



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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάλυση, σχεδιασμό και ανάπτυξη συστήματος
διαχείρισης περιεχομένου διαδικτυακών τόπων.**



Cactus

Του φοιτητή
Σταμάτη Ανέστη
Αρ. Μητρώου: 09/3441

Επιβλέπων καθηγητής
Δρ. Χατζημίσιος

Θεσσαλονίκη 2015

Ανέστης Σταμάτης

Περιεχόμενα

Περίληψη.....	4
Κεφάλαιο 1. Αρχιτεκτονική MVC.....	5
Τι είναι η αρχιτεκτονική MVC;.....	5
Model.....	5
View.....	6
Controller.....	6
Γιατί να χρησιμοποιούμε MVC αρχιτεκτονικές;.....	7
Διαχωρισμός Προβλημάτων (Separation of Concerns).	7
Επεκτασιμότητα.	7
Ελεγχιμότητα (Testability).	7
“Καθαρά” URLs.....	8
Περιγραφή cakePHP.	9
Γιατί cakePHP.....	10
Κεφάλαιο 2. Δομή και κατασκευή ιστοσελίδων.	12
Στατικές και δυναμικές ιστοσελίδες.....	12
CMS.....	12
Συστήματα CMS Ανταγωνισμός.....	18
Το Joomla.....	18
Τι είναι το WordPress.....	21
Drupal.....	25
Τι είναι open source.....	32
Κεφάλαιο 3. Cactus CMS.	34
Πλεονεκτήματα.....	34
Μειονεκτήματα.....	35
Περιπτώσεις Χρήσης.....	36
Περίπτωση Χρήσης: Ο χρήστης τροποποιεί το περιεχόμενο της Ιστοσελίδας.....	36
Περίπτωση Χρήσης: Ο χρήστης επεκτείνει το περιεχόμενο της Ιστοσελίδας.	37
Μοντέλο εφαρμογής (Model).....	38
User.....	38
Post.....	39
Category.....	40
Gallery Category.....	41
Gallery.....	43
Page.....	44
Menu.....	45
Controller Εφαρμογής - Λειτουργικότητα.....	47

Οι Controller του Cactus.....	48
Controller του Posts:.....	48
Controller του Users:.....	49
Controller του Pagem.....	50
Menu - Mcategories Controller.....	51
CategoriesController.....	51
GalleryController.....	52
GCategoriesController.....	52
Component FileUploadComponent.....	54
Δομή αρχείων συστήματος Cactus CMS.....	55
Κεφάλαιο 4. Πρόσθετες λειτουργίες του συστήματος.....	58
Πρόσθετο ηλεκτρονικού καταστήματος.....	58
Πρόσθετο ηλεκτρονικών κρατήσεων ξενοδοχείων.....	58
Interface - διεπαφή εφαρμογής.....	59
Responsive Layout.....	61
Κεφάλαιο 5. Συμπεράσματα.....	63
Βιβλιογραφία.....	64

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1: Συνέδριο Drupal.....	30
Εικόνα 2 Διάγραμμα οντοτήτων.....	38
Εικόνα 3 Κώδικας Μοντέλου User.....	46
Εικόνα 4 Κώδικας Controller Posts.....	53
Εικόνα 5 Δομή αρχείων.....	57
Εικόνα 6 Φόρμα σύνδεσης στο σύστημα.....	60
Εικόνα 7 Βασική φόρμα εφαρμογής.....	61
Εικόνα 8 Διεπαφή εισαγωγής δεδομένων.....	62

Πρόλογος.

Η εργασία αυτή έχει ως στόχο να αναδείξει την υφιστάμενη κατάσταση στο τρόπο κατασκευής ιστοσελίδων τα πλεονεκτήματα αλλά και τις αδυναμίες. Σε αυτήν την εργασία περιγράφεται το σύστημα διαχείρισης περιεχομένου Cactus CMS. Το Cactus CMS έρχεται να αναθεωρήσει αρκετά πάνω στον τρόπο που κατασκευάζουμε τις ιστοσελίδες. Η βασική καινοτομία που εισάγει είναι η δυνατότητα επέκτασης της εκάστοτε ιστοσελίδας όχι μόνο προς το περιεχόμενο αλλά και προς την λειτουργικότητα. Μία δεύτερη καινοτομία είναι η δυνατότητα που παρέχεται στους προγραμματιστές να λειτουργούν σε ομάδα και να τροποποιούν τα αρχεία τις εκάστοτε ιστοσελίδας από κοινού. Ο τρόπος υλοποίησης και λειτουργίας του συστήματος έχει σίγουρα ακαδημαϊκή αξία καθώς έχει να μας διδάξει πολλά για τον σύγχρονο προγραμματισμό διαδικτυακών εφαρμογών και συστημάτων.

Περίληψη.

Η ανάγκη κατασκευής και η διαχείριση δυναμικών ιστοσελίδων οδηγούν κάθε φορά στην κατασκευή έξυπνων συστημάτων που καλούνται να απλοποιήσουν τις διαδικασίες και να αναπτύξουν τις ιστοσελίδες ακόμα περισσότερο. Η εργασία αυτή αρχικά εισάγει όλες τις βασικές έννοιες που απαιτούνται για να καταλάβει κάποιος την ουσία του προγραμματισμού μιας σύγχρονης ιστοσελίδας. Στην συνέχεια η εργασία φέρνει σε σύγκριση τα 3 βασικά συστήματα διαχείρισης περιεχομένου που έχουμε αυτήν την στιγμή στα χέρια μας. Τέλος παρουσιάζετε εκτενώς η δομή ενός νέου συστήματος διαχείρισης περιεχομένου που κατασκευαστικέ ιδικά για τις ανάγκες αυτής της εργασίας.

Κεφάλαιο 1. Αρχιτεκτονική MVC.

Τι είναι η αρχιτεκτονική MVC;

Η ονομασία MVC αναπαριστά τα αρχικά για τις λέξεις Model View Controller και ξεκίνησε την δεκαετία του 1980 σε μια αντικειμενοστραφής γλώσσα την SmallTalk. Η αρχιτεκτονική MVC είναι ένα μοντέλο αρχιτεκτονικής λογισμικού οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Τα τρία αυτά μέρη είναι διακριτά και το καθένα έχει τον δικό του ρόλο και σημασία.

Model

Το model ή Μοντέλο αναπαριστά την γνώση. Το model μπορεί να είναι ένα αντικείμενο ή μπορεί να είναι και μια δομή από αντικείμενα. Θα πρέπει να είναι σχέση ένα προς ένα από το μοντέλο και τα μέρη του στην πρώτη περίπτωση ενώ στην δεύτερη να αντιπροσωπεύει τον κόσμο με την αντίληψη του ιδιοκτήτη του μοντέλου.

Οι κόμβοι ενός μοντέλου θα πρέπει να αντιπροσωπεύουν ένα αναγνωρίσιμο μέρος του προβλήματος. Επίσης οι κόμβοι ενός μοντέλου θα πρέπει όλοι να αφορούν στο ίδιο πρόβλημα ειδάλλως θα υπάρχει σύγχυση και θα θεωρείτε κακή υλοποίηση και σχεδίαση.

Στην ουσία το Μοντέλο παίζει τον ρόλο του ενδιάμεσου ανάμεσα στην αποθήκη δεδομένων και στην λογική της εφαρμογής μας. Συνήθως έχουμε να κάνουμε με την αποτύπωση προγραμματιστικά κάποιο πίνακα, κάποιας βάσης δεδομένων.

View

Ένα View ή Διεπαφή Χρήστη είναι μια (οπτική) αναπαράσταση του μοντέλου. Θα τονίσει συνήθως ορισμένα χαρακτηριστικά του μοντέλου και θα υποβαθμίσει κάποια άλλα. Είναι έτσι ενεργεί ως ένα φίλτρο παρουσίασης. Ένα View είναι συνδεδεμένο με το μοντέλο (ή ενός μέρος του μοντέλου) και παίρνει τα στοιχεία που είναι απαραίτητα για την παρουσίαση του με τις κατάλληλες ερωτήσεις. Μπορεί επίσης να ενημερώσει το μοντέλο με την αποστολή των κατάλληλων μηνυμάτων. Όλες αυτές οι ερωτήσεις και τα μηνύματα πρέπει να είναι στην σημασιολογία του μοντέλου και όπως είναι λογικό ένα View θα πρέπει να γνωρίζει τη σημασιολογία των χαρακτηριστικών του μοντέλου που αντιπροσωπεύει για να μπορεί να επικοινωνεί σωστά με αυτό.

Ουσιαστικά η διεπαφή χρήστη αποτελείτε από όλα εκείνα τα γραφικά στοιχεία που βοηθάνε τον εκάστοτε χρήστη να αλληλεπιδράσει με την εφαρμογή και τα στοιχεία ή δεδομένα. Είναι ωφέλιμο να πούμε ότι πολλές φορές για τον τελικό χρήστη η διεπαφή μπορεί να καθορίσει και το γενικότερο αποτέλεσμα της εφαρμογής μας, καθώς είναι πολύ σημαντικό ο τρόπος που θα παρουσιάζονται και θα διαχειρίζονται τα δεδομένα.

Controller.

Ένας Controller είναι ο συνδετικός κρίκος μεταξύ ενός χρήστη και του συστήματος. Παρέχει στο χρήστη είσοδο μεριμνώντας για τα σχετικά Views που πρέπει να του παρουσιαστούν στις κατάλληλες θέσεις στην οθόνη. Αυτό παρέχει τα μέσα για την έξοδο του χρήστη από το View με μενού ή άλλα μέσα για να δώσει εντολές και δεδομένα. Ο Controller λαμβάνει τέτοιες εξόδους χρήστη, μεταφράζει στα κατάλληλα μηνύματα και τα περνάει στα Views.

Στον Controller ουσιαστικά βρίσκεται η λογική του συστήματος μας, εκεί υλοποιούνται όλες οι διαδικασίες και εν συνεχεία παρουσιάζονται στον χρήστη. Ο Controller δεν έχει άμεση πρόσβαση στα δεδομένα αλλά έχει άμεση πρόσβαση στο Μοντέλο της εφαρμογής. Το γεγονός αυτό μας δίνει την δυνατότητα να σχεδιάσουμε αλγόριθμους που είναι ανεξάρτητοι από τις λεπτομέρειες των δεδομένων.

Γιατί να χρησιμοποιούμε MVC αρχιτεκτονικές;

Διαχωρισμός Προβλημάτων (Separation of Concerns).

Αυτό είναι και το πιο βασικό πλεονέκτημα του MVC. Ουσιαστικά δημιουργείται μία εφαρμογή η οποία έχει τρία επίπεδα, το επίπεδο των Models, το επίπεδο των Controllers και το επίπεδο των Views -που θα αναλυθούν παρακάτω- και το κάθε επίπεδο επιτελεί ξεχωριστό έργο και ταυτόχρονα συνεργάζεται με τα άλλα επίπεδα. Μία σωστή MVC εφαρμογή είναι εκείνη που τα τρία επίπεδα είναι ξεκάθαρα καθορισμένα και δεν συμπλέκονται. Για παράδειγμα είναι λάθος στο επίπεδο των View να υπάρχει κώδικας που “μιλάει” με την βάση δεδομένων και “τραβάει” δεδομένα.

Επεκτασιμότητα.

Το δεύτερο πλεονέκτημα της MVC αρχιτεκτονικής είναι πολύ εξίσου σημαντικό. “Επεκτασιμότητα” είναι η δυνατότητα που διαθέτει μία εφαρμογή, κατά την οποία μπορούμε μελλοντικά να προσθέσουμε λειτουργίες σε αυτή ή να αλλάξουμε κάποιες από τις ήδη υπάρχουσες λειτουργίες και να έχουμε άλλα αποτελέσματα. Για να το δούμε εντελώς απλά και κατανοητά, η πλατφόρμα WordPress είναι επεκτάσιμη με τη χρήση των διάφορων plugins διότι προσθέτουμε στις ήδη υπάρχουσες λειτουργίες και άλλες λειτουργίες. Τα προγράμματα που είναι φτιαγμένα με MVC αρχιτεκτονική έχουν βασικό χαρακτηριστικό ότι είναι επεκτάσιμα. Είναι ίσως το σημαντικότερο για μία εφαρμογή να μπορεί να επεκτείνεται εύκολα και χωρίς να δημιουργεί η εκάστοτε αλλαγή προβλήματα σε άλλες συνιστώσες από την δεδομένη που αλλάζει ή επεκτείνεται.

Ελεξιμότητα (Testability).

Αυτό είναι ένα πολύ κρίσιμο χαρακτηριστικό. Οι MVC εφαρμογές έχουν την δυνατότητα να είναι ελέγξιμες και με τον τρόπο αυτό συντηρούνται πιο εύκολα. Ας δώσουμε ένα απλό παράδειγμα. Έστω ότι έχουμε μία εφαρμογή η οποία διαθέτει μία λειτουργία Login - σύνδεσης, δηλαδή ζητά από τον χρήστη να πληκτρολογήσει κάποια στοιχεία σε μία φόρμα και εν συνέχεια αν

ταυτοποιηθεί, τον εισάγει μέσα στο σύστημα. Αυτή τη λειτουργία την ελέγχει κάποιος Login Controller ο οποίος περιέχει κώδικα που διαχειρίζεται τα δεδομένα αυτά που εισήχθησαν από τον χρήστη. Αυτός ο Controller θεωρείται μία “μονάδα” ή αλλιώς Unit. Στα MVC Frameworks μπορούμε με πολλή ευκολία να γράψουμε έναν απλό κώδικα - tests με τον οποίο τεστάρουμε αυτόν τον Controller αλλά και κάθε μία από τις λειτουργίες του. Παίρνουμε τα αποτελέσματα και βλέπουμε αν η συγκεκριμένη μονάδα της εφαρμογής μας λειτουργεί σωστά.

Σε online συστήματα που πολλές φορές απευθύνονται σε χιλιάδες χρήστες είναι πολύ κρίσιμο να γνωρίζουμε πριν την διανομή του κώδικά μας εάν είναι σωστά υλοποιημένα όλα τα υποσυστήματα μας. Στις διαδικτυακές εφαρμογές ο έλεγχος είναι μέγιστης σημασίας.

“Καθαρά” URLs.

Τα περισσότερα MVC Frameworks για web applications – διαδικτυακές εφαρμογές δίνουν τη δυνατότητα να έχουμε “καθαρά” urls. Καθαρά URLs σημαίνει ότι οι διευθύνσεις που θα εμφανίζονται στον Browser του επισκέπτη καθώς αυτός θα περιηγείται στην εφαρμογής μας, θα βγάζουν νόημα και θα είναι γραμμένα σε φυσική γλώσσα. Το γεγονός αυτό θα εξηγήσουμε σε επόμενο κεφάλαιο γιατί είναι ωφέλιμο.

Στο σημείο αυτό θα προχωρήσουμε σε μία ανάλυση του MVC Framework που χρησιμοποιήθηκε για τον σκοπό αυτής της εργασίας. Αυτό είναι το CakePHP ένα από τα πιο διαδεδομένα PHP Frameworks και ταυτόχρονα απλό και εύχρηστο.

Περιγραφή cakePHP.

Το CakePHP είναι ένα πλαίσιο web εφαρμογής ανοικτού κώδικα. Ακολουθεί την προσέγγιση Model View Controller (MVC) και είναι γραμμένο σε PHP, διαμορφώθηκε μετά από τις έννοιες της Ruby on Rails, και διανέμεται υπό την άδεια MIT.

Το CakePHP χρησιμοποιεί γνωστές έννοιες της μηχανικής λογισμικού και τα πρότυπα σχεδιασμού λογισμικού, όπως Convention over configuration, Model View Controller, Active Record, Association Data Mapping, και Front Controller.

