε το σύστημα μας δίνει τη δυνατότητα να διαγράψουμε (drop) τους πίνακες αυτής της βάσης ή να δημιουργήσουμε αντίγραφα ασφαλείας.

Μπορούμε επίσης να αλλάξουμε το Πρόθεμα Πίνακα (Table Prefix) το οποίο το Joomla! θα χρησιμοποιήσει για όλους τους πίνακες της βάσης. Η δυνατότητα αυτή είναι χρήσιμη σε περίπτωση που διαθέτουμε περισσότερες από μία ιστοσελίδες μέσα στην ίδια βάση δεδομένων. Το default πρόθεμα πίνακα είναι το jos_.

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

Joomla! 1.6.1 Εγκατάσταση

Βήματα

- 1: Γλώσσα
- 2: Προληπτικός Έλεγχος
- 3: Άδεια Χρήσης
- 4: Βάση Δεδομένων**
- 5: Ρυθμίσεις FTP
- 6: Ρυθμίσεις
- 7: Τέλος

Ρυθμίσεις Βάσης Δεδομένων

Ρυθμίσεις Σύνδεσης

4 απλά βήματα ακόμη...

Δώστε το hostname του server όπου θα εγκατασταθεί το Joomla!

Δώστε το Όνομα Χρήστη MySQL, τον κωδικό και τη Βάση Δεδομένων που θα χρησιμοποιήσετε για το Joomla!

Δώστε ένα πρόθεμα για τους πίνακες της βάσης της συγκεκριμένης εγκατάστασης του Joomla!. Επιλέξτε τι θέλετε να γίνει με υφιστάμενους πίνακες τυχόν προηγούμενης εγκατάστασης.

Βασικές Ρυθμίσεις

Είδος Βάσης Δεδομένων *
MySQL *Είναι συνήθως "MySQL"*

Όνομα Διακομιστή *
localhost *Είναι συνήθως "localhost"*

Όνομα Χρήστη *
root *Το Όνομα Χρήστη της Βάσης Δεδομένων*

Κωδικός
.... *Για λόγους ασφαλείας, η χρήση κωδικού για την mysql είναι απαραίτητη*

Όνομα Βάσης Δεδομένων *
hotel *Μερικοί hosts επιτρέπουν έναν ορισμένο αριθμό βάσεων δεδομένων. Στην περίπτωση αυτή, χρησιμοποιήστε το πρόθεμα (table prefix) ώστε να ξεχωρίσετε διαφορετικά Joomla! sites.*

Πρόθεμα πίνακα *
jos_ *Επιλέξτε ένα πρόθεμα για τους πίνακες της βάσης δεδομένων. Καλό είναι να αποτελείται από 3 ή 4 αλφαριθμητικούς χαρακτήρες και να τελειώνει σε κάτω παύλα (underscore). Μην χρησιμοποιήσετε το πρόθεμα "bak_" επειδή είναι δεσμευμένο για τα αντίγραφα των πινάκων.*

Βάση Δεδομένων από προηγούμενη

Εικόνα 5 Ρυθμίσεις βάσης δεδομένων

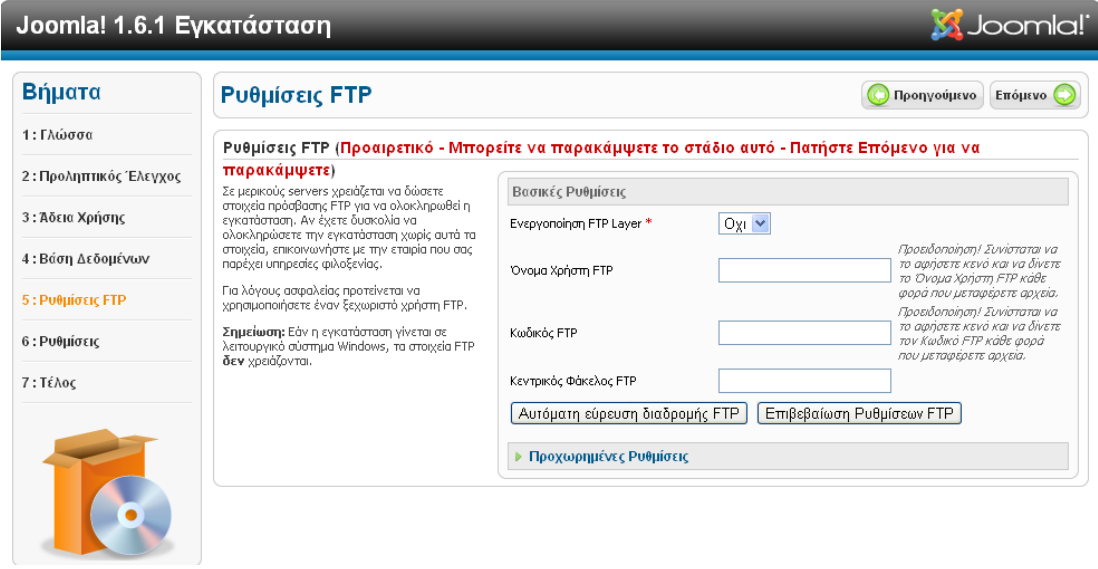
FTP

Βήμα 9: Η λειτουργία αυτή έρχεται για να λύσει προβλήματα που συναντώνται, λόγω περιορισμών στα δικαιώματα (permissions) των αρχείων, σε ορισμένα συστήματα Linux. Επιτρέπει στους χρήστες να χρησιμοποιούν τις διαδικασίες εγκατάστασης (installation) και φόρτωσης (upload) που υπάρχουν στον Πίνακα Ελέγχου (Control Panel) του συστήματος. Το στρώμα FTP υπάρχει με σκοπό την καλύτερη διαχείριση του συστήματος αρχείων του Joomla.

Στο βήμα αυτό συμπληρώνουμε το Όνομα Χρήστη (User Name) και τον Κωδικό Πρόσβασης (Password) για το FTP, στοιχεία τα οποία εισάγαμε κατά τη διαδικασία ρύθμισης και εγκατάστασης του διακομιστή FTP. Με τη ρύθμιση αυτή εξασφαλίζουμε πρόσβαση στον κατάλογο ρίζα του Joomla.

Πρόκειται για ένα λογαριασμό FTP που διαχειρίζεται όλες τις διαδικασίες του συστήματος αρχείων του Joomla.

Το βήμα της ρύθμισης του FTP δεν είναι απαραίτητο στην περίπτωση τοπικής εγκατάστασης, με εξαίρεση ίσως την εγκατάσταση σε τοπικό δίκτυο. Μόλις ολοκληρώσουμε τη διαδικασία ρύθμισης, ή αν δε χρειαζόμαστε στρώμα FTP, πατάμε το πλήκτρο Next.



Joomla! 1.6.1 Εγκατάσταση

Βήματα

- 1: Γλώσσα
- 2: Προληπτικός Έλεγχος
- 3: Άδεια Χρήσης
- 4: Βάση Δεδομένων
- 5: Ρυθμίσεις FTP**
- 6: Ρυθμίσεις
- 7: Τέλος

Ρυθμίσεις FTP

Ρυθμίσεις FTP (Προαιρετικό - Μπορείτε να παρακάμψετε το στάδιο αυτό - Πατήστε Επόμενο για να παρακάμψετε)

Σε μερικούς servers χρειάζεται να δώσετε στοιχεία πρόσβασης FTP για να ολοκληρωθεί η εγκατάσταση. Αν έχετε δυσκολία να ολοκληρώσετε την εγκατάσταση χωρίς αυτά τα στοιχεία, επικοινωνήστε με την εταιρία που σας παρέχει υπηρεσίες φιλοξενίας.

Για λόγους ασφαλείας προτείνεται να χρησιμοποιήσετε έναν ξεχωριστό χρήστη FTP.

Σημείωση: Εάν η εγκατάσταση γίνεται σε λειτουργικό σύστημα Windows, τα στοιχεία FTP δεν χρειάζονται.

Βασικές Ρυθμίσεις

Ενεργοποίηση FTP Layer *

Όνομα Χρήστη FTP

Κωδικός FTP

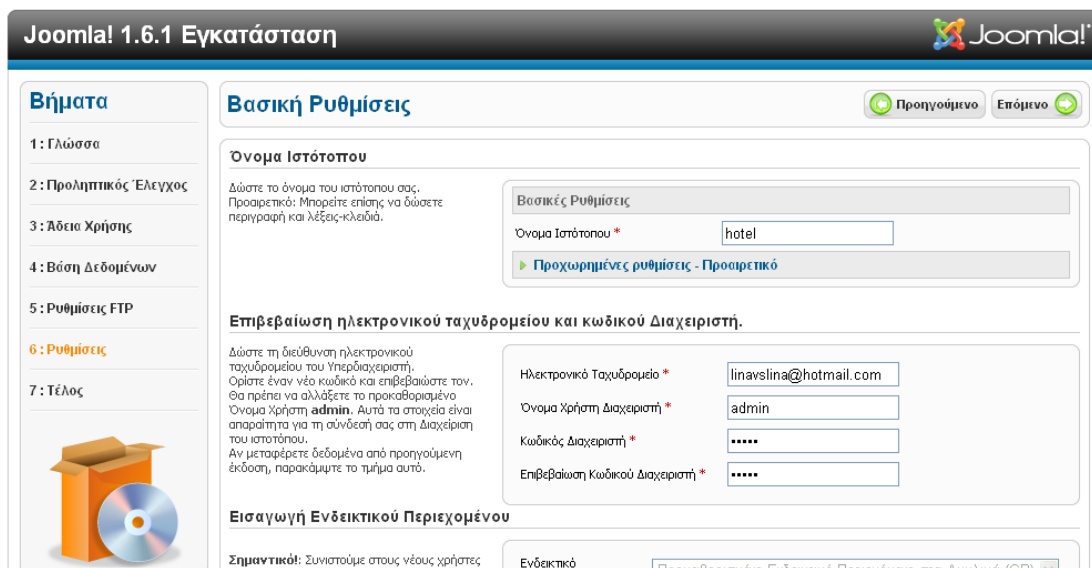
Κεντρικός Φάκελος FTP

[Προχωρήστε](#)