Το CakePHP ξεκίνησε τον Απρίλιο του 2005, όταν ο Πολωνός προγραμματιστής Michal Tatarynowicz έγραψε μια ελάχιστη έκδοση του Rapid Application Framework σε PHP, και του έδωσε όνομα cake. Έχει εκδώσει το Framework κάτω από την άδεια MIT, και το έδωσε στην online κοινότητα των προγραμματιστών. Τον Δεκέμβριο του 2005, ο Λ Masters και ο GJ Woodworth ίδρυσαν το Cake Software Foundation για την προώθηση της ανάπτυξης εφαρμογών που σχετίζονται με το CakePHP. Έκδοση 1.0 που κυκλοφόρησε τον Μάιο του 2006.

Μία από τις εμπνεύσεις του έργου ήταν η Ruby on Rails, χρησιμοποιώντας πολλές από τις ιδέες της. Η κοινότητα έχει αυξηθεί από τότε και γέννησε αρκετά υπο-έργα και επέκτεινε αρκετά τις λειτουργίες και δυνατότητες.

Τον Οκτώβριο του 2009, ο project manager Woodworth και προγραμματιστής N Abele παραιτήθηκαν από το πρόγραμμα με σκοπό να επικεντρωθούν στα δικά τους σχέδια, συμπεριλαμβανομένου του Lithium framework (προηγουμένως μέρος του έργου του CakePHP). Η υπόλοιπη ομάδα ανάπτυξης συνέχισαν να επικεντρώνονται στην αρχική σχεδίαση που είχε προηγουμένως οριστεί.

Γιατί cakePHP.

Το CakePHP αποβάλλει τη μονοτονία από τον προγραμματισμό και κάνει τα πράγματα πολύ πιο πρακτικά, προσοχή όχι κατά ανάγκη πιο εύκολα αλλά πιο πρακτικά. Παρέχει όλα τα εργαλεία που χρειαζόμαστε για να ξεκινήσουμε την κωδικοποίηση και ό, τι χρειάζεται για να γίνει η εκάστοτε διαδικτυακή εφαρμογή μας. Αντί να ανακαλύπτουμε τον τροχό κάθε φορά που ξεκινάμε ένα νέο έργο, το CakePHP έχει ένα σύνολο από εργαλεία και μεθόδους που μπορούν να φτάσουν σε βάθος ακόμα και στην πιο απαιτητική εφαρμογή διαδικτύου.

Το CakePHP Framework έχει μια ενεργή ομάδα προγραμματιστών και μια πολυάριθμη κοινότητα για υποστήριξη . Εκτός από τη διατήρηση μας από το να ανακαλύψουμε τον τροχό εκ νέου, με το CakePHP έχουμε έναν πυρήνα εφαρμογής πολύ καλά δοκιμασμένο και συνεχώς αναπτυσσόμενο.

Το CakePHP σε σύγκριση με ανταγωνιστές του είναι απλούστερο και πολύ ευκολότερο σε συγκεκριμένες λειτουργίες. Ένα βασικό γεγονός που καθόρισε την επιλογή του CakePHP για την συγκεκριμένη εργασία είναι, ο μεγάλος αριθμός δωρεάν ηλεκτρονικών μαθημάτων που υπάρχουν στο διαδίκτυο.

Παραθέτω κάποιες από τις δυνατότητες της CakePHP:

- *Active, friendly Official CakePHP discussion group*
- Flexible licensing
- Compatible with versions PHP 5.2.8 and greater
- Integrated CRUD for database interaction
- Application scaffolding
- Code generation
- MVC architecture
- Request dispatcher with clean, custom URLs and routes
- Built-in validation
- Fast and flexible templating (PHP syntax, with helpers)
- View helpers for AJAX, JavaScript, HTML forms and more

- Email, cookie, security, session, and request handling Components
- Flexible ACL
- Data sanitization
- Flexible caching
- Localization
- Works from any web site directory, with little to no Apache configuration involved

Κεφάλαιο 2. Δομή και κατασκευή ιστοσελίδων.

Στατικές και δυναμικές ιστοσελίδες.

Στις απαρχές του διαδικτύου όπου η δικτυακή υποδομή ανά τον κόσμο ήταν ακόμα κατά το πλείστον το τηλεφωνικό δίκτυο δεν μας επιτρεπόταν να υλοποιήσουμε ένα διαδίκτυο ιστοσελίδων με πλούσιο περιεχόμενο, που κατά συνεπαγωγή ήταν και μεγάλου όγκου δεδομένων. Έτσι μερικά χρόνια πριν θα αρκούσε σε κάποιον χρήστη μια απλή ιστοσελίδα με λίγο κείμενο και ίσως κάποιες λίγες χαμηλής ποιότητας φωτογραφίες. Αυτές οι μικρές και απλές ιστοσελίδες του τότε ήταν στατικές, δηλαδή το περιεχόμενο ήταν αρκετά δύσκολο να αλλάξει ή να τροποποιηθεί.

Με την ραγδαία ανάπτυξη του διαδικτύου εμφανίστηκε και η ανάγκη όλο και πιο συχνά οι κάτοχοι των ιστοσελίδων να πρέπει να εμπλουτίσουν το περιεχόμενό τους. Τα πρώτα βήματα ήταν δύσκολα και χρονοβόρα η πιο συνηθισμένη τακτική ήταν να ανεβαίνουν και να κατεβαίνουν στον server τα ανανεωμένα αρχεία της εκάστοτε ιστοσελίδας όπου ο κώδικας HTML και τα DATA ήταν ένα και το αυτό. Στην συνέχεια όμως εμφανίστηκαν τα πρώτα συστήματα διαχείρισης περιεχομένου τα λεγόμενα και αλλιώς CMS. Τα CMS έδωσαν την ευκαιρία στους χρήστες πλέον των ιστοσελίδων και όχι μόνο τους προγραμματιστές να ανανεώνουν το περιεχόμενο εύκολα, όσο συχνά το επιθυμούν χωρίς να χρειάζεται να το κάνουμε δια μέσου FTP και φυσικά χωρίς να απαιτούνται ιδικές γνώσεις. Δυναμικές ιστοσελίδες λοιπόν είναι εκείνες οι ιστοσελίδες που μπορούν να ανανεώνουν το περιεχόμενό τους μέσω ενός CMS συστήματος. Παρακάτω θα εξηγήσουμε με κάθε λεπτομέρεια τι είναι τα CMS και θα αναδείξουμε μερικά από τα πλέον δημοφιλή

CMS

Μέχρι πριν από λίγα χρόνια, ο μόνος τρόπος για να διατηρήσει μια εταιρία την ιστοσελίδα της ενημερωμένη ήταν να συνάψει συμβόλαιο με μια εταιρία παροχής υπηρεσιών συντήρησης. Τα τελευταία χρόνια, όμως, οι ίδιες οι εταιρίες παροχής τέτοιων υπηρεσιών προσφέρουν μια πολλά υποσχόμενη

εναλλακτική λύση. Πολλές από αυτές έχουν αναπτύξει ειδικά συστήματα, τα οποία μειώνουν το χρόνο και το κόστος λειτουργίας ενός δικτυακού τόπου.

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

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

Αυτό που αποκαλούμε πολλές φορές "δυναμικό περιεχόμενο" σε μία ιστοσελίδα δεν είναι άλλο παρά οι πληροφορίες που παρουσιάζονται στη ίδια την ιστοσελίδα και μπορούν να αλλάξουν από τους ίδιους τους διαχειριστές της μέσω κάποιας εφαρμογής, η οποία ουσιαστικά μπορεί να εισάγει (προσθέτει), διορθώνει και να διαγράφει εγγραφές σε πίνακες βάσεων δεδομένων, όπου τις περισσότερες φορές καταχωρούνται όλες αυτές οι πληροφορίες.

Αυτό σημαίνει ότι δεν χρειάζεται να δημιουργηθούν πολλές ξεχωριστές ιστοσελίδες για την παρουσίαση των πληροφοριών στο εκάστοτε website, αλλά αρκεί ένας ενιαίος σχεδιασμός στα σημεία όπου θέλουμε να εμφανίζεται το περιεχόμενό μας, καθώς και να υπάρχει ο ειδικός σε κάποια συγκεκριμένη

γλώσσα προγραμματισμού (ASP, PHP, Coldfusion, Perl, CGI κ.λπ.), ο οποίος αναλαμβάνει να εμφανίσει τις σωστές πληροφορίες στις σωστές θέσεις.

Έτσι, για το δικτυακό τόπο μιας εφημερίδας π.χ., που απαιτεί εύλογα καθημερινή ενημέρωση αλλά δεν χρησιμοποιεί κάποιο σύστημα Content Management, θα πρέπει ο υπεύθυνος για το σχεδιασμό του (designer) να δημιουργήσει μια σελίδα με τα γραφικά, την πλοήγηση και το περιβάλλον διεπαφής (interface) της ιστοσελίδας, ο υπεύθυνος ύλης - περιεχομένου να τοποθετήσει το περιεχόμενο στα σημεία της ιστοσελίδας που θέλει, και να ενημερωθούν οι σύνδεσμοι των υπόλοιπων σελίδων ώστε να συνδέονται με την καινούργια. Αφού την αποθηκεύσει, πρέπει να την ανεβάσει στον σέρβερ το αρχείο της ιστοσελίδας μαζί με τις υπόλοιπες σελίδες που άλλαξαν. Αυτή ήταν η διαδικασία ενημέρωσης κάποιας ιστοσελίδας μερικά χρόνια πριν.

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

Με την αυξητική τάση χρήσης των CMS στην Ελλάδα και το εξωτερικό, γίνεται εμφανές ότι το μέλλον του Διαδικτύου σε ότι αφορά περιεχόμενο και πληροφορίες που πρέπει να ανανεώνονται τακτικά, ανήκει στα προγράμματα διαχείρισης περιεχομένου, αφού προσφέρουν πολλά πλεονεκτήματα, ταχύτητα και ευκολίες στη χρήση τους.

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

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

Ένα ολοκληρωμένο CMS πρέπει να μπορεί να διαχειρίζεται όλες τις δυναμικές πληροφορίες της εκάστοτε ιστοσελίδας και να προσφέρει υπηρεσίες που εξυπηρετούν πλήρως τις ανάγκες των διαχειριστών του.

Επιγραμματικά, μερικά από τα πλεονεκτήματα και τα χαρακτηριστικά ενός ολοκληρωμένου CMS είναι:

- Γρήγορη ενημέρωση, διαχείριση και αρχειοθέτηση του περιεχομένου του δικτυακού τόπου
- Ενημέρωση του περιεχομένου από οπουδήποτε
- Ταυτόχρονη ενημέρωση από πολλούς χρήστες και διαφορετικούς υπολογιστές
- Να μην απαιτούνται ειδικές τεχνικές γνώσεις από τους διαχειριστές του

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

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

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

- Διαχείριση μελών
- Newsletters
- Forum

Συστήματα CMS Ανταγωνισμός.

Υπάρχουν πολλά CMS συστήματα με το καθένα από αυτά με τις δικές του ιδιαιτερότητες και λειτουργίες. Στο πλαίσιο αυτής της εργασίας θα σας παρουσιάσω τα 3 πιο δημοφιλή συστήματα (Joomla , WordPress , Drupal) διαχείρισης περιεχομένου με τα πλεονεκτήματά τους. Είναι σημαντικό να αναφέρουμε ότι και τα τρία αυτά συστήματα είναι γραμμένα σε γλώσσα προγραμματισμού PHP και είναι αρχιτεκτονήματα σύμφωνα με τις αρχές της MVC αρχιτεκτονικής. Επίσης και τα τρία χρησιμοποιούν την βάση δεδομένων MySQL για την αποθήκευση των δεδομένων.

To Joomla

Το Joomla είναι ένα βραβευμένο σύστημα διαχείρισης περιεχομένου (CMS), το οποίο μας επιτρέπει να κατασκευάζουμε ισχυρές ιστοσελίδες και online εφαρμογές. Πολλές πτυχές, συμπεριλαμβανομένης της ευκολίας χρήσης του και της επεκτασιμότητας του, έχουν κάνει το Joomla ένα από τα πιο δημοφιλή λογισμικά στο διαδίκτυο. Ένα σημαντικό κεφάλαιο που έκανε το Joomla να ξεχωρίσει είναι ότι το Joomla είναι ένα ανοικτού λογισμικού σύστημα ελεύθερα διαθέσιμο σε όλους χωρίς κόστος.

Ποια είναι μερικά πραγματικά παραδείγματα για το τι μπορεί να κάνει το Joomla.

Το Joomla χρησιμοποιείται σε όλο τον κόσμο για την κατασκευή ιστοσελίδων κάθε είδους και μεγέθους. Για παράδειγμα:

- Εταιρικές ιστοσελίδες ή διαδικτυακές πύλες (portals).
- Εταιρικά intranets και extranets.
- Online περιοδικά, εφημερίδες και εκδόσεις.
- Ιστοσελίδες ηλεκτρονικού εμπορίου και online κρατήσεις.
- Κυβερνητικές εφαρμογές.
- Ιστοσελίδες μικρών επιχειρήσεων.
- Ιστοσελίδες μη κερδοσκοπικού χαρακτήρα και ιστοσελίδες οργανισμών.
- Διαδικτυακές πύλες κοινοτήτων (Community-based portals).

- Σχολικές και Εκκλησιαστικές ιστοσελίδες.
- Ατομικές ή οικογενειακές ιστοσελίδες.
- Ιστοσελίδες μέσω μαζικής ενημέρωσης.
- Ιστοσελίδες κοινωνικής δικτύωσης.
- Forum.
- Μεσιτικές ιστοσελίδες.
- Κατάλογοι επιχειρήσεων.
- Πολύγλωσσες ιστοσελίδες
- Οι δυνατότητες του Joomla! είναι φαινομενικά ατελείωτες...

Η διαφορά μεταξύ του Joomla! και άλλων εμπορικών λύσεων CMS;

Σε αντίθεση με εμπορικές λύσεις που προσφέρουν διαφορετικές εκδόσεις ανάλογα με τον προϋπολογισμό σας, το Joomla διατίθεται σε μία μόνο έκδοση «που να τα καλύπτει όλα» - και αυτό διατίθεται δωρεάν. Πολλές εμπορικές λύσεις CMS διαθέτουν μια μεγάλη σειρά από πρόσθετα - σε ορισμένες περιπτώσεις έως και 80 από αυτά. Το Joomla προσφέρει ένα τρομερό αριθμό πρόσθετων εφαρμογών, που αριθμεί σήμερα περισσότερα από 9000 πρόσθετα. Το Joomla το έχουν κατεβάσει και εγκαταστήσει περισσότερες από 30.000.000 φορές. Είναι όμως φανερό ότι πλέον υπάρχει μια τάση προς τα κάτω και στα downloads και στις τελικές εγκαταστάσεις.