Εικόνα 6 Ρύθμιση FTP (προαιρετικό)

Κύρια ρύθμιση

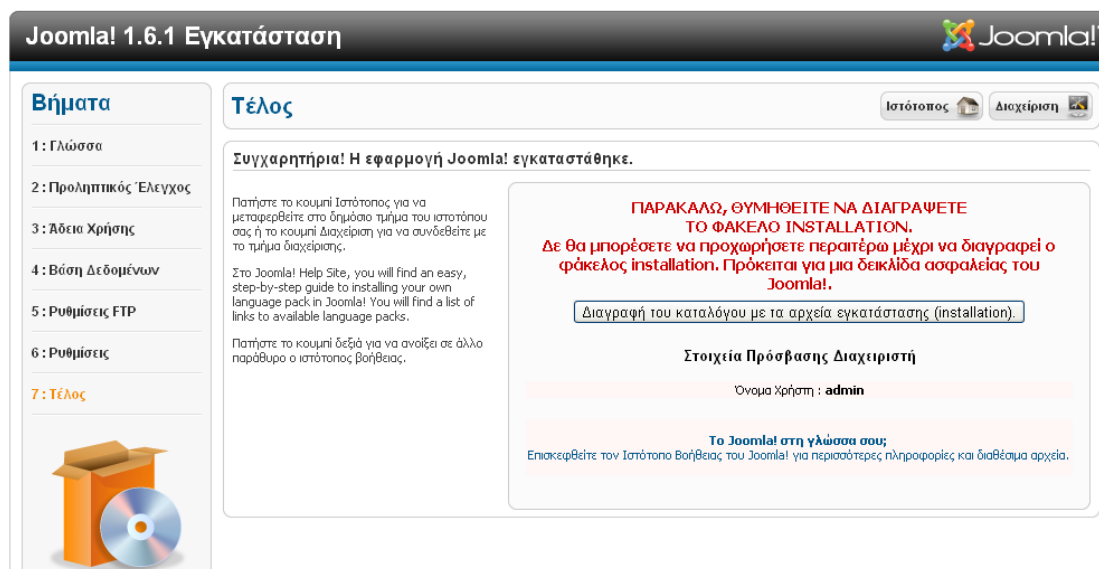
Βήμα 9: Η οθόνη Κύριας Ρύθμισης επιτρέπει τη ρύθμιση τριών κρίσιμων στοιχείων της εγκατάστασης του Joomla στο διακομιστή Web.



Εικόνα 7 Βασικές Ρυθμίσεις

Τρία τμήματα στην οθόνη Κύριας Ρύθμισης:

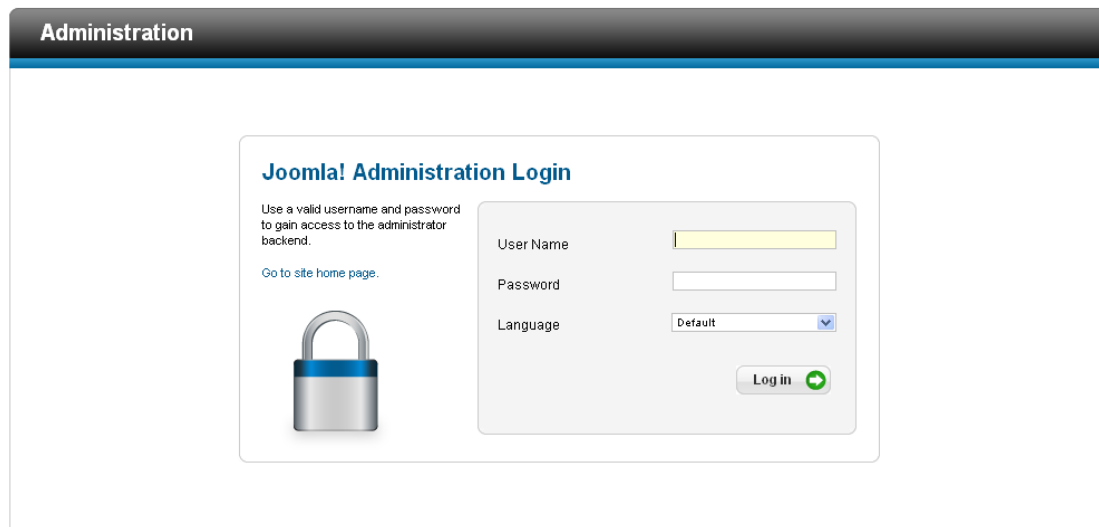
1. Το όνομα της ιστοσελίδας (Site Name). Πέρα από τη χρήση του ως ονομασία της ιστοσελίδας, αποτελεί επίσης και το όνομα που θα χρησιμοποιείται στις περιπτώσεις αποστολής e-mail από την ιστοσελίδα, εμφάνισης λαθών και μηνυμάτων προειδοποίησης. Το όνομα αυτό δεν εμφανίζεται στην μπάρα τίτλου του φυλλομετρητή, αλλά υπάρχει η δυνατότητα ρύθμισης από τον Πίνακα Ελέγχου του συστήματος.
2. Αφορά τον υπερδιαχειριστή (Super Administrator) του συστήματος. Στο τμήμα αυτό συμπληρώνουμε το email. Υπάρχουν επίσης και δύο πεδία στα οποία συμπληρώνουμε και επιβεβαιώνουμε τον κωδικό με τον οποίο θα έχουμε πρόσβαση στην ιστοσελίδα ως υπερδιαχειριστές, σε συνδυασμό με το εξ' ορισμού όνομα χρήστη για τον υπερδιαχειριστή που είναι admin.
3. Αφορά με τα δεδομένα τα οποία θέλουμε να εισάγουμε στη σελίδα μας μέσω της διαδικασίας εγκατάστασης. Στην ουσία έχουμε τρεις επιλογές: 1) Να εισάγουμε τα εξ ορισμού δεδομένα δείγματος (default sample data) που περιέχονται μέσα στο πακέτο του Joomla. 2) Να μεταφέρουμε δεδομένα από κάποια προηγούμενη έκδοση του Joomla, χρησιμοποιώντας τα διαθέσιμα "εργαλεία μετανάστευσης" (migration tools). 3) Να μην κάνουμε τίποτα από τα δύο και να ξεκινήσουμε μία καινούργια και άδεια εγκατάσταση του Joomla. Αφού έχουμε κάνει την επιλογή μας μπορούμε να προχωρήσουμε στο επόμενο βήμα πατώντας Next.



Εικόνα 8 Τέλος εγκατάστασης (μήνυμα για διαγραφή του αρχείου installation)

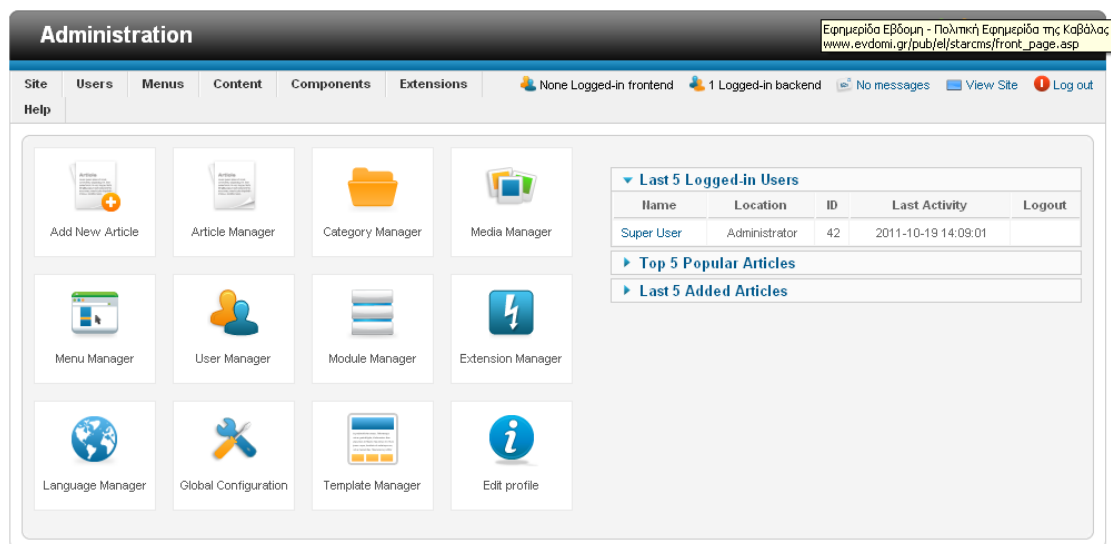
Βήμα 10: Η εγκατάστασή μας έχει ολοκληρωθεί με επιτυχία και είμαστε πλέον έτοιμοι να προχωρήσουμε με τη διαχείριση της ιστοσελίδας μας. Πριν προχωρήσουμε όμως είναι απαραίτητο, και εμφανίζεται και με γραπτό μήνυμα προειδοποίησης, να ανοίξουμε το φάκελο ρίζα του Joomla και να σβήσουμε το φάκελο Installation. Εάν ο φάκελος αυτός δεν αφαιρεθεί, δε θα μπορέσουμε να προχωρήσουμε στην επόμενη οθόνη και να ολοκληρώσουμε την εγκατάσταση.

Μόλις σβήσουμε το φάκελο, μπορούμε να επιλέξουμε ανάμεσα στις επιλογές Site ή Διαχείριση (Admin). Το πρώτο μας μεταφέρει στην αρχική σελίδα της ιστοσελίδας μας (το Front-end) και το δεύτερο στη σελίδα διαχείρισης (Administration) του συστήματος (το Back-end).



Εικόνα 9 Administration login (για να εισέλθουμε στο back end)

Στην οθόνη αυτή συμπληρώνουμε το όνομα χρήστη admin και τον κωδικό πρόσβασης που εισάγαμε στο βήμα 6 της διαδικασίας εγκατάστασης, και πατάμε το πλήκτρο Login. Μεταφερόμαστε στο back-end απ' όπου ο σαν διαχειριστές ρυθμίζουμε και διαχειριζόμαστε τη σελίδα μας και βλέπουμε την παρακάτω οθόνη:



Εικόνα 10 Οθόνη back end όπου ο διαχειριστής πραγματοποιεί ρυθμίσεις για το site

2.3 Δομή ενός Component

Η Joomla υιοθετεί ένα συγκεκριμένο σχήμα ονομάτων, το οποίο χρησιμοποιείται από όλα τα components. Κάθε component στο σύστημα έχει ένα μοναδικό όνομα χωρίς διαστήματα. Ο κώδικας είναι χωρισμένος σε δύο φακέλους, κάθε ένας με το όνομα του component που προτάσσεται από com_. Τα components καλούνται reviews. Επομένως, θα πρέπει να δημιουργηθούν δύο φάκελοι που ονομάζονται com_reviews:

- Δημιουργούμε ένα στο φάκελο που ονομάζεται τα components για το front-end.
- Δημιουργούμε ένα στο φάκελο που ονομάζεται τα components μέσα στο φάκελο administrator για το back-end.

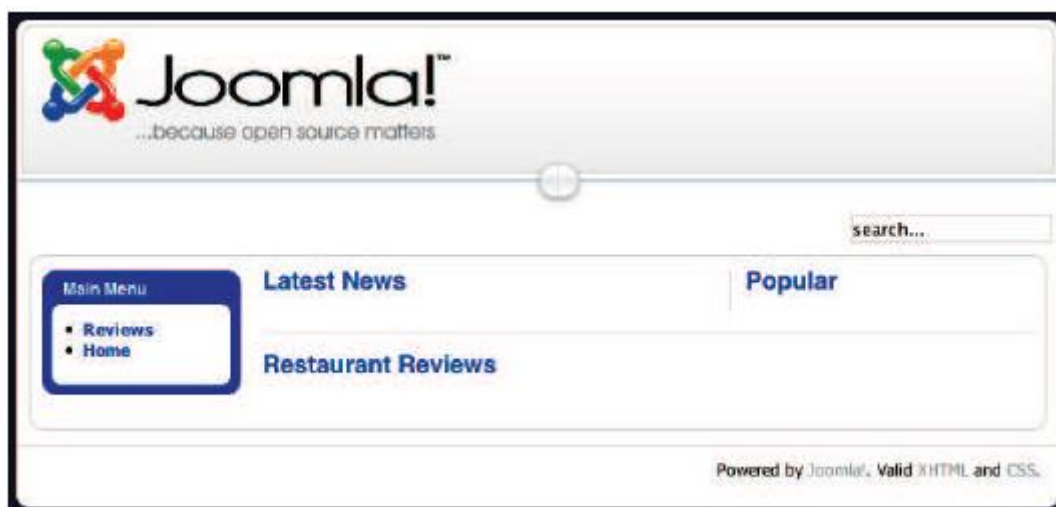
Όταν το συστατικό φορτώνεται από το front-end, το Joomla! θα ψάξει ένα αρχείο με το μοναδικό όνομα του component που τελειώνει με μια .php επέκταση. Μέσα στο φάκελο components/com_reviews δημιουργούμε ένα φάκελο reviews.php. Ομοίως, αν το τρέξουμε στο back-end προϋποθέτει την παρουσία ενός αρχείου που προλογίζεται με το admin. ακολουθούμενος από το όνομα του component και τελειώνει σε .php. Προσθέτουμε το φάκελο admin.reviews.php μέσα στο administrator/components/com_reviews. Και αφήνουμε τους 2 φακέλους άδειους προς το παρόν.

2.5 Εκτελώντας το Component

Όλα τα αιτήματα του front-end στη Joomla περνούν μέσα από το index.php που βρίσκεται στο root directory. Διαφορετικά components μπορούν να φορτωθούν, καθορίζοντας τη μεταβλητή option στο URL GET string. Εάν εγκαταστήσουμε το Joomla! σε έναν τοπικό web server σε ένα κατάλογο με τον τίτλο joomla, το URL για την πρόσβαση στη σελίδα θα είναι: <http://localhost/joomla/index.php>. Υποθέτοντας ότι συμβαίνει αυτό, μπορούμε να φορτώσουμε το front-end του component ανοίγοντας στον περιηγητή μας το: http://localhost/joomla/index.php?option=com_reviews. Σε αυτό το σημείο, η οθόνη πρέπει να είναι ουσιαστικά κενή, εκτός από την ύπαρξη των στοιχείων του template και τα modules. Για να κάνουμε αυτό το component ελαφρώς πιο χρήσιμο, ανοίγουμε το reviews.php και προσθέτουμε τον ακόλουθο κώδικα, κι έπειτα κάνουμε ανανέωση στον browser:

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

Η οθόνη τώρα θα δείχνει περίπου ίδια με την εξής:



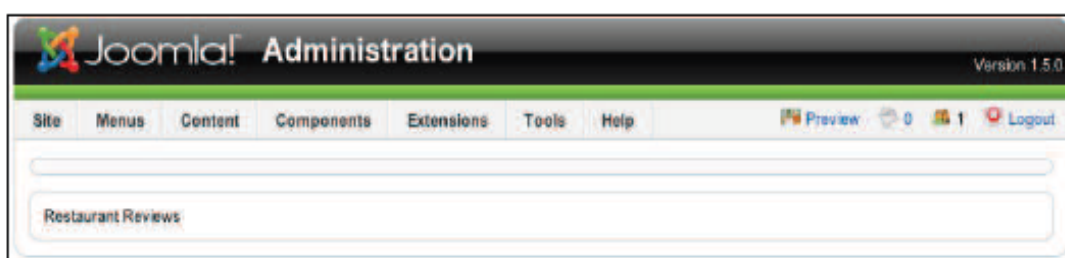
Καλέσαμε την `defined()` στην αρχή του αρχείου, για να γίνει ο έλεγχος που θα μας εξασφαλίσει ότι ο κώδικας καλείται μέσω Joomla! αντί να προσεγγιστεί άμεσα στα `components/com_reviews/reviews.php`. Η Joomla! αυτόματα διαμορφώνει το περιβάλλον με μερικά μέτρα ασφάλειας που μπορούν να νικηθούν εάν κάποιος είναι σε θέση να εκτελέσει άμεσα τον κώδικα για το component μας

Για το back end βάζουμε τον παρακάτω κώδικα μέσα στο `administrator/components/com_reviews/admin.reviews.php`:

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

Πηγαίνουμε στο:

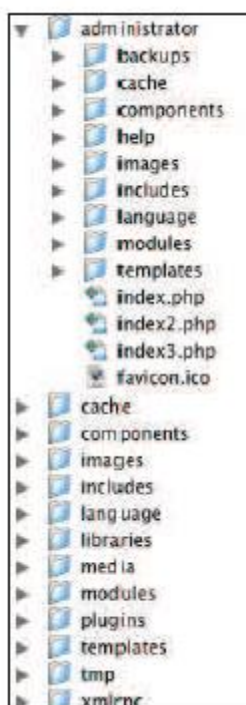
http://localhost/joomla/administrator/index.php?option=com_reviews, και θα πρέπει να εμφανίζεται κάτι τέτοιο:



2.6 Διαχωρισμός μεταξύ frond-end και back-end

Για όλα τα components της Joomla! ο κώδικας που εξουσιοδοτεί το τμήμα του back-end κρατιέται μακριά από τον frond-end κώδικα. Σε μερικές περιπτώσεις, όπως είναι η database table class, το back-end θα χρησιμοποιήσει ορισμένα αρχεία από το frond-end, αλλά σε κάθε άλλη περίπτωση αυτά τα δύο είναι χωριστά. Η ασφάλεια ενισχύεται δεδομένου ότι είναι λιγότερο πιθανό να ρίξει κάποιος τις διοικητικές λειτουργίες(back-end) στον frond-end κώδικα. Αυτό είναι μια σημαντική διάκριση δεδομένου ότι το frond-end και το back-end είναι παρόμοια στη δομή.