Ένα μεγάλο πλεονέκτημα που δεν πρέπει να παραβλέπουμε είναι η πολλή μεγάλη διάδοση του σε όλο τον κόσμο. Αυτή το γεγονός είναι πολύ ωφέλιμο. Αν για παράδειγμα μια εταιρεία ή ο οργανισμός προσλάβει έναν προγραμματιστή Joomla για την ανάπτυξη της ιστοσελίδας του, δημιουργείτε μια ιστοσελίδα ή μια εφαρμογή που θα βασίζεται σε μια ανοικτή πλατφόρμα που δεν δεσμεύει κανέναν σε μόνιμη συνεργασία με μια εταιρεία κατασκευής ιστοσελίδων ή με έναν συγκεκριμένο προγραμματιστή ή ακόμα με μια ιδιόκτητη, κλειστή εφαρμογή. Η ιστοσελίδα σας θα είναι ιδιοκτησία σας και θα έχετε όλα τα πνευματικά δικαιώματα, έτσι στο μέλλον μπορεί η εκάστοτε εταιρεία να συνεργαστεί με όποια εταιρεία κατασκευής επιθυμεί ή ακόμα και να προσλάβει έναν προγραμματιστή στην εταιρεία που θα διαχειρίζεται και θα σας κατασκευάζει την ιστοσελίδα. Αυτό έχει τεράστια σημασία ιδικά σε

ιστοσελίδες που θέλουμε να αντέξουν στον χρόνο και να συνεχίζουν να επεκτείνονται και να λειτουργούν.

Ένα ακόμα πολύ δυνατό πλεονέκτημα είναι η μελλοντική υποστήριξη του Joomla. Το Joomla είναι ένα από τα πιο δημοφιλή open source CMS που υπάρχουν σήμερα, όπως αυτό αποδεικνύεται από την ενεργή και αυξανόμενη κοινότητα των χρηστών και των ταλαντούχων προγραμματιστών που το χρησιμοποιούν. Οι Ρίζες του Joomla πάνε πίσω στο 2000 και με πάνω από 200.000 χρήστες της κοινότητας του Joomla να συνεισφέρουν.

- Το Joomla. Προσφέρει μια ισχυρή πλατφόρμα για την κατασκευή ιστοσελίδων που πληροί όλες τις προϋποθέσεις του SEO (search engine optimization)
- Η HTML διάταξη των ιστοσελίδων Joomla είναι καλά σχηματισμένη με αποτέλεσμα να καθιστάτε εύκολο για της μηχανές αναζήτησης να ψάχνουν την ιστοσελίδα για λέξεις κλειδιά.
- Όλες οι διευθύνσεις URL είναι απόλυτα φιλικές προς τις μηχανές αναζήτησης και δεν περιέχουν σύμβολα και αριθμούς αλλά λέξεις κλειδιά που ορίζεται εσείς. Για παράδειγμα μια διεύθυνση URL μπορεί να έχει την ακόλουθη μορφή www.yoursite.com/company-profile
- Το Joomla μας δίνει την δυνατότητα να προσθέσουμε λέξεις κλειδιά σε γενικό επίπεδο για όλη την ιστοσελίδα αλλά και σε κάθε σελίδα και άρθρο κάτι που στο μέλλον θα βοηθήσει πολύ στην εμφάνιση τις ιστοσελίδας σας στα αποτελέσματα των μηχανών αναζήτησης.

Το Joomla και η ασφάλεια της εκάστοτε ιστοσελίδας

Το Joomla είναι μια open source εφαρμογή, που ελέγχεται καθημερινά από εκατοντάδες ανθρώπους. Αν ρυθμιστεί σωστά και φιλοξενείται σε ένα ασφαλή και ποιοτικό διακομιστή, τότε μπορεί να είναι μια ασφαλής και αξιόπιστη λύση ακόμα και για μεγάλους δικτυακούς τόπους εταιρειών που επεξεργάζονται ευαίσθητες και απόρρητες πληροφορίες. Αυτό μπορεί να αποδειχθεί από τις μεγάλες εταιρείες που το χρησιμοποιούν για την κατασκευή της ιστοσελίδας τους.

Τι είναι το WordPress.

Το WordPress είναι ένα απλό σύστημα διαχείρισης περιεχομένου (CMS content management system) και αυτό με τις περισσότερες εγκαταστάσεις παγκοσμίως που μπορεί να χρησιμοποιηθεί στην κατασκευή οποιουδήποτε τύπου διαδικτυακής εφαρμογής. Αρχικά αναπτύχθηκε για την κατασκευή και δημοσίευση μόνο blogs ειδήσεις, άρθρα και νέα. Τα τελευταία χρόνια έχει εξελιχθεί σε ένα πλήρες πρόγραμμα κατασκευής ιστοσελίδων, e-shop, forum και όλων των ειδών web sites.

Είναι ένα λογισμικό ανοιχτού κώδικα (open source) και είναι πανίσχυρο στην δημιουργία, διαχείριση και διαμοιρασμό περιεχομένου στο διαδίκτυο και τα social media. Επίσης, όσες ιστοσελίδες φτιάχνονται με Wordpress είναι SEO Friendly – φιλικές προς τις μηχανές αναζήτησης, δηλαδή βγαίνουν στην «πρώτη σελίδα της Google» δημιουργώντας έτσι ωφέλιμη κίνηση στην εκάστοτε ιστοσελίδα.

Σήμερα, το WordPress έχει δώσει πνοή σε χιλιάδες blogs που είναι ιδιαίτερα δημοφιλή και έχουν πολύ μεγάλη επισκεψιμότητα. Το WordPress είναι ένα πολύ δυνατό εργαλείο και έχει πάρα πολλά καλογραμμένα πρόσθετα. Τα WordPress plugins ή πρόσθετα που είναι δωρεάν και είναι περισσότερα από 24.000, επιπρόσθετος διατίθενται παραπάνω από 1.700 θέματα δωρεάν καθώς και ένας τεράστιος αριθμός επί πληρωμή.

Ο αριθμός τους συνέχεια αυξάνει, και αυτό σημαίνει ότι αν ξεκινήσετε σήμερα την ιστοσελίδα σας θα έχετε απεριόριστη πρόσβαση σε χιλιάδες θέματα και πρόσθετα, χωρίς να χρειαστεί να αγοράσετε κάτι.

Γιατί να επιλέξουμε το WordPress ;

Υπάρχουν χιλιάδες άλλα Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems) παρόμοια με το WordPress και πολλά από αυτά είναι πάρα πολύ αποτελεσματικά στην αγορά που απευθύνονται. Ενδίκτικα να αναφέρουμε το drupal, joomla, magento, opencart, pligg, phpBB, flynax και πολλά άλλα.

Όμως, κανένα δεν φτάνει σε **SEO** το WordPress και κανένα δεν έχει την ίδια τεχνική υποστήριξη και την ίδια κοινότητα προγραμματιστών σε όλο τον κόσμο που με τις τεχνικές τους γνώσεις έχουν οδηγήσει το WordPress σε πραγματικά πάρα πολύ υψηλό επίπεδο.

Το WordPress είναι Open Source.

Το WordPress είναι μια open source platform ή μια πλατφόρμα ανοιχτού κώδικα, που σημαίνει ότι προγραμματιστές και ειδικοί από όλο τον κόσμο εργάζονται σκληρά με κοινό σκοπό να το βελτιώσουν. Το λογισμικό του WordPress μπορεί να χρησιμοποιηθεί άπειρες φορές και σε άπειρες ιστοσελίδες και ηλεκτρονικά καταστήματα χωρίς κανέναν περιορισμό και κανένα κόστος αγοράς άδειας.

Δεν χρειάζεται να πληρώσετε σε κανένα τρίτο μέρος ή εταιρεία για να αγοράσετε κάποια άδεια λογισμικού για το WordPress, όπως γίνεται με τα Windows. Το μόνο που χρειάζεται να κάνετε είναι να πληρώσετε για το Domain name, το Hosting και το κόστος κατασκευής της ιστοσελίδας σας.

Είναι επεκτάσιμο.

Εάν ήδη χρησιμοποιείτε WordPress, τότε δεν θα πρέπει να ανησυχείτε σχετικά με την επεκτασιμότητα του. Είναι πολύ εύκολο να προσθέσετε κάποιο πρόσθετο.

Δεν υπάρχουν πολλές περιπτώσεις ιστοσελίδων που δεν μπορούν να υλοποιηθούν από κάποιο πρόσθετο επεκτασιμότητας του WordPress. Αρκετές ιστοσελίδες το χρησιμοποιούν ήδη για πάνω από μια δεκαετία και

έχουν δημοσιεύσει χιλιάδες σελίδες και άρθρα χωρίς ποτέ να είχαν κάποιο πρόβλημα. Θεωρώ ότι δεν θα βρεθεί ποτέ κάποιο πρόβλημα τεχνικό ή μη που να μην μπορεί το WordPress να το ξεπεράσει από άποψη λάθους ιδικά σε ιστοσελίδες τύπου Blog.

Το WordPress.com από μόνο του αποτελεί ένα φωτεινό παράδειγμα επεκτασιμότητας. Με μία μόνο εγκατάσταση multisite το WordPress τρέχει σχεδόν πάνω από 30 εκατομμύρια web sites όπως τα **TechCrunch**, **Gigaom**, **CNN blogs** και πολλά άλλα.

Έχει Spam Protection.

Η προσωπική μου εμπειρία αλλά και η έρευνα που έκανα δείχνουν ότι κανένα από τα χιλιάδες ανταγωνιστικά πρόσθετα - κώδικες και προγράμματα για spam protection δεν είναι τόσο αξιόπιστα, το WordPress έχει την καλύτερη λύση για spam protection. Τα παλιότερα παραδοσιακά web site χωρίς WordPress, είχανε μεγάλο θέμα στη λειτουργία τους, γιατί εάν άφηνες ανοιχτά τα σχόλια ή κάποια πεδία εγγραφής, αυτά γέμιζαν με spam. Με ψεύτικα δηλαδή κομμάτια κειμένου που είχαν ως στόχο να μειώσουν το κύρος της ιστοσελίδας.

Με πρόσθετα - plugins όπως το **Akismet**, **GASP**, **Conditional Captcha**, και πολλά άλλα, μπορείτε να σταματήσετε χιλιάδες **spam comments** και νέες spam εγγραφές χρηστών. Πιστεύω ότι το WordPress έχει την καλύτερη λύση στο spam protection από οποιοδήποτε άλλο ανταγωνιστικό σύστημα διαχείρισης περιεχομένου.

Μεγαλύτερη Κοινότητα Τεχνικής Υποστήριξης.

Η τεράστια **κοινότητα τεχνικής υποστήριξης** από bloggers, developers, designers και απλούς χρήστες σε όλο τον κόσμο, κάνει το WordPress τόσο λειτουργικό, δυνατό και σταθερό.

Δεν θα αντιμετωπίσετε ποτέ κανένα πρόβλημα στο να προσθέσετε ή να αφαιρέσετε μια νέα δυνατότητα στο WordPress. Υπάρχουν πάρα πολλές πηγές στο διαδίκτυο και πλήθος ιστοσελίδων που μπορούμε να βρούμε

λύσεις και βοήθεια σε όλα τα ζητήματα που θα προκύψουν. Σας παραθέτω μερικά από τα σημαντικότερα support forums που παρέχουν δωρεάν υποστήριξη:

- WordPress.org Forums – Το επίσημο **support forum** για WordPress.
- WordPress Stackexchange – Το μεγαλύτερο μη επίσημο forum για βοήθεια στο WordPress.

Άνθρωποι από όλο τον κόσμο, υποστηρίζουν και συνεισφέρουν στην συνεχή του ανάπτυξη και εξέλιξη. Αυτό και μόνο το καθιστά το καλύτερο blogging software για την ανάπτυξη και δημιουργία άρτιων ιστοσελίδων.

WordPress Themes & Plugins.

Το WordPress.org είναι το πρώτο web site που μπορεί να βρει κάποιος δωρεάν WordPress themes. Το επίσημο αποθετήριο θεμάτων της WordPress (WordPress Repository) προσφέρει χιλιάδες WordPress themes & plugins χωρίς κάποιο κόστος.

Έχουνε πολύ μεγάλη ποικιλία δωρεάν θεμάτων για να ξεκινήσει κάποιος την ιστοσελίδα του, όμως δεν τα συνιστώ γιατί είναι περιορισμένων δυνατοτήτων. Συνιστώ μόνο την αγορά πληρωμένων WordPress Themes.

Αν ψάχνετε να προσθέσετε νέες δυνατότητες σε κάποιο WordPress web site, σίγουρα θα βρείτε κάποιο plugin για αυτό. Τις περισσότερες φορές το WordPress plugins repository έχει το plugin δωρεάν. Σε αντίθετη περίπτωση ο εκάστοτε κατασκευαστής θα αναγκαζόταν να ξοδέψει χρόνο ή μια μικρή περιουσία για να αποκτήσει όλα αυτά τα πρόσθετα που θα κάνουν την ιστοσελίδα λειτουργική.

Μερικά από τα πιο χρήσιμα και βασικά WordPress plugins – πρόσθετα είναι: WordPress SEO by Yoast, Akismet, GASP, JetPack, BuddyPress, τα οποία υλοποιούν πολύ απλά μερικές από τις βασικότερες λειτουργίες μιας σύγχρονης δυναμικής ιστοσελίδας.

Drupal

Λόγος Νούμερο 1

Ο νούμερο ένα λόγος που έπρεπε να περιγράψουμε το Drupal σε αυτήν την εργασία είναι ότι είναι πραγματικά ευέλικτο. Σε σχέση με το σύστημα διαχείρισης περιεχομένου Joomla, είναι πολύ σύνηθες το γεγονός να μην μπορεί να εξυπηρετηθεί μία απαίτηση κάποιου πελάτη επειδή το σύστημα μας περιορίζει προγραμματιστικά. Δεν μας δίνει εργαλεία με τα οποία να μπορεί κάποιος προγραμματιστής να επεκτείνει την λειτουργικότητα. Για παράδειγμα, σε μία ιστοσελίδα υπάρχει μία νέα απαίτηση μετά την κατασκευή της σε πλατφόρμα Joomla και αυτή είναι να βάλουμε ένα νέο πεδίο στην ιστοσελίδα, για να εκτελεστεί κάτι τέτοιο στο Joomla με τα εργαλεία που δίνει ο πυρήνας του είναι αδύνατον για αυτόν τον λόγο πρέπει να χρησιμοποιήσουμε εξωτερικά πρόσθετα όπως το K2, το JReviews ή κάποιο άλλο CCK για να κατασκευάσουμε και να επεκτείνουμε την ιστοσελίδα. Αυτό όμως δεν ήταν πάντα εύκολο και κομψό στην υλοποίηση γιατί πολλές φορές αυτές οι απαιτήσεις ερχόντουσαν αφού η ιστοσελίδα είναι ήδη στον "αέρα" - online, και έτσι δεν είναι πολύ εύκολη η αλλαγή της δομής της.

Από την άλλη, το Drupal έχει ένα εγγενές, μέσα στον πυρήνα του, πακέτο σχεδιασμού περιεχομένου (CCK), το οποίο σας επιτρέπει να κατασκευάσετε τους δικούς σας τύπους περιεχομένου με τα δικά σας πεδία. Οι τύποι περιεχομένου είναι πακέτα δεδομένων από πεδία. Για παράδειγμα, το Joomla είχε μόνο έναν τύπο περιεχομένου, τον τύπο "Άρθρο", όπου υπήρχε η δυνατότητα να εισάγετε τον τίτλο του άρθρου, το κυρίως κείμενο του άρθρου, την ημερομηνία δημοσίευσης και κάποια άλλα στοιχεία. Στο Drupal μπορούμε να κατασκευάσετε τους δικούς μας τύπους περιεχομένου και να έχουμε όσα πεδία θέλουμε μέσα σε αυτούς. Επίσης μπορούμε να προσθέσουμε περισσότερα πεδία ακόμα και αν η ιστοσελίδα είναι στον "αέρα" - online. Έτσι, για παράδειγμα, εάν κατασκευάσουμε ένα online κατάστημά σε Drupal και κάποια στιγμή σκεφτούμε ότι πρέπει να χρεώνονται τα έξοδα αποστολής βασισμένα στο βάρος των προϊόντων, είναι πολύ εύκολο να προσθέσετε ένα

πεδίο στον τύπο περιεχομένου "προϊόν" το οποίο να ονομάζεται "βάρος" και να εισάγετε εκεί το βάρος των προϊόντων σας.

Αυτό που συνοδεύει τα περισσότερα web sites σε Drupal είναι ένα εξωτερικό module το οποίο ονομάζεται "Views". Το module αυτό είναι το δημοφιλέστερο στον κόσμο του Drupal επειδή σας επιτρέπει να αντλείτε δεδομένα από την βάση δεδομένων σας και συγκεκριμένα από τους τύπους περιεχομένου που έχετε δημιουργήσει μόνοι σας και να τα χρησιμοποιείτε με όποιον τρόπο θέλετε στην κατασκευή της σελίδας σας. Βέβαια, εδώ δεν φαίνεται πολύ καλά, αλλά με το "Views", μπορούμε να επιλέξουμε τα πεδία που θέλουμε, να ορίσουμε τα κριτήρια των φίλτρων μας και τα κριτήρια της ταξινόμησής μας και να κατασκευάσουμε την ιστοσελίδα όπως ακριβώς θέλουμε να εμφανίζεται στους επισκέπτες μας. Αυτό είναι πάρα πολύ ισχυρό, έτσι στην επερχόμενη έκδοση του Drupal, η οποία αναμένεται στο τέλος της χρονιάς, το "Views" θα ενσωματωθεί στον πυρήνα του Drupal. Ουσιαστικά, το "Views" είναι ένα εργαλείο κατασκευής ερωτημάτων SQL που επιτρέπει σε κάποιον χωρίς να είναι προγραμματιστής να κατασκευάσει τα ερωτήματα και να λάβει δεδομένα από την βάση εμφανίζοντάς τα με τον τρόπο που επιθυμεί στους επισκέπτες της ιστοσελίδας του.

Ένα άλλο αξιοσημείωτο σημείο είναι ότι στο Drupal μπορούμε να έχουμε ομάδες χρηστών οι οποίοι έχουν ρόλους και ο κάθε ρόλος έχει τα δικά του δικαιώματα.

Έτσι, μπορούμε να φτιάξουμε όσους ρόλους θέλουμε για το διαχειριστικό μέρος της ιστοσελίδας μας και να επιτρέψουμε στους χρήστες να κάνουν μόνο αυτά που πρέπει και θέλουμε να κάνουν ή να έχουν πρόσβαση. Για παράδειγμα, μπορούμε να κατασκευάσουμε ένα ηλεκτρονικό κατάστημα (e-shop) και να έχουμε μερικούς χρήστες οι οποίοι έχουν πρόσβαση στις τιμές χονδρικής γιατί είναι οι πελάτες χονδρικής σας. Αυτό πάλι είναι κάτι το οποίο μπορούμε να ρυθμίσουμε μέσα από γραφικό περιβάλλον και τις περισσότερες φορές δεν θα χρειαστείτε καθόλου βοήθεια από κάποιον προγραμματιστή.

Άλλο ένα προηγμένο και πολύ καλό χαρακτηριστικό του Drupal, είναι το πρόσθετο "Rules" το οποίο σας επιτρέπει να κατασκευάζετε τις δικές σας

ροές εργασιών - (workflows). Για παράδειγμα, μπορούμε να ρυθμίσουμε το "Rules" στο Drupal έτσι ώστε να στείλει ένα γραπτό μήνυμα στα κινητά των πελατών κάθε φορά που βάζουν μία παραγγελία. Πάλι, χρησιμοποιώντας ένα πρακτικό γραφικό περιβάλλον, μπορούμε να κατασκευάσουμε αυτό τον κανόνα και μπορείτε να τον παραμετροποιήσουμε και να τον επεξεργαστούμε όπως ακριβώς θέλουμε.

Για να είμαστε απόλυτα ειλικρινείς πρέπει να σημειώσουμε ότι κάποιες από αυτές τις δυνατότητες χρειάζονται αρκετά καλή εξοικείωση με το Drupal και τα εργαλεία του θέλει δηλαδή περισσότερο διάβασμα σε online οδηγούς και περισσότερο ανάλυση στον τρόπο που λειτουργεί στα ενδότερά του το Drupal σαν σύστημα. Συνήθως τέτοιες εργασίες δεν αναλαμβάνει κάποιος απλό χρήστης αλλά τα πραγματοποιεί κάποιος προγραμματιστής ή η εταιρεία, βέβαια σε αρκετές περιπτώσεις οι προγραμματιστές απλώς εγκαθιστούν την πλατφόρμα του Drupal μαζί με τα modules- πρόσθετα με τρόπο τέτοιο ώστε εν' συνεχεία κάποιος απλός χρήστης μπορεί πάρα πολύ εύκολα να το παραμετροποιήσει και να το "πειράξει" για να δουλέψει όπως ακριβώς πρέπει.

Λόγος Νούμερο 2

Ο δεύτερος λόγος θα παραθέσω είναι ότι το Drupal είναι ένα πάρα πολύ καλό σύστημα διαχείρισης περιεχομένου, γιατί είναι πάρα πολύ φιλικό στις μηχανές αναζήτησης. Αυτό είναι πάρα πολύ σημαντικό για τις ιστοσελίδες μας και για τις ιστοσελίδες των πελατών που θέλουν να πετύχουν κάποια θέση στις αναζητήσεις.

Έτσι, όταν άρχισα να ερευνώ για το Drupal CMS, έλεγα εάν τα πηγαίνει καλά με τις μηχανές αναζήτησης. Αυτό που ανακάλυψα ήταν ότι η κοινότητα του Drupal έχει κατασκευάσει αναρίθμητα πρόσθετα που αντιμετωπίζουν το ζήτημα αυτό ως θέμα ύψιστης σημασίας.

Υπάρχουν πολλά πρόσθετα που θα μας βοηθήσουν να εμφανίσουμε την ιστοσελίδα που κατασκευάσαμε ψηλότερα, και θα μπορούσα να μιλάω ώρες για αυτά, αλλά θα επιλέξω στα πλαίσια αυτής της εργασίας να παρουσιάσω μόνο το Pathauto, το οποίο μας επιτρέπει να κατασκευάζουμε την δομή των

URL με τον τρόπο που θέλουμε. Έτσι, για παράδειγμα, όταν κατασκευάζουμε μια ιστοσελίδα με το Drupal, μπορούμε να τοποθετήσουμε τα άρθρα της ιστοσελίδας στον φάκελο "articles" - άρθρα και έτσι το κάθε άρθρο να έχει το τίτλο "articles" - άρθρα στο URL του. Αυτό είναι πολύ σημαντικό για τις μηχανές αναζήτησης, αλλά και από θέμα ευχρηστίας για τους χρήστες επειδή οι χρήστες κοιτούν στο URL της εκάστοτε ιστοσελίδας για να προσανατολιστούν και να καταλάβουν πού ακριβώς βρίσκονται. Αυτό που ανακάλυψα είναι ότι αρκετοί κατασκευάστε μετά την μεταφορά πολλών ιστοσελίδων από Joomla σε Drupal και από συστήματα κλειστού κώδικα σε Drupal, ήταν ότι αμέσως μετά την μεταφορά σε Drupal, οι ιστοσελίδες είχαν πολύ καλύτερα αποτελέσματα στην κατάταξη του Google. Εδώ μπορούμε να δούμε ένα παράδειγμα σχετικά με μία Ελληνική ιστοσελίδα στην οποία μετά την μεταφορά της σε Drupal, είχαμε σχεδόν 40% αύξηση στην κατάταξη και στην επισκεψιμότητα από τα οργανικά αποτελέσματα. Αυτό που συνηθίζετε στην αγορά να λέγετε είναι ότι "το Google πράγματι αγαπάει το Drupal".

Λόγος Νούμερο 3

Το Drupal είναι αξιόπιστο και μερικά από τα μεγαλύτερα web site στον κόσμο το χρησιμοποιούν. Εντυπωσιακό είναι το γεγονός ότι στα πλαίσια της έρευνας μου για το Drupal ανακάλυψα ότι κάποια από τα μεγαλύτερα web site, όπως ο Λευκός Οίκος των ΗΠΑ είναι κατασκευασμένο σε Drupal. Ή το MTV UK. France24.com. Το μουσείο του Λούβρου. Το Economist. Το web site των βραβείων Grammy. ING US. Examiner. Έτσι, είναι προφανές ότι ένα τόσο δυνατό πελατολόγιο κρύβει πίσω του μια δυνατή εφαρμογή δηλαδή αφού αυτοί οι οργανισμοί βασίζονται σε Drupal, τότε θα ήταν πολύ καλό και για μικρότερες ιστοσελίδες. Εδώ μπορούμε να δούμε την ανάπτυξη των δημοφιλέστερων συστημάτων διαχείρισης περιεχομένου ανοιχτού κώδικα αυτή την στιγμή. Όπως μπορείτε να διαπιστώσετε, τον προηγούμενο χρόνο, το Drupal είχε περισσότερο από 20% ανάπτυξη στην χρήση του. Αυτή την περίοδο, πάνω από ένα εκατομμύριο web site τρέχουν σε Drupal.

Λόγος Νούμερο 4

Το Drupal είναι ασφαλές. Οι προμηθευτές των συστημάτων κλειστού κώδικα έχουν πάντα ένα επιχείρημα εναντίον των συστημάτων ανοιχτού κώδικα

λέγοντας ότι αφού είναι ανοιχτού κώδικα, οποιοσδήποτε μπορεί να ρίξει μια ματιά στον κώδικά τους, να μελετήσει τον κώδικα και να βρει κενά ασφαλείας. Αυτό ισχύει για τα υπόλοιπα συστήματα, αλλά όχι για το Drupal. Επειδή το Drupal έχει δώσει πολύ μεγάλη σημασία στα θέματα ασφάλειας και έχει σχηματίσει μία ομάδα από επαγγελματίες ειδικούς σε θέματα ασφαλείας προγραμματιστές οι οποίοι ψάχνουν συνεχώς και κλείνουν τα κενά ασφαλείας που ανακαλύπτουν στο σύστημα.

Και αυτό επειδή γνωρίζουν ότι αυτό το σύστημα χρησιμοποιείται από τα μεγαλύτερα web site του κόσμου. Για να κάνουμε την ουσιαστική σύγκριση είναι αλήθεια, ότι στατιστικά ιστοσελίδες σε Drupal και Joomla έχουν περισσότερα κενά ασφαλείας, από ότι αυτές που είναι κατασκευασμένες σε Drupal.

Λόγος Νούμερο 5

Το Drupal έχει μία πολύ ενεργή κοινότητα. Περισσότερο από 1 εκατομμύριο άνθρωποι εργάζονται για να κάνουν το σύστημα καλύτερο και βρίσκονται πάντα διαθέσιμοι για να μας βοηθήσουν είτε δωρεάν μέσω των καναλιών IRC είτε μέσω των Forum, είτε επί πληρωμή. Έτσι, είναι πολύ εύκολο για κάποιον να βρει καλούς προγραμματιστές, καλές εταιρείες κατασκευής ιστοσελίδων και μπορείτε να συνεργαστείτε με όποιον θέλει και να αλλάξει συνεργάτες οποτεδήποτε θέλει.

Επίσης, υπάρχουν πάνω από 20.000 δωρεάν modules - πρόσθετα έτοιμα να τα χρησιμοποιηθούν κατά βούληση και να επεκτείνουν τις δυνατότητες της εκάστοτε ιστοσελίδας.

Εδώ μπορούμε να δούμε μερικές φωτογραφίες από τα συνέδρια DrupalCon, τα συνέδρια του Drupal οργανώνονται μέσα στο έτος και μπορεί ο οποιοσδήποτε να λάβει μέρος και να παρευρεθεί σε αυτά. Τα Drupal Con είναι δύο φορές τον χρόνο αλλά υπάρχουν και μηνιαία μικρότερα συνέδρια, τα Drupal Camps και τα Drupal Meetups στα οποία μπορεί κάποιος να παρακολουθήσει και να διαπιστώσει πόσο ενεργή είναι η κοινότητα του Drupal. Ενέργειες σας τα DrupalCon είναι πολύ χρήσιμες για μια ανοιχτού κώδικα εφαρμογή καθώς σε βάθος χρόνου αναπτύσσεται με τρόπο τέτοιο

ώστε να διατηρείτε το ανταγωνιστικό πλεονέκτημα. Όσο μεγαλύτερη η συμμετοχή τόσο περισσότερες οι ιδέες που θα εξελίξουν τελικά την εφαρμογή.

Στην πλατφόρμα του Drupal έχουν κατασκευαστεί πάρα πολλά modules καθώς και πολλές διανομές Drupal. Οι διανομές για Drupal είναι σετ από εγκαταστάσεις Drupal τις οποίες μπορείτε να κατεβάσουμε, να εγκαταστήσουμε και να έχουμε μία ιστοσελίδα έτοιμη στον αέρα σε μερικά λεπτά. Οι διανομές στην ουσία είναι ένα σύνολο από πρόσθετα μαζί με τον πύρινα που μπορούν να εξυπηρετήσουν μια συγκεκριμένη ανάγκη ιστοσελίδας πολύ γρήγορα. Ένα παράδειγμα είναι η Ελληνική διανομή Open Deals, η οποία είναι μια διανομή "Group on out of the box" είναι μια διανομή που υλοποιεί πολύ γρήγορα διαδικτυακές εφαρμογές συναλλαγών. Την κατεβάζουμε, την εγκαθιστούμε, και η ιστοσελίδα είναι σχεδόν έτοιμη, με σχεδόν μηδενικό κόστος.



Εικόνα 1: Συνέδριο Drupal

Λόγος Νούμερο 6

Το Drupal είναι καινοτόμο. Η κοινότητα του Drupal ενσωματώνει ενεργά κάθε νέα τεχνολογία μέσα στο Drupal. Ως αποτέλεσμα, το Drupal ήταν αυτό που ενσωμάτωσε πρώτο τα Responsive Themes τα οποία θα σας παρουσιάσω αργότερα. Ήταν το Drupal το οποίο ξεκίνησε την υλοποίηση της άμεσης επεξεργασίας (inline editing), μία τεχνολογία που θα είναι έτοιμη τους επρχόμενους μήνες και θα μπορεί να χρησιμοποιηθεί στο Drupal 7 αλλά και στο επερχόμενο Drupal 8, η οποία μας επιτρέπει να επεξεργαστούμε το περιεχόμενο της ιστοσελίδας κατ' ευθείαν χωρίς να χρειαστεί να μεταβούμε σε περιβάλλον διαχείρισης. Από το περιβάλλον χρήσης, εκεί που επισκέπτεστε την ιστοσελίδα, όταν δούμε κάποιο λάθος, τυπογραφικό ή οτιδήποτε άλλο, μπορούμε να πατήσουμε στο κουμπί της επεξεργασίας και να αλλάξουμε τα πάντα "live" εκείνη την στιγμή.

Επίσης, υπάρχουν μηχανισμοί antisppam όπως το Mollom, το οποίο προστατεύει την ιστοσελίδα μας από κακόβουλες ενέργειες που γεμίζουν με άχρηστο περιεχόμενο την ιστοσελίδα μας σε σχέση με άλλες πλατφόρμες δεν έχει το πλεονέκτημα να μην είναι ενοχλητικό για τους χρήστες.