Το ακόλουθο διάγραμμα αρχείων παρουσιάζει το root της Joomla με το φάκελο administrator επεκταμένο:



Παρατηρούμε ότι ο φάκελος administrator έχει παρόμοια δομή και περιεχόμενο με τον root, γι' αυτό είναι σημαντικό να κάνουμε τη διάκριση μεταξύ των δύο τους, ώστε να μη βάλουμε τον κώδικα στο λάθος φάκελο γιατί τότε θα αποτύχει να εκτελεστεί μέχρι να πάμε πάλι πίσω και να τον μεταφέρουμε στο σωστό αυτή τη φορά.

2.7 Καταχώρηση του component στη βάση δεδομένων (Registering)

Τώρα γνωρίζουμε πώς να έχουμε πρόσβαση στο front-end και το back-end του component. Αν και θα μπορούσαμε να συνεχίσουμε να πληκτρολογούμε τα URLs κάθε φορά που θέλουμε να εκτελέσουμε ένα κομμάτι του κώδικα, αυτό δεν θα ήταν αποδεκτό για τους χρήστες. Η περιήγηση μπορεί να παρασχεθεί εάν καταχωρήσουμε το component στη βάση δεδομένων με την προσθήκη μιας σειράς στον πίνακα του component.

Θα εκτελέσουμε αυτήν την εγγραφή χρησιμοποιώντας το ακόλουθο query. Υποτίθεται ότι το πρόθεμα βάσεων δεδομένων είναι jos_. Εάν όχι, αντικαθιστούμε το jos_ με το πρόθεμα που έχουμε επιλέξει. Εάν προτιμάμε να εργαστούμε με άμεσες δηλώσεις SQL σε command-line, εισάγουμε το ακόλουθο query στην κονσόλα:

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

Αν προτιμάμε να χρησιμοποιήσουμε ένα web-based database manager όπως είναι η phpMyAdmin, εισάγουμε Restaurant Reviews ως name, option=com_reviews ως link και ως admin_menu_link, Manage Reviews ως admin_menu_alt, com_reviews ως option, και js/ThemeOffice/component.png για admin_menu_img (εικόνα). Αφήνουμε όλα τα άλλα πεδία κενά. Τα πεδία menuid, parent, ordering, και iscore θα είναι εξ' ορισμού 0, ενώ όταν γίνονται enabled θα παίρνουν την τιμή 1.

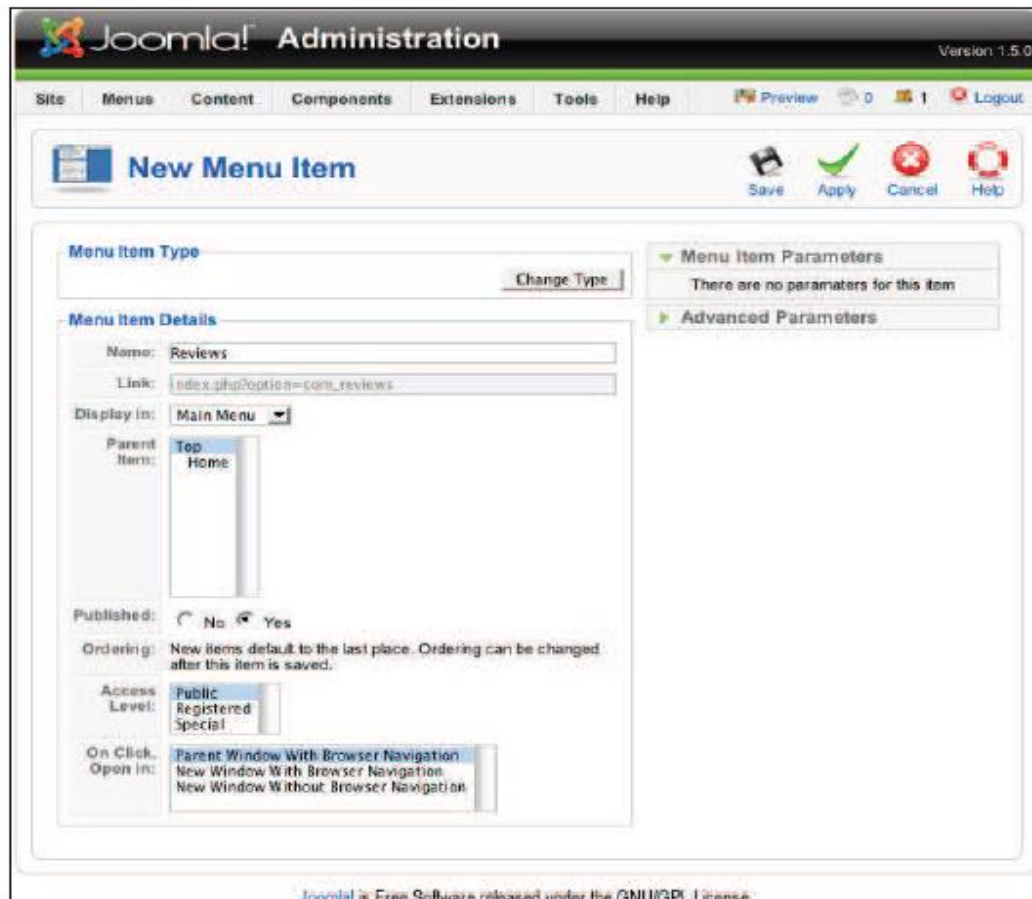
Field	Type	Function	Null	Value
id	int(11)			
name	varchar(150)			Restaurant Reviews
link	varchar(255)			option=com_reviews
menuid	int(11) unsigned			0
parent	int(11) unsigned			0
admin_menu_link	varchar(255)			option=com_reviews
admin_menu_alt	text			Manage Reviews
option	varchar(50)			com_reviews
ordering	int(11)			0
admin_menu_img	varchar(255)			js/ThemeOffice/component.png
iscore	tinyint(4)			0
params	text			
enabled	tinyint(4)			1

Η προσθήκη αυτού του αρχείου δίνει στο σύστημα κάποιες βασικές πληροφορίες για το component. Δηλώνει το όνομα που θέλουμε να χρησιμοποιήσουμε για το component, το οποίο μπορεί να περιέχει διαστήματα και στίξη. Μπορούμε να υποβάλουμε τις συγκεκριμένες συνδέσεις links για να πλοηγηθούμε στο front-end και στο back-end.

Αφότου εισαχθεί επιτυχώς το αρχείο, πηγαίνουμε σε οποιαδήποτε σελίδα στο back-end και το ανανεώνουμε. Όταν κινούμε το ποντίκι πάνω από το μενού των Components πρέπει να βλέπουμε τη νέα επιλογή:



Τώρα που καταχωρήθηκε το component, μπορούμε επίσης να δημιουργήσουμε μια σύνδεση για το frond-end. Menus →Main Menu →New. Από αυτή την οθόνη επιλέγουμε Restaurant Reviews. Εισάγουμε Restaurant Reviews ως name. Παρατηρούμε τώρα την ακόλουθη οθόνη:



The screenshot shows the Joomla! Administration interface for creating a new menu item. The title bar reads 'Joomla! Administration' and 'Version 1.5.0'. The navigation menu includes 'Site', 'Menus', 'Content', 'Components', 'Extensions', 'Tools', and 'Help'. The main content area is titled 'New Menu Item' and contains several sections:

- Menu Item Type:** A dropdown menu with a 'Change Type' button.
- Menu Item Details:**
 - Name:** A text input field containing 'Reviews'.
 - Link:** A text input field containing 'index.php?option=com_reviews'.
 - Display In:** A dropdown menu set to 'Main Menu'.
 - Parent Item:** A tree view showing 'Top' and 'Home'.
 - Published:** Radio buttons for 'No' and 'Yes', with 'Yes' selected.
 - Ordering:** A text area stating 'New items default to the last place. Ordering can be changed after this item is saved.'
 - Access Level:** A dropdown menu with 'Public', 'Registered', and 'Special' options.
 - On Click, Open In:** A dropdown menu with 'Parent Window With Browser Navigation', 'New Window With Browser Navigation', and 'New Window Without Browser Navigation' options.
- Menu Item Parameters:** A section stating 'There are no parameters for this item'.
- Advanced Parameters:** A section with a right-pointing arrow.

At the bottom, there is a footer that reads 'Joomla! is Free Software released under the GNU/GPL License.'

Τώρα αποθηκεύουμε τις εργασίες και πηγαίνουμε στο frond-end και θα παρατηρήσουμε ότι το Reviews-όπως το ονομάσαμε-φαίνεται στη λίστα ως μια επιλογή.

Σε αυτό το σημείο θα πρέπει να γνωρίσουμε ότι από τη στιγμή που θα αρχίσουμε να γράφουμε κώδικα php για το component, ότι έχει να κάνει με το front-end θα εκτελείται μέσω του

http://localhost/joomla/index.php?option=com_reviews

και ο κώδικας για το back-end μέσω του

http://localhost/joomla/administrator/index.php?option=com_reviews.

2.8 Δημιουργώντας Toolbars

Σε όλο το back-end όλα τα τμήματα εφαρμόζουν τις ράβδους εργαλείων με ίδια κουμπιά για αποθήκευση, διαγραφή, διόρθωση και έκδοση. Μπορούμε να χρησιμοποιήσουμε αυτά τα κουμπιά κι εμείς ώστε οι συχνοί χρήστες της περιοχής διαχείρισης να μην βλέπουν διαφορές στη εμφάνιση για λόγους ευκολίας και συνήθειας.