Το Drupal παρέχει εξειδικευμένα εργαλεία για ηλεκτρονικά καταστήματα ένα από αυτά που έχει αναπτυχθεί είναι το εδώ "Faceted Search" σε Drupal όπου μπορεί με 2-3 κλικ, να βρει αυτό ακριβώς που ψάχνει ο χρήστης ανάμεσα σε χιλιάδες προϊόντα. Το "Faceted Search" είναι μία τεχνολογία αναζήτησης, στην οποία μπορούμε να χρησιμοποιήσουμε πολλαπλά φίλτρα Ajax για να εντοπίσουμε αυτό ακριβώς που ψάχνουμε. Επίσης όσα αφορά τις αναζητήσεις στο Drupal, έχει αναπτυχθεί ένα εξωτερικό module βασισμένο στο Search API του Drupal, το οποίο ονομάζεται "Findastic", και είναι πάρα πολύ γρήγορο και προηγμένο σε τέτοιο βαθμό που δίνει ακόμα ένα ανταγωνιστικό πλεονέκτημα στο Drupal.

Επίσης ένα ακόμα πολύ δυνατό ανταγωνιστικά κομμάτι του Drupal είναι το Symfony2. Το Symfony2 είναι η επόμενη μεγάλη καινοτομία του Drupal. Το Symfony2 είναι ένα PHP Framework που έχει αναπτυχθεί από μία άλλη ομάδα προγραμματιστών, το οποίο ενσωματώνεται και αυτό στο Drupal 8 και

κατά ένα τρόπο ενώνει αυτές τις δύο κοινότητες με αποτέλεσμα τα πράγματα να κινούνται πιο γρήγορα και στο κομμάτι της επεκτασιμότητας.

Είναι ακόμα πολύ σημαντικό σημείο, που στην πλατφόρμα του Drupal υλοποιείται με τον πλέον αξιόπιστο τρόπο είναι αυτό του "Responsive Design". Το «Responsive Design» είναι στην ουσία η σχεδίαση εκείνη που επιτρέπει στην ιστοσελίδα μας να είναι προσβάσιμη από κινητές συσκευές Smartphone's και Tablets.

Λόγος Νούμερο 7

Τέλος, το Drupal είναι δωρεάν και ανοιχτού κώδικα και μπορούμε να το κατεβάσουμε άμεσα, να το δοκιμάσουμε και να διαπιστώσουμε τα πλεονεκτήματα και τα μειονεκτήματα και τι μπορούμε να κάνουμε με αυτό και τι όχι. Επίσης είναι σημαντικό το γεγονός ότι είναι πολύ εύκολο κάποιος να βρει ένα Drupal ιδικό καθώς είναι πολύ διαδεδομένο. Τέλος δεν είμαστε υποχρεωμένοι να πληρώσουμε για οποιαδήποτε άδεια ή δικαιώματα, πληρώνουμε μόνο για τον χρόνο του προγραμματισμού και της σχεδίασης της ιστοσελίδας.

Τι είναι open source.

Open source είναι μία έννοια που υποδηλώνει πως κάποιο πρόγραμμα ή λειτουργικό σύστημα είναι ανοιχτού κώδικα και μπορούμε να επεμβούμε στον τρόπο λειτουργίας του.

Το open source δηλαδή σημαίνει ότι μπορεί κάποιος απλός χρήστης να επέμβει στον κώδικα που τρέχει για να επιτελέσει διάφορες λειτουργίες, και να το κάνει να τρέχει όπως θέλει αυτός, ή να προσθαφαιρέσει διάφορα πραγματάκια, ώστε να το φέρει στα μέτρα του.

Αυτή είναι και η σημασία του **open source** - ανοιχτού κώδικα και μόνη προϋπόθεση για να το κάνει αυτό, είναι να γνωρίζει προγραμματισμό ώστε να είναι ικανός να επέμβει σωστά και να μην τα κάνει "μαντάρα".

Ενώ προγράμματα που είναι κλειστού κώδικα, είναι ας το πούμε fix τρέχουν μόνο με ένα συγκεκριμένο τρόπο και δεν μπορεί κανένας να επέμβει, παρά μόνο ο/οι προγραμματιστής/ές που το φτιάξανε.

Ένα λειτουργικό σύστημα που είναι open source είναι το Linux και ένα από τα πάρα πολλά προγράμματα είναι το OpenOffice που είναι και δωρεάν, σε αντίθεση με τα Windows και τα Office που είναι κλειστού κώδικα και δεν είναι δωρεάν.

Κεφάλαιο 3. Cactus CMS.

Πλεονεκτήματα.

Στο κεφάλαιο αυτό νομίζω ότι είναι σκόπιμο να εξηγήσουμε την διαφορετικότητα του Συστήματος διαχείρισης περιεχομένου που πραγματεύεται η συγκεκριμένη εργασία. Όπως είδαμε στο προηγούμενο κεφάλαιο υπάρχουν αρκετά παρόμοια συστήματα που συγκεντρώνουν πολλά πλεονεκτήματα. Η βασική θεώρηση που έρχεται να καταρρίψει αυτή η εργασία είναι ότι κάθε CMS έχει ως τελικό χρήστη κάποιον που μπορεί να διαχειριστεί το περιεχόμενο μόνο από την σκοπιά του κειμένου και των πολυμέσων. Δεν είναι σε θέση δηλαδή να επεκτείνει το περιεχόμενο της ιστοσελίδας του με κώδικα. Αυτή η τελευταία θέση πυροδοτεί μια μεγάλη αντιπαράθεση ανάμεσα σε εκείνους που θέλουν να προσφέρουν στον τελικό χρήστη μιας δυναμικής ιστοσελίδας εργαλεία διαχείρισης τέτοια που δεν απαιτούν γνώσεις προγραμματισμού. Είναι γεγονός ότι όσο πιο απλή είναι η διαδικασία στο τελικό χρήστη τόσο μεγαλύτερη είναι η πολυπλοκότητα του συστήματος από προγραμματιστική άποψη.

Το Cactus CMS έχει ως στόχο να δέσει ταυτόχρονα μεταξύ τους τις δύο βασικές και αντίθετες ως τώρα θέσεις. Πρώτη θέση: Πρέπει να παρέχονται εργαλεία διαχείρισης της εκάστοτε ιστοσελίδας που δεν απαιτούν ιδιικές γνώσεις προγραμματισμού. Δεύτερη θέση: Πρέπει κάθε ιστοσελίδα να είναι προγραμματιστικά επεκτάσιμη ανά πάσα στιγμή από πολλούς προγραμματιστές.

Όπως γίνεται εύκολα αντιληπτό το Cactus CMS έρχεται να λειτουργήσει σαν αμφίβιο σύστημα. Δηλαδή έχει λειτουργίες που καλύπτουν ταυτόχρονα και τις δυο παραπάνω θέσεις. Η ταυτόχρονη συμπεριφορά του Cactus σας ένα online php editor και ενός κλασικού CMS του δίνει το ανταγωνιστικό πλεονέκτημα ως προς τους ανταγωνιστές τους.

Ένα βασικό πλεονέκτημα που προκύπτει από την αμφίβια λειτουργία του Cactus είναι ότι επιτρέπει την ταυτόχρονη παρέμβαση χρηστών σε διαφορετικά κομμάτια ενώ η ιστοσελίδα είναι online. Για παράδειγμα έστω ότι

έχουμε μία ιστοσελίδα που προβάλλει άρθρα και μια δεδομένη στιγμή αποφασίζουμε ότι η ιστοσελίδα θα πρέπει να επεκτείνει το περιεχόμενο της με περισσότερες λειτουργίες χωρίς όμως να ανακόψουμε την λειτουργία της απλής δημοσίευσης άρθρων. Τότε το σύστημα δίνει την λειτουργία στον προγραμματιστή ή προγραμματιστές να γράψουν τον κώδικα επέκτασης online χωρίς να χρειαστεί να χρησιμοποιήσουν εργαλεία εκτός του Cactus. Είναι σημαντικό να τονίσουμε ότι δεν χρειάζεται να κάνουμε deploy τον κώδικα στον server καθώς ο κώδικας δημιουργείται και παράγεται online στον Server.

Τέλος είναι σημαντικό να αναφέρουμε ότι το Cactus CMS θα δώσει λειτουργίες τέτοιες που θα κάνουν την εκάστοτε ιστοσελίδα φιλική προς τις μηχανές αναζήτησης. Είναι δεδομένο ότι δεν μπορεί να υπάρξει ιστοσελίδα χωρίς τις μηχανές αναζήτησης. Έχουν υλοποιηθεί μηχανισμοί τέτοιοι που βάζουν τα θεμέλια ως προς την σωστή δομή, δεν φτάνει όμως μόνο αυτό. Είναι γεγονός ότι η πρώτη θέση σε κάποια λέξη κλειδί μιας μηχανής αναζήτησης είναι παράγωγο πολλών συνιστωσών που δεν έχουν πάντα σχέση με το προγραμματιστικό κομμάτι της ιστοσελίδας.

Μειονεκτήματα.

Το βασικό μειονέκτημα της εφαρμογής είναι ότι δεν υπάρχει ακόμα κοινότητα υποστήριξης και επέκτασης. Ένα από τα βασικά που κάνει ένα τέτοιο σύστημα να ξεχωρίσει είναι η κοινότητα που το επεκτείνει και το συντηρεί, οι λόγοι που ένα τέτοιο σύστημα αποκτάει μεγάλη κοινότητα είναι η λογική του ανοικτού λογισμικού αλλά και η απλότητα χειρισμού του.

Περιπτώσεις Χρήσης.

Νομίζω ότι στο σημείο αυτό είναι χρήσιμο να καταγράψουμε τις βασικές περιπτώσεις χρήσης Cactus CMS γύρο από τις οποίες θα στηθεί το σύστημα μας.

Περίπτωση Χρήσης: Ο χρήστης τροποποιεί το περιεχόμενο της Ιστοσελίδας.

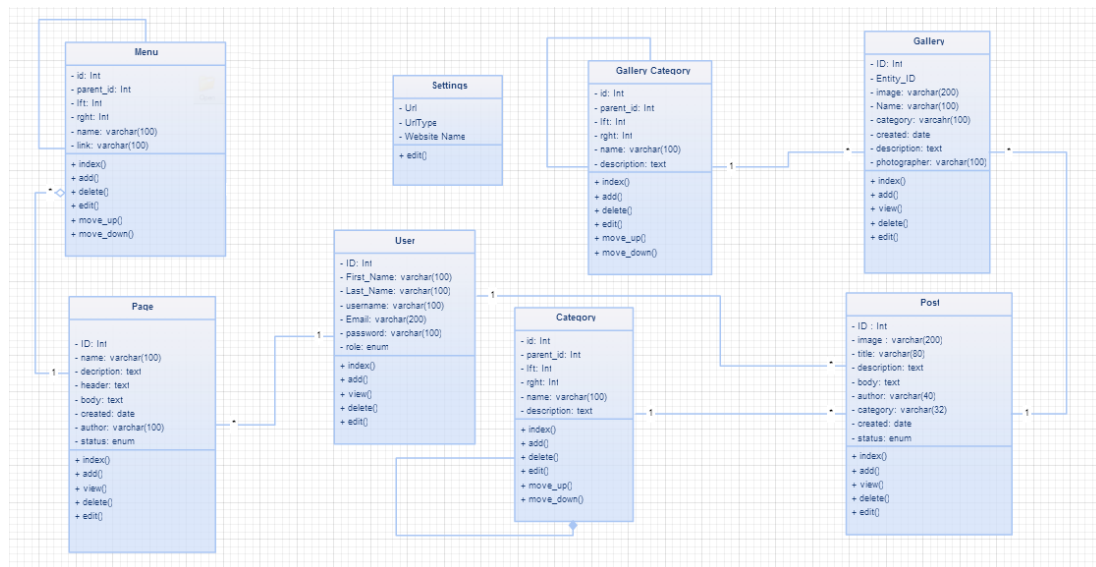
1. Ο χρήστης συνδέεται στο σύστημα.
2. Ο χρήστης επιλέγει την οντότητα στην οποία θα βάλει περιεχόμενο ανάμεσα σε Articles , Galleries , Categories , Products , Rooms.
3. Το σύστημα επιστρέφει τις πληροφορίες και τα περιεχόμενα στον χρήστη ανάλογα με την οντότητα που επέλεξε.
4. Ο χρήστης επιλέγει την λειτουργία πάνω στην οντότητα που επέλεξε ανάμεσα σε (Add , Edit , Delete) .
5. Το σύστημα απαντάει στον χρήστη την κατάλληλη φόρμα στοιχείων.
6. Ο χρήστης τοποθετεί τα στοιχεία που θέλει να προσθέσει ή να τροποποιήσει.
7. Το σύστημα εκτελεί έναν έλεγχο ορθότητας των δεδομένων.
8. Το σύστημα ολοκληρώνει την διαδικασία και επιστρέφει ένα μήνυμα στον χρήστη.
9. Ο χρήστης αποσυνδέεται από το σύστημα.

Περίπτωση Χρήσης: Ο χρήστης επεκτείνει το περιεχόμενο της Ιστοσελίδας.

1. Ο χρήστης συνδέεται στο σύστημα.
2. Ο χρήστης επιλέγει την οντότητα στην οποία θα βάλει περιεχόμενο ανάμεσα σε Pages , Menu , Users.
3. Το σύστημα επιστρέφει τις πληροφορίες και τα περιεχόμενα στον χρήστη ανάλογα με την οντότητα που επέλεξε.
4. Ο χρήστης επιλέγει την λειτουργία πάνω στην οντότητα που επέλεξε ανάμεσα σε (Add , Edit , Delete) .
5. Το σύστημα απαντάει στον χρήστη την κατάλληλη φόρμα στοιχείων.
6. Ο χρήστης τοποθετεί τα στοιχεία που θέλει να προσθέσει ή να τροποποιήσει.
7. Το σύστημα εκτελεί έναν έλεγχο ορθότητας των δεδομένων και κατασκευάζει εκείνη την στιγμή τα καινούργια αρχεία rhp.
8. Το σύστημα ολοκληρώνει την διαδικασία και επιστρέφει ένα μήνυμα στον χρήστη.
9. Ο χρήστης αποσυνδέεται από το σύστημα.

Μοντέλο εφαρμογής (Model)

Class Diagram:



Εικόνα 2 Διάγραμμα οντοτήτων

Το μοντέλο του πυρήνα της εφαρμογής έχει 7 βασικές οντότητες και 1 βοηθητική. Ο όρος βοηθητική σημαίνει ότι η συγκεκριμένη οντότητα δεν συμμετέχει άμεσα στην λειτουργία του συστήματος.

Οι 7 βασικές οντότητες:

User.

Περιγραφή.

Η οντότητα User (Χρήστης) περιλαμβάνει όλα εκείνα τα χαρακτηριστικά που περιγράφουν ένα χρήστη του συστήματος. Είναι θεμελιώδες οντότητα για την σωστή λειτουργία της εφαρμογής. Κάθε αλληλεπίδραση με το σύστημα θα πρέπει να έχει ταυτότητα.

Τα πεδία.

ID, First_Name, Last_Name, username, Email, password και role. Τα πεδία First_Name και Last_Name υπάρχουν μόνο για λόγους εμπλουτισμού του συστήματος με πληροφορίες ελέγχου. Σε αντίθεση τα πεδία username, password και role είναι υπεύθυνα για πιο σύνθετες λειτουργίες και

εμπλέκονται άμεσα με λειτουργίες όπως Σύνδεση και αποσύνδεση στο σύστημα καθώς και διαχείριση των υπόλοιπων οντοτήτων.

Οι περιορισμοί.

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε χρήστη. (Not null). Είναι Integer μέχρι 10 χαρακτήρων

First_Name , Last_Name. Είναι πεδία χαρακτήρων 100 χαρακτήρες μέγιστο.

Username. Πρέπει να είναι μοναδικό για κάθε χρήστη 100 χαρακτήρες μέγιστο. (Not null).

Password. Είναι αποθηκευμένο σε μορφή MD5. Είναι κρυπτογραφημένο . (Not null).

Role. Στο πεδίο Role υπάρχουν 2 βασικές κατηγορίες Admin και author. (Not null).

Email. Στο πεδίο email πρέπει να είναι αποθηκευμένη πληροφορία που να έχει την μορφή διεύθυνσης ηλεκτρονικού ταχυδρομείου.

Post.

Περιγραφή:

Η οντότητα Post, (άρθρο) είναι ίσως η βάση του συστήματος διαχείρισης περιεχομένου. Με αυτήν την οντότητα καλούμαστε να παρουσιάσουμε το μεγαλύτερο όγκο πληροφοριών που θα διαχειρίζεται το σύστημα μας. Ουσιαστικά ένα σύστημα διαχείρισης περιεχομένου έχει ως βασική του λειτουργία να δημιουργεί , να διορθώνει , να διαγράφει , να ταξινομεί και να προβάλει άρθρα.

Τα πεδία:

ID, image, title, description, body, author, category, created, status, tags. Τα βασικά πεδία ID και title είναι υπεύθυνα για την ταυτοποίηση του κάθε άρθρου (Post). Τα πεδία author και category ουσιαστικά λειτουργούν σαν ξένα κλειδιά μέσα στην οντότητα. Το πεδίο category είναι πολύ σημαντικό για να επιτευχθεί ένα βασικός μηχανισμός ταξινόμησης των άρθρων (Post). Τα description και body είναι το κύριο μέρος του άρθρου (Post) που περιέχει και την πληροφορία, μια βασική διάκριση είναι ότι το πεδίο description επιτρέπεται

να περιέχει μόνο χαρακτήρες ενώ σε αντίθεση το πεδίο body μπορεί να περιέχει ένα εύρος διαφορετικού είδους πληροφορία όπως φωτογραφίες , html tags και οτιδήποτε άλλο μπορεί να περιέχει ένα HTML αρχείο. Συνεχίζοντας τα πεδία created και status που πλαισιώνουν την οντότητα έχουν ρόλο ρυθμιστικό και ελέγχου απευθύνονται κυρίως στον διαχειριστή του συστήματος. Στο πεδίο image αποθηκεύουμε την διαδρομή που βρίσκετε η κύρια εικόνα του άρθρου (Post). Είναι καλό κάθε άρθρο (Post) να συνοδεύεται με μία εικόνα δεν είναι όμως και αναγκαίο. Τέλος το πεδίο tags έχει ως λειτουργία να αποθηκεύει λέξεις οι οποίες είναι σχετικές με το άρθρο (Post) με σκοπό να υπάρχουν έξυπνοι μηχανισμοί αναζήτησης.

Οι περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε άρθρο. (Not null). Είναι Integer μέχρι 10 χαρακτήρων

image. Είναι πεδίο χαρακτήρων, 250 χαρακτήρες μέγιστο.

title. Είναι πεδίο χαρακτήρων, 250 χαρακτήρες μέγιστο. (Not null).

description. Είναι πεδίο κειμένου text.

body. Είναι πεδίο κειμένου text.

Author. Είναι πεδίο χαρακτήρων , 100 χαρακτήρες μέγιστο. (Not null).

Category. Είναι πεδίο χαρακτήρων , 100 χαρακτήρες μέγιστο. (Not null).

Created. Είναι πεδίο ημερομηνίας (date). (Not null).

Status. Είναι πεδίο επιλογής (enum) οι 2 επιλογές είναι "Published" και "hide". Έχει ως προεπιλογή το "published".

Tags. Είναι πεδίο χαρακτήρων, 250 χαρακτήρες μέγιστο.

Category.

Περιγραφή:

Η οντότητα category είναι υπεύθυνη να φτιάξει σύνολα από άρθρα (Post) είναι συνοδευτική οντότητα και από μόνη της δεν έχει νόημα. Κάθε κατηγορία (Category) μπορεί να περιέχει απεριόριστα άρθρα (Post). Είναι χρήσιμο να σημειώσουμε ότι κάθε κατηγορία (Category) μπορεί να περιέχει και άλλες

κατηγορίες (Category) , υποκατηγορίες δηλαδή. Η πληροφορία της συγκεκριμένης οντότητας πρέπει να υπακούει στους κανόνες της δενδρικής δομής δεδομένων. Κάθε κατηγορία (Category) που περιέχει υποκατηγορίες (Category) δεν μπορεί ταυτόχρονα να περιέχει και άρθρα (Post). Η οντότητα του άρθρου (Post) παίζει τον ρόλο του φύλλου (leaf) στην δενδρική δομή δεδομένων Κατηγορία (Category).

Τα πεδία:

ID, parent_id , lft , rght, name, description. Το πεδίο ID και name είναι χαρακτηριστικό ταυτοποίησης της κάθε κατηγορίας. Τα πεδία parent_id, lft ,rght είναι τα πεδία που αναπαριστούν μία δενδρική δομή. Στο πεδίο parent_id τοποθετείται το ID της κατηγορίας «πάτερα», το συγκεκριμένο πεδίο μπορεί να είναι και null αν πρόκειται για κατηγορία ρίζα. Τα πεδία lft και rght χρησιμεύουν στην ταξινόμηση των κατηγοριών μέσα στην δενδρική δομή, οι τιμές είναι τα ID των κατηγοριών που βρίσκονται πριν και μετά. Το πεδίο description είναι βοηθητικό και απλό περιγράφει το περιεχόμενο κάθε κατηγορίας.

Περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε κατηγορία. (Not null).

parent_id. Είναι Integer μέχρι 10 χαρακτήρων.

Lft. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Rght. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Name. Είναι πεδίο χαρακτήρων (Varchar), 100 χαρακτήρες μέγιστο. (Not null).

Description. Είναι πεδίο κειμένου text.

Gallery Category.

Περιγραφή:

Η οντότητα Gallery Category είναι υπεύθυνη να φτιάξει σύνολα από Images φωτογραφίες είναι συνοδευτική οντότητα και από μόνη της δεν έχει νόημα. Κάθε κατηγορία (Category) μπορεί να περιέχει απεριόριστα σε αριθμό Images φωτογραφίες. Είναι χρήσιμο να σημειώσουμε ότι κάθε Gallery Category

μπορεί να περιέχει και άλλες Gallery Category, υποκατηγορίες δηλαδή. Η πληροφορία της συγκεκριμένης οντότητας πρέπει να υπακούει στους κανόνες της δενδρικής δομής δεδομένων. Κάθε Gallery Category που περιέχει υποκατηγορίες Gallery Category δεν μπορεί ταυτόχρονα να περιέχει και Images φωτογραφίες. Η οντότητα του Image φωτογραφία παίζει τον ρόλο του φύλλου (leaf) στην δενδρική δομή δεδομένων Gallery Category.

Τα πεδία:

ID, entity_id, parent_id, lft, right, name, description. Το πεδίο ID και name είναι χαρακτηριστικό ταυτοποίησης της κάθε κατηγορίας. Τα πεδία parent_id, lft, right είναι τα πεδία που αναπαριστούν μία δενδρική δομή. Στο πεδίο parent_id τοποθετείται το ID της κατηγορίας «πάτερα», το συγκεκριμένο πεδίο μπορεί να είναι και null αν πρόκειται για κατηγορία ρίζα. Τα πεδία lft και right χρησιμεύουν στην ταξινόμηση των κατηγοριών μέσα στην δενδρική δομή, οι τιμές είναι τα ID των κατηγοριών που βρίσκονται πριν και μετά. Το πεδίο description είναι βοηθητικό και απλό περιγράφει το περιεχόμενο κάθε κατηγορίας.

Το πεδίο entity_id συνδέει μία κατηγορία με Galleries με ένα άρθρο με μία οντότητα τύπου article. Αυτό είναι εξαιρετικά χρήσιμο όταν θέλουμε να δημιουργήσουμε πλούσιο περιεχόμενο σε μία ιστοσελίδα.

Περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment, είναι μοναδικό για κάθε κατηγορία. (Not null).

Entity_id Είναι Integer μέχρι 10 χαρακτήρων.

parent_id. Είναι Integer μέχρι 10 χαρακτήρων.

Lft. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Rght. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Name. Είναι πεδίο χαρακτήρων (Varchar), 100 χαρακτήρες μέγιστο. (Not null).

Description. Είναι πεδίο κειμένου text.

Gallery

Περιγραφή:

Η οντότητα Gallery έχει ως σκοπό την αποθήκευση και ταξινόμηση φωτογραφιών στο σύστημα. Κάθε φωτογραφία που θα αποθηκεύεται στον διακομιστή θα πρέπει να πλαισιώνεται από πληροφορίες οι οποίες θα αποθηκεύονται στα δεδομένα του συστήματος μας με σκοπό να μην παρέχουμε στον χρήστη απλώς ένα αρχείο εικόνας αλλά μία οντότητα. Οι συνοδευτικές πληροφορίες χρησιμοποιούνται και για την βελτιστοποίηση του περιεχομένου στις μηχανές αναζήτησης. (SEO Friendly). Ένα δεύτερο στοιχείο που πρέπει να προσέξουμε είναι ότι τα δεδομένα θα χρησιμοποιηθούν και με τρόπο τέτοιο ώστε ο παραγόμενος κώδικας να είναι φιλικός σε άτομα με ειδικές ανάγκες (AMEA).

Τα πεδία:

ID, category, image, title, description, photographer, status, created. Το πεδίο ID και title είναι χαρακτηριστικό ταυτοποίησης της κάθε οντότητας τύπου Gallery. Το πεδίο category περιέχει το ID μίας οντότητας τύπου Gallery_Category, της κατηγορίας δηλαδή, με αυτόν τον τρόπο ομαδοποιούμε της εγγραφές Gallery. Το πεδίο image περιέχει το Path του αρχείου την θέση δηλαδή της κάθε φωτογραφίας. Τα πεδία description, και photographer παίζουν ρόλο πληροφοριών πλαισίωσης και εμπλουτισμού είναι επίσης εκείνα τα πεδία που θα χρησιμοποιήσουμε για να δομήσουμε περιεχόμενο SEO Friendly και AMEA. Τα πεδία created και status που πλαισιώνουν την οντότητα έχουν ρόλο ρυθμιστικό και ελέγχου απευθύνονται κυρίως στον διαχειριστή του συστήματος ορίζουν το αν μία εικόνα είναι ενεργή ή όχι, το created ορίζει την ημερομηνία δημιουργίας.

Περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε κατηγορία. (Not null).

Category. Είναι πεδίο χαρακτήρων , 100 χαρακτήρες μέγιστο. (Not null).

Image. Είναι πεδίο χαρακτήρων, 500 χαρακτήρες μέγιστο. (Not null).

title. Είναι πεδίο χαρακτήρων, 250 χαρακτήρες μέγιστο. (Not null).

description. Είναι πεδίο κειμένου text.

Photographer. Είναι πεδίο χαρακτήρων , 100 χαρακτήρες μέγιστο. (Not null).

Created. Είναι πεδίο ημερομηνίας (date). (Not null).

Status. Είναι πεδίο επιλογής (enum) οι 2 επιλογές είναι “Published” και “hide”. Έχει ως προεπιλογή το “published”.

Page

Περιγραφή:

Η οντότητα Page ή «Σελίδα» είναι κεντρικής σημασίας για το σύνολο του συστήματος. Με αυτή την οντότητα ουσιαστικά δημιουργούμε τον σχεδιασμό της και την δομή της ιστοσελίδας (Front Office). Κάθε εγγραφή Page αντιστοιχεί σε 2 αρχεία .php που τοποθετούνται σε συγκεκριμένα σημεία. Θα αναλύσουμε την παραπάνω λειτουργία στο κεφάλαιο «Δομή του συστήματος αρχείων». Η οντότητα Page συνοψίζει ουσιαστικά όλες τις προβολές (Views) της ιστοσελίδας που θέλει να δημιουργήσει ο χρήστης του συστήματος (Front Office). Είναι σημαντικό σε αυτό το σημείο να ξεκαθαρίσουμε το γεγονός ότι μέσα στις “Σελίδες” συναντάμε κώδικα html , php και σε κάποιες περιπτώσεις εντολές SQL. Μπορεί να υπάρξει και περιεχόμενο μέσα στην “Σελίδα” όπως κείμενα και εικόνες ανάμεσα από τον κώδικα αλλά το γεγονός αυτό κάνει το περιεχόμενο πιο «στατικό», όχι απόλυτα στατικό καθώς οι «Σελίδες» μπορούν να τροποποιηθούν όσο η ιστοσελίδα βρίσκεται σε λειτουργία αλλά είναι γεγονός ότι ο χρήστης θα πρέπει να έχει στοιχειώδεις γνώσεις προγραμματισμού για να διαχειριστεί αυτό το περιεχόμενο. Η οντότητα Page αποτελείτε επίσης και από συνοδευτικές πληροφορίες ελέγχου που διευκολύνουν την διαχείριση του κώδικα μεγάλων ιστοσελίδων. Βασικές πληροφορίες είναι η ημερομηνία δημιουργίας και ο συντάκτης της «Σελίδας» με αυτές τις δύο βασικές πληροφορίες μπορούμε να δομήσουμε και ένα υποτυπώδες σύστημα διαχείρισης κώδικα από πολλούς συντάκτες.

Τα πεδία:

ID, name, description, body, head, status, author, created. Το ID είναι πεδίο ταυτοποίησης όπως και το name. Το πεδίο description περιέχει μια συνοπτική

περιγραφή για την «Σελίδα» δεν εμφανίζετε στην ιστοσελίδα η χρήση του είναι αποκλειστικά για τους διαχειριστές τους συστήματος. Τα πεδία head και body περιέχουν κώδικα .php, html , SQL και είναι υπεύθυνα για την κατασκευή του Front Office είναι πολύ σημαντικό να δηλώσουμε ότι ο το περιεχόμενο των δυο αυτών πεδίων «τυπώνεται» σε δύο αρχεία .php. και δεν αποθηκεύετε στην βάση δεδομένων. Τα πεδίο status ενεργοποιεί ή απενεργοποιεί την εκάστοτε «Σελίδα». Τα πεδία author και created λειτουργούν με σκοπό να διαχειριστούμε κάθε «Σελίδα» από την σκοπιά του προγραμματιστή.

Περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε κατηγορία. (Not null).

name. Είναι πεδίο χαρακτήρων, 250 χαρακτήρες μέγιστο. (Not null). είναι μοναδικό για κάθε «Σελίδα».

description. Είναι πεδίο κειμένου text.

author. Είναι πεδίο χαρακτήρων , 100 χαρακτήρες μέγιστο. (Not null).

Created. Είναι πεδίο ημερομηνίας (date). (Not null).

Status. Είναι πεδίο επιλογής (enum) οι 2 επιλογές είναι “Published” και “hide”. Έχει ως προεπιλογή το “published”.

Τα πεδία body Και head δεν αποθηκεύουν το περιεχόμενο τους στη βάση δεδομένων του συστήματος.

Menu.

Περιγραφή:

Η οντότητα Menu περιγράφει την σύνθεση του βασικού μενού που έχει κάθε ιστοσελίδα. Το Menu είναι δεντρική δομή και αποτελείται από κόμβους «παιδιά» και από κόμβους τύπου «πατέρα». Κάθε κόμβος μπορεί να έχει πολλά «παιδιά» αλλά μόνο έναν «πατέρα». Κάθε κόμβος συνδέεται με κάποια οντότητα τύπου “Page”. Το Menu ουσιαστικά είναι ο βασικός τρόπος που μπορεί ο εκάστοτε επισκέπτης να περιηγηθεί στις «Σελίδες».