Ξεκινώντας, δημιουργούμε το αρχείο `toolbar.reviews.html.php` μέσα στο φάκελο `administrator/components/com_reviews` και εισάγουμε τον ακόλουθο κώδικα:

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
class TOOLBAR_reviews {

    function _NEW() {
        JToolBarHelper::save();
        JToolBarHelper::apply();
        JToolBarHelper::cancel();
    }

    function _DEFAULT() {
        JToolBarHelper::title( JText::_ ( 'Restaurant Reviews' ),
            'generic.png' );
        JToolBarHelper::publishList();
        JToolBarHelper::unpublishList();
        JToolBarHelper::editList();
        JToolBarHelper::deleteList();
        JToolBarHelper::addNew();
    }
}
?>
```

Τα αρχεία που περιέχουν τους κώδικες οι οποίοι έχουν να κάνουν με τη σχεδίαση και δίνουν ένα output, οργανώνονται σε κλάσεις, όπως εδώ ο κώδικας για το `TOOLBAR_reviews`. Κάθε συνάρτηση αντιπροσωπεύει μια διαφορετική ράβδο εργαλείων. Η κλάση `JToolBarHelper` περιέχει τις λειτουργίες που παράγουν όλη την HTML που είναι απαραίτητη για να χτίσει τις ράβδους εργαλείων. Αν θέλουμε, μπορούμε επίσης να προσθέσουμε το δικό μας κώδικα HTML μέσα από τις συναρτήσεις αυτές. Πρέπει όμως να γνωρίζουμε ότι οι ράβδοι εργαλείων βρίσκονται μέσα στους πίνακες HTML, οπότε πιθανώς να χρειαστεί να προσθέσουμε επικέτες `<td>` και την θέση που επιλέγουμε να εμφανίσουμε το στοιχείο μας στην οθόνη (navigation).

Προσθέτουμε τον ακόλουθο κώδικα μέσα στο toolbar.reviews.php που βρίσκεται 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 που περιέχει το require_once(...) χρησιμοποιεί τη συνάρτηση getPath() της κλάσης JApplicationHelper. Η κλήση της getPath() επιτρέπει να καλέσουμε το αρχείο toolbar.reviews.html.php χωρίς να αναφερθούμε σε κάποιο όνομα component. Αργότερα, ακόμη και αν αλλάξουμε το όνομα Restaurants και τα ονόματα των αρχείων, αυτή η γραμμή θα συνεχίσει να φορτώνει τον κώδικα με τα στοιχεία εμφάνισης του Toolbar χωρίς αλλαγές.

Αφότου το toolbar.reviews.php φορτώσει το αρχείο με την κλάση που είναι υπεύθυνη για την εμφάνιση, θα χρειαστεί να αποφασίσουμε ποιο toolbar θέλουμε να εμφανίζεται. Η μεταβλητή \$task αναγνωρίζεται αυτόματα από τη Joomla και χρησιμοποιείται για να κατευθύνει τη λογική ροή μέσα στο component. Έχοντας προσθέσει τον κώδικα για την εργαλειοθήκη, ανανεώνουμε τον περιηγητή στο back-end και πηγαίνουμε στο μενού Components → Restaurant Reviews και θα πρέπει να δούμε κάτι αντίστοιχο:



Για να δούμε την άλλη εργαλειοθήκη, προσθέτουμε \$task=add στο τέλος της URL στον περιηγητή και πατάμε μετάβαση για να φορτωθεί και να εμφανιστεί το εξής:



Όταν το κουμπί New από την πρώτη εικόνα πατηθεί, θα φέρει στη επιφάνεια τη δεύτερη εργαλειοθήκη. Επειδή μέχρι τώρα εμείς δεν έχουμε καθόλου φόρμες στο back-end, αυτές οι εργαλειοθήκες δε λειτουργούν.

2.9 Κουμπιά εργαλειοθήκης

Η Joomla επιτρέπει να υπερισχύσουμε σε κάθε κουμπί με το δικό μας task και τη δική μας ετικέτα (label), εισάγοντάς τα σαν πρώτη και δεύτερη παράμετρο αντίστοιχα. Τα ακόλουθα κουμπιά είναι διαθέσιμα με την τυποποιημένη διανομή Joomla!:

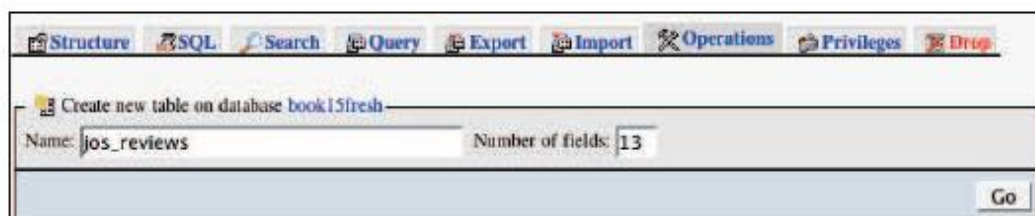
Toolbar code for specific buttons	
<code>mosMenuBar::save();</code> <code>mosMenuBar::saveNew();</code> <code>mosMenuBar::saveedit();</code>	
<code>mosMenuBar::back();</code>	
<code>mosMenuBar::addNew();</code>	
<code>mosMenuBar::editList();</code>	
<code>mosMenuBar::trash();</code> <code>mosMenuBar::deleteList();</code>	
<code>mosMenuBar::publish();</code> <code>mosMenuBar::publishList();</code> <code>mosMenuBar::makeDefault();</code> <code>mosMenuBar::assign();</code>	
<code>mosMenuBar::unpublish();</code> <code>mosMenuBar::unpublishList();</code>	
<code>mosMenuBar::archiveList();</code>	
<code>mosMenuBar::unarchiveList();</code>	
<code>mosMenuBar::editHtml();</code>	
<code>mosMenuBar::editCss();</code>	
<code>mosMenuBar::preview();</code>	
<code>mosMenuBar::media_manager();</code>	
<code>mosMenuBar::apply();</code>	
<code>mosMenuBar::cancel();</code>	

2.10 Ανάπτυξη του Back-end

Πριν δημιουργήσουμε μια διεπαφή για την είσοδο των reviews, πρέπει να δημιουργήσουμε μια θέση στη βάση δεδομένων όπου θα πάνε. Θα αρχίσουμε με έναν πίνακα, όπου μια σειρά (row) θα αντιπροσωπεύει ένα review στο σύστημα. Θεωρούμε ότι το πρόθεμα των βάσεων δεδομένων είναι jos_ και εισάγουμε το ακόλουθο query στην κονσόλα SQL:

```
CREATE TABLE 'jos_reviews'
(
'id' int(11) NOT NULL auto_increment,
'name' varchar(255) NOT NULL,
'address' varchar(255) NOT NULL,
'reservations' varchar(31) NOT NULL,
'quicktake' text NOT NULL,
'review' text NOT NULL,
'notes' text NOT NULL,
'smoking' tinyint(1) unsigned NOT NULL default '0',
'credit_cards' varchar(255) NOT NULL,
'cuisine' varchar(31) NOT NULL,
'avg_dinner_price' tinyint(3) unsigned NOT NULL default '0',
'review_date' datetime NOT NULL,
'published' tinyint(1) unsigned NOT NULL default '0',
PRIMARY KEY ('id')
);
```

Αν χρησιμοποιούμε phpMyAdmin, ανοίγουμε τη ακόλουθη οθόνη και εισάγουμε jos_reviews ως όνομα του πίνακα, που θα έχει 13 πεδία.



Πατάμε Go και εμφανίζεται ένα grid με λεπτομέρειες όπως αυτο:

Field	Type ¹	Length/Values ¹	Collation	Attributes	Null	Default ²
id	INT	11			not null	
name	VARCHAR	255			not null	
address	VARCHAR	255			not null	
reservations	VARCHAR	31			not null	
quicktake	TEXT				not null	
review	TEXT				not null	
notes	TEXT				not null	
smoking	TINYINT	1			not null	0
credit_cards	VARCHAR	255			not null	
cuisine	VARCHAR	31			not null	
avg_dinner_p	TINYINT	3			not null	0
review_date	DATETIME				not null	
published	TINYINT	1			not null	0

Πρέπει οπωσδήποτε να κάνουμε το πεδίο id κύριο κλειδί που αυξάνεται αυτόματα (automatically incremented):

Extra		
auto_increment	☒	☐
	☐	☐
	☐	☐

2.10.1 Δημιουργώντας μια κλάση Table

Η κλάση JTable παρέχει τις λειτουργίες για τη δημιουργία, την ανάγνωση, την ενημέρωση και τη διαγραφή αρχείων από έναν ενιαίο πίνακα στη βάση δεδομένων.

Για να εκμεταλλευτούμε τη JTable, πρέπει να γράψουμε μια επέκταση από αυτή για τα jos_reviews. Στο φάκελο /administrator/components/com_reviews, δημιουργούμε έναν φάκελο με όνομα tables. Σε αυτόν τον φάκελο, δημιουργούμε το αρχείο review.php και πληκτρολογούμε τον ακόλουθο κώδικα:

```
<?php
defined('_JEXEC') or die('Restricted access');
class TableReview extends JTable
{
    var $id = null;
    var $name = null;
    var $address = null;
    var $reservations = null;
    var $quicktake = 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 );
    }
}
?>
```

Όταν επεκτείνουμε (extend) την κλάση JTable, προσθέτουμε όλες τις στήλες του πίνακα της βάσης σαν μεταβλητές και τις ορίζουμε σε null. Η μέθοδος _construct() θα πάρει ένα αντικείμενο της βάσης σαν παράμετρο και θα καλέσει τον αρχικό constructor χρησιμοποιώντας το όνομα του πίνακα της βάσης (όπου το #__ είναι το πρόθεμα), το κύριο κλειδί και το αντικείμενο της βάσης.

Η κλάση TableReview κληρονομεί τις bind(), store(), load(), delete() συναρτήσεις και έτσι μπορούμε να χειριστούμε τις εγγραφές της βάσης χωρίς να χρειάζεται να γράψουμε SQL.

2.10.2 Δημιουργώντας τη φόρμα Review

Αφού έχουμε φτιάξει τον πίνακα σχηματίζουμε μια φόρμα για να προσθέτουμε μέσα τα reviews. Θέλουμε πάλι, όπως και στις εργαλειοθήκες, να διαχωρίσουμε την HTML εμφάνιση από τον λογικό μας κώδικα. Ο php κώδικας θα μπει στο αρχείο admin.reviews.php, ενώ η HTML στο admin.reviews.html.php. Ανοίγουμε το admin.reviews.php και αντικαθιστούμε το περιεχόμενο με τα ακόλουθα:

```
<?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 το require_once(JApplicationHelper::getPath('admin_html')) για να συμπεριλάβουμε το admin.reviews.html.php. Η getPath() συνάρτηση παίρνει συγκεκριμένα strings (όπως είναι τα admin_html, front_html, class) και επιστρέφει το ακριβές μονοπάτι στα ανταποκρινόμενα αρχεία του component.