Τα πεδία:

ID, page_id, parent_id , lft , rght, name, description. Το πεδίο ID και name είναι χαρακτηριστικό ταυτοποίησης της κάθε κατηγορίας. Τα πεδία parent_id, lft ,rght είναι τα πεδία που αναπαριστούν μία δενδρική δομή. Στο πεδίο

parent_id τοποθετείται το ID της κατηγορίας «πάτερα», το συγκεκριμένο πεδίο μπορεί να είναι και null αν πρόκειται για κατηγορία ρίζα. Τα πεδία lft και rght χρησιμεύουν στην ταξινόμηση των κατηγοριών μέσα στην δενδρική δομή, οι τιμές είναι τα ID των κατηγοριών που βρίσκονται πριν και μετά. Το πεδίο description είναι βοηθητικό και απλό περιγράφει το περιεχόμενο κάθε κατηγορίας.

Το πεδίο page_id συνδέει μία κατηγορία με Pages με μία Σελίδα με μία οντότητα τύπου Page.

Περιορισμοί:

ID. Είναι το primary key της οντότητας, auto increment , είναι μοναδικό για κάθε κατηγορία. (Not null).

page_id Είναι Integer μέχρι 10 χαρακτήρων.

parent_id. Είναι Integer μέχρι 10 χαρακτήρων.

Lft. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Rght. Είναι Integer μέχρι 10 χαρακτήρων. (not null).

Name. Είναι πεδίο χαρακτήρων (Varchar), 100 χαρακτήρες μέγιστο. (Not null).

Description. Είναι πεδίο κειμένου text.

```
k?php
class User extends AppModel {

    var $name = 'User';

    public $validate = array(
        'username' => array(
            'required' => array(
                'rule' => array('notEmpty'),
                'message' => 'A username is required'
            )
        ),
        'password' => array(
            'required' => array(
                'rule' => array('notEmpty'),
                'message' => 'A password is required'
            )
        ),
        'role' => array(
            'valid' => array(
                'rule' => array('inList', array('admin', 'author')),
                'message' => 'Please enter a valid role',
                'allowEmpty' => false
            )
        )
    );
};
```

Εικόνα 3 Κώδικας Μοντέλου User

Controller Εφαρμογής - Λειτουργικότητα.

Ο Controller της εφαρμογής είναι ουσιαστικά το κομμάτι εκείνο που θα ενώσει τις οντότητες μας με την διεπαφή. Επίσης εδώ περιέχονται όλες οι λειτουργίες που θα εκτελεστούν. Βέβαια είναι σκόπιμο να αναφέρουμε ότι ένα Framework σας το Cake PHP υλοποιεί αρκετά κομμάτια στον πυρήνα του όπως είναι οι μέθοδοι προσπέλασης της βάσης δεδομένων καθώς και ο έλεγχος ορθότητας των δεδομένων της εφαρμογής.

Για να κατανοήσουμε καλλίτερα τον τρόπο με τον οποίο λειτουργεί ο controller πρέπει να αναφέρουμε μερικές δεδομένες διαδικασίες που πρέπει να εκτελούμε με κάθε οντότητα που χρησιμοποιούμε. Κάθε οντότητα που αποτελείτε από ένα αρχείο model που περιγράφονται οι περιορισμοί καθώς και εάν υπάρχει κάποια διαφορά στον τρόπο που συμπεριφέρεται η οντότητα. Μία τέτοια διαφορά συναντάμε στις οντότητες που για την ευχρηστία του συστήματος πρέπει να τις θεωρήσουμε ως δεντρικές δομές. Για να γίνει κάτι τέτοιο εφικτό εκτός από τον σχεδιασμό της βάσης αυτής κάθε αυτής πρέπει να τροποποιήσουμε και το model κατάλληλα. Η αναφορά μας στο μοντέλο σε αυτό το σημείο είναι σκόπιμη καθώς το framework συνδέει κάθε οντότητα μας με ένα model και κάθε model αρχείο με το αντίστοιχο controller του.

Μία δεύτερη πρόταση που πρέπει να τοποθετήσουμε στα υπ' όψιν μας είναι ότι σε κάθε οντότητα θα υπάρχουν μέσα στον controller τουλάχιστον οι 3 βασικές μέθοδοι διαχείρισης `index()` , `add()` , `edit()` , `delete()`. Κάθε μία από τις παραπάνω μεθόδους είναι προγραμματισμένες έτσι με το framework ώστε να επικοινωνούν με κάποια διεπαφή, εκτός από την `delete()` που συνήθως δεν υπάρχει αντίστοιχη διεπαφή απλώς ακολουθεί κατά το πέρασ της κάποιο μήνυμα επιτυχίας ή αποτυχίας.

Δεν είναι απαραίτητο κάθε controller να ακολουθείται από κάποια οντότητα-μοντέλο, υπάρχουν περιπτώσεις όπου κάποιες λειτουργίες δεν έχουν άμεση επαφή με τα δεδομένα μας. Τέτοιες λειτουργίες συνήθως δεν έχουν να κάνουμε με αποθήκευση και τροποποίηση στοιχείων αλλά με μεθόδους υπολογισμού. Συνήθως περίπλοκοι αλγόριθμοι αποκόπτονται από σε κάποιον ξεχωριστό controller. Αυτό δεν είναι πάντα αναγκαίο αλλά πολλές

φορές διευκολύνει το σχεδιασμό και την περαιτέρω ανάπτυξη και συντήρηση του συστήματος.

Οι Controller του Cactus.

Το Cactus αποτελείται από 7 βασικούς Controller που χειρίζονται τα Μοντέλα των οντοτήτων (models). Οι βασικοί controller του συστήματος είναι οι εξής, UsersController, PostsController, PagesController, CategoriesController, GalleriesController, McategoriesController, GcategoriesController, Παρακάτω θα αναλύσουμε την λειτουργικότητα του κάθε Controller ξεχωριστά.

Controller του Posts:

Η οντότητα των άρθρων ίσως είναι από η πιο κεντρική από όλες τις υπόλοιπες για το σύστημα μας. Η λειτουργικότητα της καθορίζεται από τις μεθόδους CRUD `index()` , `edit()` , `delete()`, `add()` δεν σταματάνε εκεί όμως οι απαιτήσεις. Για την ανάπτυξη αυτού του Controller έπρεπε να κατασκευαστεί ένα πρόσθετο το οποίο θα ανεβάζει από το υπολογιστή του εκάστοτε χρήστη στον server, και θα το ταξινομούσε κατάλληλα ένα αρχείο φωτογραφίας. Το γεγονός ότι αυτό το πρόσθετο δεν θα χρησιμοποιηθεί μόνο από αυτόν τον Controller μας οδηγούν στο γεγονός να μην τοποθετήσουμε τον κώδικα του πρόσθετου (`FileUploadComponent`) μέσα στον `PostsController` αλλά στο φάκελο `components`. Το πρόσθετο `FileUploadComponent` φορτώνεται στην αρχή του controller μας στην μέθοδο `beforeFilter()` και χρησιμοποιείται στην μέθοδο `add()`. Σε εκείνο το σημείο δίνουμε στο σύστημα όλα τα απαραίτητα πεδία που αφορούν την οντότητα όπως ο τίτλος και η περιγραφή καλούμαστε όμως να ανεβάσουμε και μία φωτογραφία αντιπροσωπευτική για το άρθρο μας. Ένα ακόμα άξιο σημασίας γεγονός είναι ότι στο `PostController` χρησιμοποιούμε την μέθοδο του `cakePHP Framework` `Paginate`. Η `Paginate` είναι μία μέθοδος η οποία παίρνει σαν είσοδο εντολές ελέγχου κάποιου πίνακα της βάσης δεδομένων και σύμφωνα με αυτές επιστρέφει τα δεδομένα μας σελιδοποιημένα. Είναι πολύ χρήσιμο όταν έχουμε πολλά δεδομένα να τα επιστρέφουμε στον χρήστη λίγα – λίγα σε σελίδες και όχι όλα μαζί. Τέλος είναι σημαντικό να αναφέρουμε και την ξεχωριστή μέθοδο που υλοποιείται μόνο στο `PostController` την `by_category()`. Η `by_category()` έχει την αρμοδιότητα να

μην εμφανίζει όλα τα άρθρα μαζί αλλά ταξινομημένα κατά κατηγορία έτσι ώστε ο χρήστης να μπορεί να βρίσκει εύκολα αυτό που ψάχνει. Ανακεφαλαιώνοντας βλέπουμε ότι ο `PostController` σε πρώτο βαθμό εκτελεί μια διαδικασία απλής σύνδεσης του μοντέλου με την διεπαφή αλλά σε δεύτερο πλάνο εκτελεί και άλλες πιο σύνθετες λειτουργίες όπως η διαδικασία `Uploading` (`FileUploadComponent`). Σε παρακάτω κεφάλαιο θα περιγράψουμε επ' ακριβώς την λειτουργία του πρόσθετου (`FileUploadComponent`).

Controller του Users:

Ο `UsersController` είναι υπεύθυνος για την διαχείριση των χρηστών του συστήματος. Είναι οι πρώτες λειτουργίες που θα βρεθούν μπροστά μας κατά την διάρκεια της εγκατάστασης και παραμετροποίησης τους συστήματος. Όπως και οι υπόλοιποι `Controller` υλοποιούνται και εδώ οι `CRUD` λειτουργίες `add()` , `view()` , `index()`, `edit()`, `delete()`. Έχουμε όμως δύο ξεχωριστές πολύ σημαντικές λειτουργίες που είναι η `login()` και η `logout()`. Εύκολα καταλαβαίνουμε ότι εκτός από το να υπάρχει κάποιος χρήστης στο σύστημα, θα πρέπει και να συνδέετε σε αυτό και να αποσυνδέετε. Η υλοποίηση των δύο αυτών μεθόδων γίνεται με την βοήθεια μια βιβλιοθήκης του `CakePHP Framework` την `Auth`. Η `Auth` έχει πρωτογενείς μεθόδους `login()` και `logout()` που κάνουν την ταυτοποίηση με τα στοιχεία του χρήστη στην βάση δεδομένων και τον τοποθετούν στο `session` ή τον αποσυνδέουν αντίστοιχα από το `session` και την εφαρμογή κατ' επέκταση. Η `Auth` βιβλιοθήκη λειτουργεί σαν αντικείμενο του πυρήνα του `Framework` και μπορούμε να την χρησιμοποιήσουμε ανά πάσα στιγμή. Είναι χρήσιμο να αναφέρουμε και την μέθοδο `allow()` και την μέθοδο `disallow()` που είναι υπεύθυνες να δίνουν πρόσβαση σε κάποιον συνδεδεμένο χρήστη η το αντίθετο. Τέλος πρέπει να αναφέρουμε ότι για λόγους αισθητικής ο `UsersController` χρησιμοποιεί δύο `Layout` ένα για την λειτουργία του `login` και το κλασικό `Layout` για τις υπόλοιπες λειτουργίες, θα αναφερθούμε όμως περισσότερο σε αυτό το ζήτημα σε επόμενο κεφάλαιο.

Controller του Pagem.

Ο PagemController είναι υπεύθυνος για την κατασκευή και την διαχείριση του Front - Office της ιστοσελίδας που καλείτε να κατασκευάσει ο χρήστης. Είναι ο Controller εκείνος που ουσιαστικά υλοποιεί όλες εκείνες τις λειτουργίες που θα αποτελούν την σχεδίαση και την δομή της παράγωγης ιστοσελίδας. Ο PagemController δεν έχει να κάνει με τα δεδομένα που προβάλλονται στην ιστοσελίδα, όπως και το μοντέλο Pagem εξ' άλλου. Ουσιαστικά το κομμάτι PagemController είναι το κομμάτι εκείνο του συστήματος που έχει ως αντικειμενικό σκοπό να επεκτείνει την παράγωγή ιστοσελίδα ανά πάσα στιγμή και καθώς αυτή θα βρίσκετε σε λειτουργία (Online). Αυτό δεν σημαίνει βέβαια ότι δεν εκτελούνται οι καθιερωμένες λειτουργίες CRUD add(), edit(), index(), delete() μόνο που σε αυτήν την περίπτωση έχουν ρόλο δευτερεύοντα και όχι πρωταρχικό. Πριν προχωρήσουμε στην ουσία του Controller να αναφέρουμε ότι η ονομασία Pagem και όχι απλός Page για να μην υπάρξει σύγχυση με τον Controller του πυρήνα του CakePHP Framework. Ο PagemController υλοποιεί την βασική του λειτουργικότητα κατά την διαδικασία κατασκευής μια καινούργιας σελίδας στο front – office, καθώς και κατά την διαδικασία τροποποίησης κάποιας είδη υπάρχουσας. Η βασική διαδικασία δηλαδή υλοποιείτε μέσα στις μεθόδους add() και edit() αντίστοιχα. Κάθε φορά που ο χρήστης θέλει να δημιουργήσει μία σελίδα ο PagemController αναλαμβάνει να δημιουργήσει ένα αρχείο τύπου php μέσα στον φάκελο Pages του Front – Office αυτό το αρχείο περιέχει τον κώδικα (HTML/ php / mysql) που τοποθετείται στην αντίστοιχη διεπαφή. Το όνομα του εκάστοτε αρχείου είναι ίδιο με τον τίτλο της σελίδας για αυτό τον λόγο και ο τίτλος της σελίδας πρέπει να είναι μοναδικός. Μεγάλη σημασία εδώ έχει και ο Author του τού Pagem καθώς μπορούμε να έχουμε πολλούς χρήστες στο ίδιο σύστημα. Τέλος να προσθέσουμε ότι για να λειτουργήσει σωστά το σύστημα πρέπει να καταργήσουμε τις default sanitize μεθόδους του CakePHP Framework για να μπορούμε να περάσουμε εντολές SQL μέσα από Form controls.

Menu - Mcategories Controller

Ο Controller που διαχειρίζεται το βασικό μενού της εφαρμογής μας για λόγους δεσμευμένων λέξεων του CakePHP Framework το ονομάζουμε McategoriesController και όχι MenuController. Ο McategoriesController αναλαμβάνει να διαχειριστή τη δενδρική δομή του βασικού μενού της εφαρμογής μας. Τονίζουμε ότι η δομή είναι δενδρική καθώς για να διαχειριστούμε μια τέτοια κατάσταση χρησιμοποιούμε κάποιες ιδιικές μεθόδους. Επίσης είναι σημαντικό ότι εκτός από της καθιερωμένες μεθόδους CRUD index(), add(), delete(), edit() έχουμε τις μεθόδους μετακίνησης κάποιου κόμβου, την up() και down(), η κάθε μία από αυτές αντίστοιχα έχει ως σκοπό να ιεραρχήσει και να βάλει σε μία συγκεκριμένη σειρά τους κόμβους που βρίσκονται στο ίδιο βάθος του δέντρου. Δεν υπάρχει περιορισμός στον βαθμό που μπορεί να φτάσει το βάθος του δέντρου. Η αλήθεια είναι όμως ότι είναι πολύ σπάνιο να συναντήσουμε βασικό μενού με περισσότερα από 3 επίπεδα ιεραρχίας των αντικειμένων. Στο σημείο αυτό είναι χρήσιμο να αναφέρουμε και την επόμενη πολύ βασική λειτουργία του Controller είναι αυτή της σύνδεσης το εκάστοτε αντικείμενου του μενού με ένα αντικείμενο τύπου Pagem μία σελίδα δηλαδή. Ένα στοιχείο του μενού μπορεί να συνδεθεί με μία και μόνο μία σελίδα - Pagem για να υπάρχει λογική στο σύστημα εκτός και αν έχουμε μία σελίδα που συμπεριφέρεται ανάλογα με τις μεταβλητές που μπορεί να βρίσκονται στο URL τις μεταβλητές GET όπως συνηθίζουμε να αναφερόμαστε σε αυτές. Για να πετύχουμε κάτι τέτοιο πρέπει να αφήσουμε την επιλογή κάποιος να θέτει και κάποιο Link σε ελεύθερο κείμενο.