Η Joomla ταυτίζει με το JPATH_COMPONENT με το ακριβές μονοπάτι για τον κώδικα του back-end. Η editReview() ετοιμάζει τα HTML στοιχεία πριν τα περάσει στην συνάρτηση HTML_reviews::editReview().

Δημιουργούμε το admin.reviews.html.php αρχείο και γράφουμε:

```
<?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">
Name:
</td>
<td>
<input class="text_area" type="text" name="name"
id="name" size="50" maxlength="250"
value="<?php echo $row->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">
Quicktake:
</td>
<td>
<?php
echo $editor->display( 'quicktake', $row->quicktake ,
'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>
<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
}
}
?>

```

Γράφουμε στον περιηγητή:
http://localhost/joomla/administrator/index.php?option=com_reviews&task=add, και
 βλέπουμε την εικόνα:

The screenshot shows the Joomla! administrator interface for adding a new review. The form is titled "Details" and contains the following fields and options:

- Name:** Text input field.
- Address:** Text input field.
- Reservations:** Dropdown menu with "None Taken" selected.
- Quicktake:** Text area with a rich text editor toolbar.
- Review:** Text area with a rich text editor toolbar.
- Notes:** Text input field.
- Smoking:** Radio buttons for "No" (selected) and "Yes".
- Credit Cards:** Text input field.
- Cuisine:** Text input field.
- Average Dinner Price:** Text input field with a dollar sign prefix.
- Review Date:** Text input field with a calendar icon.
- Published:** Radio buttons for "No" (selected) and "Yes".

Αρχικά φέρνουμε μέσα ένα αντικείμενο που αναπαριστώντας τον HTML editor του διαχειριστή και αυτό το κάνουμε με την συνάρτηση JHTML::_('behavior.calendar') για να προσθέσει τη Javascript και το CSS που προϋπάρχει. Αυτό είναι απαραίτητο για το pop up ημερολόγιο του πεδίου Review Date.

```
class HTML_reviews
{
    function editReview( $row, $lists, $option )
    {
        $editor =& JFactory::getEditor();
        JHTML::_('behavior.calendar');
    }
}
```

Η συνάρτηση display() παίρνει τα στοιχεία της φόρμας: name, value, width, height, columns, rows. Τα τελευταία δύο είναι για τις διαστάσεις της <textarea> στην περίπτωση που ο διαχειριστής δε θέλει να χρησιμοποιήσει ημερολόγιο.

```
<td>
<?php
echo $editor->display( 'quicktake', $row->quicktake ,
    '100%', '150', '40', '5' ) ;
?>
</td>
```


2.10.3 Επεξεργασία δεδομένων

Όταν τα δεδομένα στη φόρμα έχουν συμπληρωθεί, ο διαχειριστής πατάει το κουμπί save της αποθήκευσης και χρειάζεται να αποθηκευτούν αυτές οι πληροφορίες στη βάση. Αρχικά δημιουργούμε το saveReview() μέσα στο 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->quicktake = JRequest::getVar( 'quicktake', '', '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');
}
```

Αρχικά τραβάμε την global μεταβλητή \$mainframe και την τρέχουσα σύνδεση της βάσης δεδομένων. Το αντικείμενο \$mainframe έχει πολλές συναρτήσεις που μπορούν να χρησιμοποιηθούν για να ελεγχθούν οι μεταβλητές και κεφαλίδες αυτής της συνεδρίας. Θέσαμε επίσης τη \$row σαν αναφορά για την κλάση TableReview, όπου το όνομα της κλάσης έχει συναρμολογηθεί από τις παραμέτρους review και table. Στη συνέχεια καλούμε την συνάρτηση της \$row που λέγεται bind(), ώστε να φορτώσει όλες τις μεταβλητές από τη φόρμα. Η bind() παίρνει μια σειρά πίνακα ως παράμετρο και προσπαθεί να ταιριάξει όλα τα στοιχεία της στις μεταβλητές μελών του αντικειμένου. Αν αποτύχει η συνάρτηση, εμφανίζουμε ένα JavaScript alert και μεταφέρουμε το χρήστη στην προηγούμενη οθόνη με τη συνάρτηση history.

Μετά το bind(), μπορούμε να χειριστούμε τις μεταβλητές μελών \$row άμεσα. Δεδομένου ότι τα πεδία quicktake και review δέχονται το περιεχόμενο HTML, χρειάζονται ειδικό χειρισμό, αφού η bind() αφαιρεί όλη την HTML. Για να ξεπεραστεί αυτό χρησιμοποιούμε την συνάρτηση getVar() της JRequest. Έτσι περνάμε το όνομα της φόρμας, την εξ'ορισμού τιμή, τον πίνακα από τον

οποίο θέλουμε να πάρουμε δεδομένα, το αναμενόμενο φορμάτ, και την JREQUEST_ALLOWRAW flag αντίστοιχα.

Έχουμε την δυνατότητα να θέτουμε κάποια προκαθορισμένα δεδομένα, όπως κάνουμε εδώ για την τρέχουσα ημερομηνία. Ρυθμίσαμε να συμπληρώνεται το πεδίο αυτόματα, σε περίπτωση που δεν είχε επιλεγεί.

Τέλος, καλούμε τη συνάρτηση store() που παίρνει όλες τις μεταβλητές μελών και τις μετατρέπει σε δηλώσεις UPDATE ή INSERT, ανάλογα με την τιμή του id. Δεδομένου ότι δημιουργούμε αυτό το αρχείο για πρώτη φορά, δεν έχει τιμή το id, άρα θα γίνει INSERT και όχι UPDATE. Αν προκύψει κάποιο λάθος SQL, επιστρέφουμε στην προηγούμενη οθόνη. Αν το SQL είναι επιτυχές, χρησιμοποιούμε τη συνάρτηση redirect() από το \$mainframe για να στείλουμε το χρήστη πίσω στην βασική οθόνη του component με ένα μήνυμα επιβεβαίωσης.

Ως τώρα, το switch() στο αρχείο admin.reviews.php επεξεργαζόταν μόνο το add task. Τώρα που έχουμε δημιουργήσει μια φόρμα και μια συνάρτηση για να διαχειρίζεται την αποθήκευση, μπορούμε να προσθέσουμε ένα case για να αποθηκεύουμε τα δεδομένα.

```
switch($task)
{
    case 'add':
        editReview( $option );
        break;

    case 'save':
        saveReview( $option );
        break;
}
```

2.10.4 Δημιουργία λίστας με τα δεδομένα της βάσης

Επειδή ο διαχειριστής δεν έχει πρόσβαση στη phpMyAdmin, χρειάζεται να φτιάξουμε μια οθόνη που να συγκεντρώνει τα reviews της βάσης δεδομένων. Προσθέτουμε στο admin.reviews.php τον κώδικα:

```
function showReviews( $option )
{
    $db =& JFactory::getDBO();
    $query = "SELECT * FROM #__reviews";
    $db->setQuery( $query );
    $rows = $db->loadObjectList();
    if ($db->getErrorNum()) {
        echo $db->stderr();
        return false;
    }
    HTML_reviews::showReviews( $option, $rows );
}
```

Αυτή η συνάρτηση φορτώνει τα δεδομένα που είναι προς εμφάνιση, άρα παίρνουμε μια αναφορά της τρέχουσας σύνδεσης με τη βάση και ύστερα καλούμε την `setQuery()` η οποία δέχεται μέσα της το string του query και το κρατά για μελλοντική χρήση, χωρίς να το εκτελέσει αμέσως. Όταν όμως καλούμε το `loadObjectList()`, το προηγούμενο query εκτελείται και οι γραμμές του αποτελέσματός του αποθηκεύονται σε ένα πίνακα τύπου array σαν αντικείμενα. Αν προκύψει σφάλμα, σταματάμε το component. Αν δεν υπάρχει σφάλμα, περνάμε τον πίνακα των αποτελεσμάτων στην παρακάτω συνάρτηση που θα προστεθεί 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">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
    $k = 0;
    for ($i=0, $n=count( $rows ); $i < $n; $i++)
    {
        $row = &$rows[$i];
        $checked = JHTML::_('grid.id', $i, $row->id );
        $published = JHTML::_('grid.published', $row, $i );
        ?>
        <tr class="<?php echo "row$k"; ?>">
        <td>
        <?php echo $checked; ?>
        </td>
        <td>
        <?php echo $row->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
$х = 1 - $k;
}
?>
</table>
<input type="hidden" name="option"
value="<?php echo $option;?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="checkboxchecked" value="0" />
</form>
<?php
}

```

Αυτή η συνάρτηση αρχίζει με τον καθορισμό μιας φόρμας που δείχνει στο index.php, με το όνομα adminForm (για τις αναφορές JavaScript). Ένας πίνακας με την κλάση adminlist αρχίζει και οι επικεφαλίδες προστίθενται. Όλες οι επικεφαλίδες είναι οι τυπικές, εκτός από την πρώτη που ενεργεί σαν ένα checkbox ελέγχου, που επιλέγει αυτόματα όλες τις εγγραφές στην οθόνη.

Τώρα ξεκινά μια επανάληψη στις γραμμές. Οι μεταβλητή \$i είναι αρχικά 0 και η \$n ίση με τον αριθμό των γραμμών. Η επανάληψη εκτελείται όσο υπάρχουν ακόμη διαθέσιμες γραμμές προς εμφάνιση. Μέσα στην επανάληψη έχουμε μια αναφορά στην τρέχουσα γραμμή για να μπορούμε να εμφανίζουμε τα περιεχόμενα. Αλλάζουμε την τιμή της μεταβλητής \$k μια σε 0 και μια σε 1, ώστε να εναλλασσόμαστε μεταξύ δύο διαφορετικών CSS κλάσεων με διαφορετικές background properties.

Αρκετές από τις μεταβλητές παίρνουν μορφή εμφάνισης αμέσως, όμως κάποιες θέλουν ειδική μεταχείριση. Χρησιμοποιώντας το JHTML::_('grid.id') μπορούμε να πάρουμε τον HTML κώδικα ενός checkbox που θα αναγνωρίζεται από το back-end Javascript. Η συνάρτηση JHTML::_('grid.published') παράγει μία εικόνα κουμπιού βασισμένη στην τιμή της published μεταβλητής της γραμμής. Όταν αυτή είναι 1, παίρνουμε ένα "check" ως εικόνα, ενώ όταν είναι 0 εμφανίζεται ένα "x".

Ανανεώνουμε και πάλι τον κώδικα του switch() στο admin.reviews.php:

```

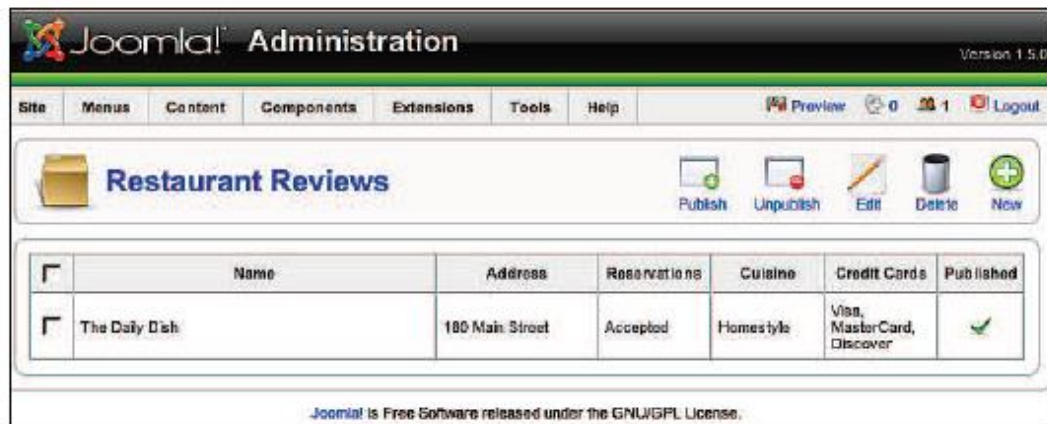
switch($task)
{
    case 'add':
        editReview( $option );
        break;

    case 'save':
        saveReview( $option );
        break;

    default:
        showReviews( $option );
        break;
}

```

Μόλις γράψουμε στον περιηγητή
http://localhost/joomla/administrator/index.php?option=com_reviews,
εμφανίζεται η οθόνη:



The screenshot shows the Joomla! Administration interface for the 'Restaurant Reviews' component. The top navigation bar includes links for Site, Menus, Content, Components, Extensions, Tools, and Help. The main content area features a table with columns for Name, Address, Reservations, Cuisine, Credit Cards, and Published. A single entry, 'The Daily Dish', is listed with an address of '180 Main Street', 'Accepted' reservations, 'Homestyle' cuisine, and 'Visa, MasterCard, Discover' credit cards. The 'Published' column shows a green checkmark. The interface also includes a sidebar with icons for Publish, Unpublish, Edit, Delete, and New. The footer states 'Joomla! is Free Software released under the GNU/GPL License.'

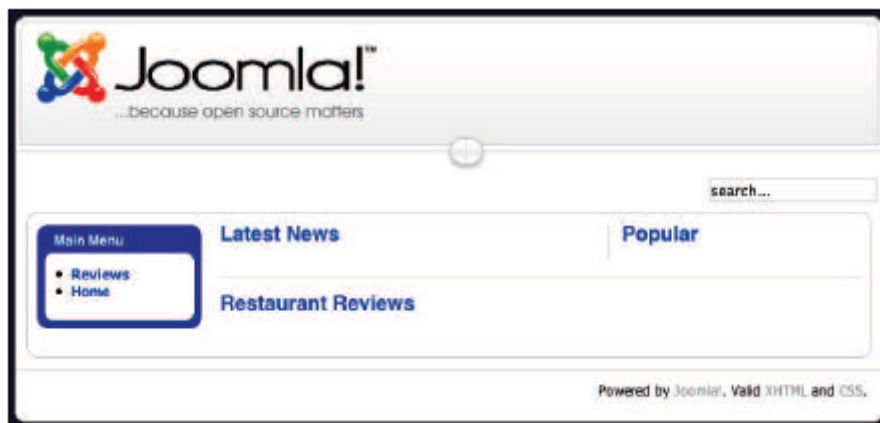
	Name	Address	Reservations	Cuisine	Credit Cards	Published
☐	The Daily Dish	180 Main Street	Accepted	Homestyle	Visa, MasterCard, Discover	☒

2.11 Ανάπτυξη του Front-end

Έχοντας γράψει κάποια δεδομένα, reviews, στο back-end, τώρα οι χρήστες θέλουν να δουν πώς αυτά εμφανίζονται για να τα βλέπουν οι επισκέπτες του ιστοτόπου στο front-end.

2.11.1 Συγκέντρωση των reviews

Μέχρι τώρα όταν πληκτρολογήσαμε http://localhost/joomla/index.php?option=com_reviews, μεταφερόμασταν στην εικόνα:



Τώρα θα γεμίσουμε αυτή την εικόνα με μια λίστα που θα περιέχει συνδέσμους που θα δείχνουν σε ξεχωριστές reviews το καθένα, τις οποίες έχουμε προσθέσει στη βάση, ώστε όταν οι επισκέπτες φορτώνουν τον ιστότοπο να μπορούν να περιηγηθούν στα reviews.

Πηγαίνουμε στο φάκελο `/components/com_reviews` στο αρχείο `reviews.php` και εισάγουμε τον κώδικα:

```
jimport('joomla.application.helper');
require_once( JApplicationHelper::getPath( 'html' ) );
JTable::addIncludePath(JPATH_ADMINISTRATOR.DS.
    'components'.DS.$option.DS.'tables');

switch($task)
{
default:
showPublishedReviews($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);
}

```

Περνάμε TO
JPATH_ADMINISTRATOR.DS.'components'.DS.\$option.DS.'tables' στο
Jtable::addIncludePath() για να τραβήξουμε την κλάση table που είχαμε
γράψει στο back-end. Η συνάρτηση switch() λειτουργεί με την προεπιλεγμένη
περίπτωση case, η οποία καλεί τη συνάρτηση για να εμφανίσει όλες τις
κοινοποιημένες (published) reviews. Το query σε αυτή τη συνάρτηση
διαβεβαιώνει το ότι θα εμφανιστούν μόνο οι “published reviews” και αυτές θα
είναι ταξινομημένες με αντίστροφη χρονολογική με βάση την ημερομηνία που
έγινε το review (review date).

Σε αυτό το σημείο πρέπει να προσθέσουμε την κλάση HTML_reviews
class για το frond-end. Στο φάκελο /components/com_reviews δημιουργούμε
το αρχείο reviews.html.php:

```

<?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
'<tr>
<td>
<a href="' . $link . '"> . $row->name . '</a>
</td>
</tr>';
}
?></table><?php
}
}
?>

```

Η συνάρτηση `showReviews()` παίρνει ένα σετ από γραμμές της βάσης και το όνομα του component. Σε κάθε επανάληψη η συνάρτηση προσθέτει ένα σύνδεσμο link για κάθε γραμμή.



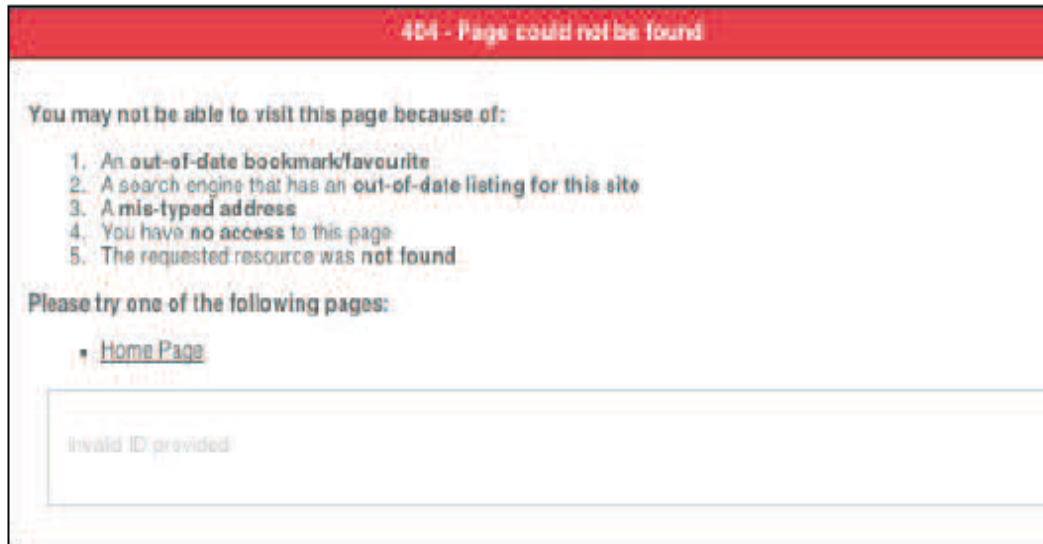
2.11.2 Εμφάνιση των reviews

Αν κάνουμε κλικ σε κάποιο από τους συνδέσμους τώρα, θα δούμε απλώς τη ίδια οθόνη αφού δεν έχουμε ακόμη φτιάξει τον κώδικα που θα χειρίζεται το view task. Γι' αυτό προσθέτουμε στο `reviews.php` τον κώδικα:

```
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);
}
```

Πρώτα παίρνουμε το `id` από το request χρησιμοποιώντας τη `getVar()`, η οποία ελέγχει τη μεταβλητή ως προς διαφορετικούς τύπους. Αν η τιμή για το `id` λείπει ή δεν είναι συμβατή, τότε λαμβάνεται ως τιμή το 0 που δίνεται ως δεύτερη επιλογή παραμέτρου.

Στη συνέχεια παίρνουμε μια αναφορά από την κλάση table του back-end. Αφού φορτωθεί η γραμμή με το εκάστοτε id, γίνεται έλεγχος για να πιστοποιηθεί ότι η επιλεγμένη review είναι “published”. Αν δεν είναι, χρησιμοποιούμε την συνάρτηση `raiseError()` του `JError` για να εμφανίσει το μήνυμα «404 - Page could not be found».



Τώρα προσθέτουμε στο `switch()` τον παρακάτω κώδικα:

```
switch($task)
{
    case 'view':
        viewReview($option);
        break;
    default:
        showPublishedReviews($option);
        break;
}
```

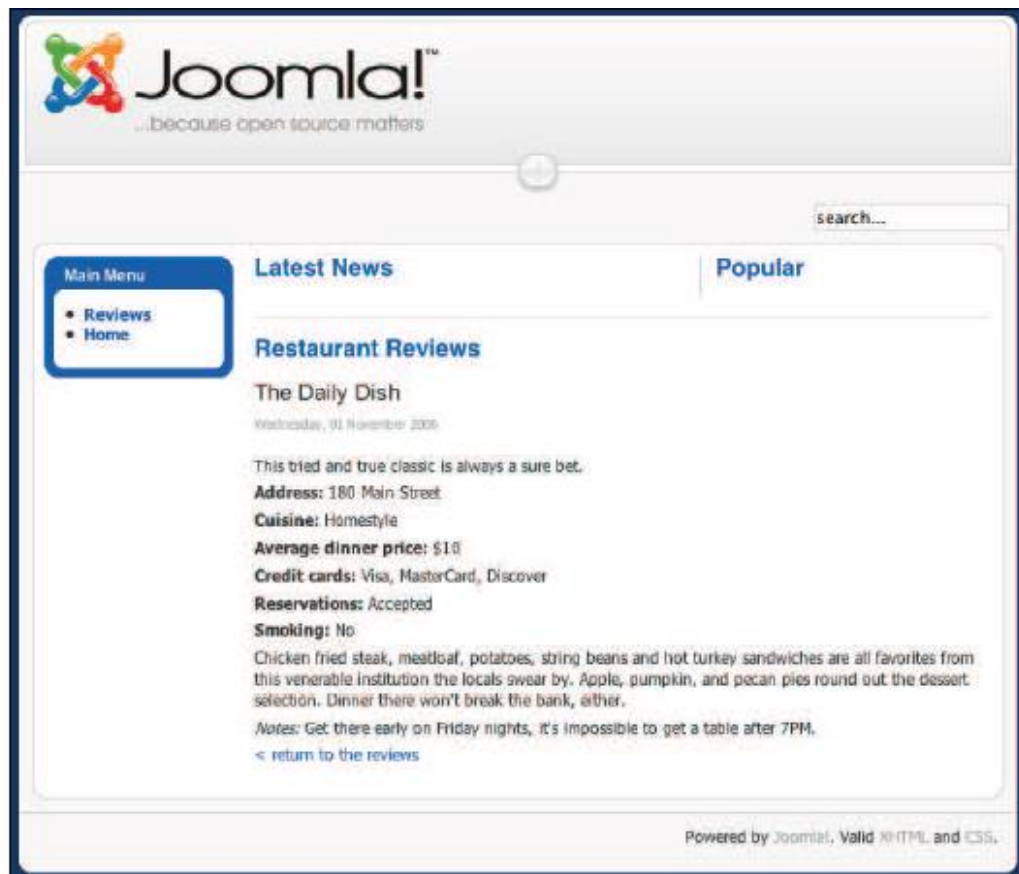
Επίσης πρέπει να φτιάξουμε μια συνάρτηση για εμφάνιση. Προσθέτουμε την συνάρτηση showReview() στο HTML_reviews στο φάκελο reviews.html.php:

```
function showReview($row, $option)
{
    ?>
    <p class="contentheading"><?php echo $row->name; ?></p>

    <p class="createdate"><?php echo JHTML::Date
        ($row->review_date); ?></p>
    <p><?php echo $row->quicktake; ?></p>
    <p><strong>Address:</strong> <?php echo $row->address; ?></p>
    <p><strong>Cuisine:</strong> <?php echo $row->cuisine; ?></p>
    <p><strong>Average dinner price:</strong> $<?php echo
        $row->avg_dinner_price; ?></p>
    <p><strong>Credit cards:</strong> <?php echo
        $row->credit_cards; ?></p>
    <p><strong>Reservations:</strong> <?php echo
        $row->reservations; ?></p>
    <p><strong>Smoking:</strong> <?php
    if($row->smoking == 0)
    {
        echo "No";
    }
    else
    {
        echo "Yes";
    }
    ?></p>
    <p><?php echo $row->review; ?></p>
    <p><em>Notes:</em> <?php echo $row->notes; ?></p>
    <?php $link = 'index.php?option=' . $option ; ?>
    <a href="<?php echo $link; ?>">&lt; return to the reviews</a>
    <?php
}
```

Η συνάρτηση showReview() παίρνει μια γραμμή της βάσης σαν αντικείμενο και το όνομα του component, σαν παραμέτρους. Οι περισσότερες στήλες της γραμμής εμφανίζονται με HTML. Η στήλη “smoking” θα επιστρέφει «Ναι» ή «Όχι». Η κλήση της JHTML::Date() μετασχηματίζει το timestamp της βάσης στην τοπική μορφή που προτιμάται κάθε φορά. Τέλος πηγαίνουμε πάλι πίσω στη λίστα με τα reviews.

Τώρα επιλέγουμε να κάνουμε κλικ πάλι σε μια επιλογή ενός review και βλέπουμε μια ολοκληρωμένη εικόνα:



ΑΝΑΦΟΡΕΣ

[1] Content Management Systems:

<http://searchsoa.techtarget.com/definition/content-management-system>.
Ιούνιος 2011.

[2] Πλεονεκτήματα των CMSs:

<http://www.cpdservices.com/53.0.html>. Ιούνιος 2011.

[3] Μειονεκτήματα των CMSs:

<http://www.passioncomputing.com.au/Dynamic-Websites/CMS-disadvantages.aspx> . Ιούνιος 2011.

[4] Μειονεκτήματα των CMSs:

<http://www.drostdesigns.com/the-disadvantage-of-a-content-management-system/>. Ιούνιος 2011.

[5] Σύγκριση:

http://www.siteground.com/best_cms_tools.htm. Ιούνιος 2011.

[6] Σύγκριση:

<http://www.digitalpacific.com.au/blog/drupal-joomla-and-wordpress-which-cms-to-choose/>. Ιούνιος 2011.

[7] Διάγραμμα σύγκρισης:

<http://www.mybestratedwebhosting.com/best-web-hosting-tips/cms-comparison-chart.html>. Ιούνιος 2011.

[8] Joseph LeBlanc, “Learning Joomla! 1.5 Extension Development”, Chpt. 3, Chpt. 4. June 2007.

[9] Steve White and Andy Wallace, “Joomla! User Manual”. October 2006.

[10] Εγκατάσταση: User manual for Joomla! 1.5,

<http://www.joomlawired.com/support/user-manual-for-joomla-15>. Αύγουστος 2011.