CategoriesController

Ο CategoriesController είναι υπεύθυνος να υλοποιήσει μια δενδρική δομή που αναφέρετε στις κατηγορίες μέσα στις οποίες ταξινομούνται τα άρθρα μας. Τονίζουμε ότι η δομή είναι δενδρική καθώς για να διαχειριστούμε μια τέτοια κατάσταση χρησιμοποιούμε κάποιες ιδιικές μεθόδους. Επίσης είναι σημαντικό ότι εκτός από της καθιερωμένες μεθόδους CRUD index(), add(), delete(), edit() έχουμε τις μεθόδους μετακίνησης κάποιου κόμβου, την up() και down(), η κάθε μία από αυτές αντίστοιχα έχει ως σκοπό να ιεραρχήσει και να βάλει σε μία συγκεκριμένη σειρά τους κόμβους που βρίσκονται στο ίδιο βάθος του

δέντρου. Δεν υπάρχει περιορισμός στον βαθμό που μπορεί να φτάσει το βάθος του δέντρου των κατηγοριών. Οι κατηγορίες δεν συνδέονται αυτές σε αυτό το σημείο με κάποιο άρθρο, τα άρθρα συνδέονται με αυτές κατά την κατασκευή τους και λειτουργία τους.

GalleryController

Ο GalleryController διαχειρίζεται τις φωτογραφίες τις οποίες θέλουμε να ομαδοποιήσουμε μέσα σε ένα σύνολο που ονομάζουμε Gallery. Ο GalleryController εκτός από την διαδικασία και της καθιερωμένες μεθόδους CRUD `index()`, `add()`, `delete()`, `edit()` για την ανάπτυξη αυτού του Controller έπρεπε να κατασκευαστή ένα πρόσθετο το οποίο θα ανεβάζει από το υπολογιστή του εκάστοτε χρήστη στον server, και θα το ταξινομούσε κατάλληλα ένα αρχείο φωτογραφίας. Το γεγονός ότι αυτό το πρόσθετο δεν θα χρησιμοποιηθεί μόνο από αυτόν τον Controller μας οδηγούν στο γεγονός να μην τοποθετήσουμε τον κώδικα του πρόσθετου (`FileUploadComponent`) μέσα στον `PostsController` αλλά στο φάκελο `components`. Το πρόσθετο `FileUploadComponent` φορτώνεται στην αρχή του controller μας στην μέθοδο `beforeFilter()` και χρησιμοποιείται στην μέθοδο `add()`. Βασική παράμετρο που καθορίζει την λειτουργία του Component `FileUploadComponent` είναι η ύπαρξη του πεδίου `Photographer`. Το πεδίο αυτό καθορίζει ότι το Component θα χρησιμοποιηθεί από την οντότητα `Gallery` και όχι από την οντότητα του `Post`. Η παραπάνω πρόταση έχει ιδιέτερη σημασία για τον τρόπο που το σύστημα καταχωρεί στον Server τα αρχεία των φωτογραφιών.

GCategoriesController

Ο GCategoriesController είναι υπεύθυνος να υλοποιήσει μια δενδρική δομή που αναφέρετε στις κατηγορίες των Gallery μέσα στις οποίες ταξινομούνται στοιχεία Gallery ή αλλιώς οι φωτογραφίες μας. Τονίζουμε ότι η δομή είναι δενδρική καθώς για να διαχειριστούμε μια τέτοια κατάσταση χρησιμοποιούμε κάποιες ιδιές μεθόδους. Επίσης είναι σημαντικό ότι εκτός από της καθιερωμένες μεθόδους CRUD `index()`, `add()`, `delete()`, `edit()` έχουμε τις μεθόδους μετακίνησης κάποιου κόμβου, την `up()` και `down()`, η κάθε μία από αυτές αντίστοιχα έχει ως σκοπό να ιεραρχήσει και να βάλει σε μία

συγκεκριμένη σειρά τους κόμβους που βρίσκονται στο ίδιο βάθος του δέντρου. Δεν υπάρχει περιορισμός στον βαθμό που μπορεί να φτάσει το βάθος του δέντρου των κατηγοριών. Οι κατηγορίες δεν συνδέονται αυτές σε αυτό το σημείο με κάποιο Gallery, τα Gallery συνδέονται με αυτές κατά την κατασκευή τους και λειτουργία τους. Επίσης οι κατηγορίες των Gallery συνδέονται και με μία οντότητα άρθρου για να εξυπηρετήσουμε πιο περίπλοκες λειτουργίες κάποιου άρθρου που απαιτεί πολλές φωτογραφίες κάτω από την οντότητα του ταξινομημένες.

```

<?php
class PostsController extends ApplicationController {
    public $name = 'Posts';
    public $helpers = array('Html');
    public $components = array('Paginator' , 'Fileupload');

    function beforeFilter(){
        parent::beforeFilter();
        $this->Fileupload->fields = null;
    }

    public $paginate = array(
        'limit' => 10,
        'order' => array('Post.created' => 'DESC')
    );

    function index() {
        $this->loadModel('Mcategory');
        $this->loadModel('User');

        $this->set('categorylist', $this->Mcategory->find('all'));
        $this -> set('userlist', $this->User->find('all'));

        $this->Paginator->settings = $this->paginate;
        $this -> set('posts', $this->Paginator->paginate('Post'));
    }

    function view($id = null){
        $this->set('post', $this->Post->read(NULL,$id));
    }

    public function add() {
        $this->loadModel('Mcategory');                                $this->loadModel('user');
        $this -> set('categorylist', $this->Mcategory->find('all'));
        $this -> set('userlist', $this->User->find('all'));
        if ($this->request->is('post')) {
            $this->Post->create();
            $this->request->data['Post']['image'] = "Photos/".$this->Fileupload->finalFile;
            $this->Post->save($this->request->data);
            $this->Session->setFlash('Your post has been saved.');
```

Εικόνα 4 Κώδικας Controller Posts

Component FileUploadComponent.

Το πρόσθετο FileUploadComponent είναι υπεύθυνο για να μεταφέρει μία φωτογραφία κάθε φορά από τον Client στον Server. Δηλαδή από τον υπολογιστή του χρήστη στο σύστημα. Το πρόσθετο δεν ενσωματώνει την λειτουργία του σε κάποιον συγκεκριμένο Controller γιατί η συγκεκριμένη λειτουργικότητα χρησιμοποιείται από δύο διαφορετικούς Controller έτσι αν ενσωματώναμε τον κώδικα θα έπρεπε να το κάνουμε δύο φορές. Το Framework σε αυτές τις περιπτώσεις προβλέπει την χρήση των Components.

Το FileUploadComponent έχει τις εξής μεθόδους, removeFile(), showErrors(), initialize(), processFile(), checkType(), uploadedFileArrey(), multiArrayKeyExists(). Η μέθοδος showErrors() έχει καθαρά ελεγκτικό χαρακτήρα και είναι υπεύθυνη στο να μας ενημερώνει με μηνύματα αν προκύψει κάποιο λάθος κατά την διαδικασία. Η μέθοδος removeFile() βρίσκει και διαγράφει το αρχείο της φωτογραφίας από τον Server. Η μέθοδος initialize() είναι υπεύθυνη για να αρχικοποιήσει τις βασικές μεταβλητές του υποσυστήματος. Η μέθοδος startup() ελέγχει σε ποιο από τα δύο μοντέλα αναφερόμαστε το μοντέλου του άρθρου ή το μοντέλο Gallery. Σύμφωνα με το μοντέλο το πρόσθετο θα επιλέξει και το σημείο στο οποίο θα αποθηκευτεί η φωτογραφία. Έτσι δημιουργείτε μια δομή αρχείο ταξινομημένη. Η μέθοδος checkType() είναι υπεύθυνη στο να ελέγξει αν το αρχείο είναι όντος φωτογραφία και ο τύπος της είναι κάποιος από τους επιτρεπόμενους τύπους που δέχεται το σύστημα, σε αντίθετη περίπτωση το σύστημα δεν ανεβάζει το αρχείο στον Server και επιστρέφει ένα μήνυμα λάθους.

Ο παραπάνω έλεγχος προσδίδει ένα επίπεδο ασφάλειας στο σύστημα καθώς πολλές κακόβουλες επιθέσεις χρησιμοποιούν τέτοιου είδους κενά ασφάλειας για να εκμεταλευτούν τους πόρους τους εκάστοτε διακομιστή με αποτέλεσμα να δημιουργούν προβλήματα λειτουργίας. Η μέθοδος checkFile() έχει ως λειτουργία τον έλεγχο για την ύπαρξη του αρχείου. Η processFile() είναι η κεντρική μέθοδος του πρόσθετου και καλείτε στην Startup() αν και μόνο αν περάσουμε από του ελέγχους των μεθόδων checkType() και checkFile(). Η processFile() αρχικά ορίζει κάποιες μεταβλητές και στην συνέχεια ελέγχει αν το αρχείο που επιχειρούμε να μεταφέρουμε υπάρχει είδη στον Server, σε

περίπτωση που υπάρχει μετονομάζει το αρχείο προσθέτοντας έναν αριθμό στο τέλος του και ξανά ελέγχει, αν ξανά βρει το ίδιο όνομα ξανά προσθέτει και τοποθετεί νέο αριθμό μέχρι να δημιουργήσει ένα όνομα που δεν υπάρχει. Στην συνέχεια το πρόσθετο αναλαμβάνει να μεταφορτώσει το αρχείο και επιστρέφει ένα μήνυμα επιτυχίας.

Τέλος το πρόσθετο τερματίζει την λειτουργία του επιστέφοντας στον Controller από τον οποίο κλήθηκε το όνομα και την τοποθεσία αποθήκευσης.

Δομή αρχείων συστήματος Cactus CMS.

Στο σημείο αυτό νομίζω είναι σκόπιμο να αναλύσουμε την συνολική δομή του συστήματος διαχείρισης αλλά και να εξηγήσουμε τον διαχωρισμό Front Office με το Back Office. Να εξηγήσουμε δηλαδή για ποιόν λόγο πρώτη φορά σε σύστημα διαχείρισης περιεχομένου το Front Office δηλαδή η ιστοσελίδα που κατασκευάζουμε είναι εκτός του CakePHP Framework.

Η βασική δομή του Cactus CMS αποτελείται από τους φακέλους Admin, Ajax, Menus, Pages, Photos και τα αρχεία index.php, .htaccess. Στο φάκελο Admin βρίσκετε όλο το Back Office δηλαδή όλα τα αρχεία του CakePHP Framework μαζί με όλα τα αρχεία του μοντέλου, του controller και της διεπαφής του συστήματος διαχείρισης περιεχομένου. Στο φάκελο Ajax βρίσκετε το αρχείο DBcon.php το οποίο είναι υπεύθυνο για την σύνδεση του Front Office με την βάση δεδομένων του συστήματος. Επίσης στο φάκελο Ajax βρίσκονται όλα τα αρχεία εκείνα που λειτουργούν με την τεχνολογία τύπου Ajax δηλαδή όλα εκείνα τα σενάρια που εκτελούνται στον server και επιστρέφουν μέσω της JavaScript μόνο τα δεδομένα σε ένα συγκεκριμένο σημείο μέσα στην σελίδα χωρίς να απαιτείτε ανανέωση. Στο φάκελο Menus υπάρχει το βασικό αρχείο MainMenu.php το οποίο είναι υπεύθυνο να διαβάζει αναδρομικά την δενδρική δομή του κεντρικού μενού από την βάση δεδομένων και να την εμφανίζει στο Front Office. Στο φάκελο Pages τοποθετούνται όλες οι σελίδες .php που αποτελούν την ιστοσελίδα και παράγονται από τον controller Pages του συστήματος διαχείρισης. Τα Pages έχουν μοναδικό όνομα και συνήθως

συνδέονται με το κεντρικό της εφαρμογής - ιστοσελίδας μας Menu. Στο φάκελο photos τοποθετούνται όλες οι φωτογραφίες που έχουν να κάνουν με το σύστημα. Μέσα στο φάκελο Photos τοποθετούνται όλα τα αρχεία φωτογραφιών που προκύπτουν από τους controller Post και Gallery. Υπάρχουν δύο φάκελοι μέσα στο φάκελο Photo ο Galleries και ο Post επίσης εκεί τοποθετούνται και άλλες δομές φακέλων εάν αυτό χαρακτηριστεί χρήσιμο. Το αρχείο index.php είναι το βασικότερο αρχείο της εφαρμογής καθώς είναι το πρώτο αρχείο που εκτελείται στο server όταν κάποιος επισκεφτεί την ιστοσελίδα και χωρίζεται σε τρεις βασικές λειτουργίες, τη λειτουργία της εκχώρησης του κεντρικού μενού, τη λειτουργία εναλλαγής σελίδων-pages και την κλήση όλων των πρόσθετων αρχείων .js και .css. Η εναλλαγή των σελίδων μέσα στον index.php γίνεται με τον εξής τρόπο: καλείται η σελίδα εκείνη που το όνομά της βρίσκεται μέσα στη παράμετρο p του URL. Σε περίπτωση που δεν υπάρχει η παράμετρος p, καλείται η αρχική σελίδα-home εάν αυτή υπάρχει. Να σημειώσουμε ότι το URL τις περισσότερες φορές διαμορφώνεται από το κεντρικό μενού. Το αρχείο .htaccess είναι υπεύθυνο για κάποιες συνοδευτικές λειτουργίες όπως η διαμόρφωση των URLs, διαχείριση των λαθών απόκρισης του server 404 και 500.

Στο σημείο αυτό προκύπτει το ερώτημα για ποιο λόγο το front office έχει αποκοπεί από το back office. Από το ερώτημα αυτό προκύπτει και το αντικειμενικό πλεονέκτημα του συστήματος που δίνει τη δυνατότητα και σε αρχάριους προγραμματιστές να γράψουν τον κώδικα τους μέσω του cms και αυτός να τοποθετηθεί άμεσα μέσα στη δομή των φακέλων. Ουσιαστικά ο online editor που περιγράψαμε στον controller του pages δεν θα μπορούσε να "τυπώσει" τον κώδικα μέσα στα αρχεία του framework καθώς αυτό θα παρέβαινε βασικές αρχές προγραμματισμού. Με το γεγονός ότι διαχωρίσαμε το front office με το back office, δίνει τη δυνατότητα στον απλό χρήστη να γράφει κώδικα online ο οποίος δεν είναι MVC.

Πτυχιακή εργασία του φοιτητή Ανέστη Σταμάτης

Filename	Filesize	Filetype	Last modified	Per...	Owner/Gr...
Ajax		Φάκελος αρχεί...	21/8/2014 3:44:06 μμ	0755	4345 4338
cgi-bin		Φάκελος αρχεί...	10/5/2014 3:39:37 μμ	0755	4345 4338
colors		Φάκελος αρχεί...	21/8/2014 3:44:07 μμ	0755	4345 4338
css		Φάκελος αρχεί...	8/9/2014 10:22:45 πμ	0755	4345 4338
fonts		Φάκελος αρχεί...	21/8/2014 3:44:07 μμ	0755	4345 4338
gr		Φάκελος αρχεί...	13/10/2014 12:58:13...	0755	4345 4338
images		Φάκελος αρχεί...	8/9/2014 1:03:28 μμ	0755	4345 4338
js		Φάκελος αρχεί...	13/10/2014 11:51:18...	0755	4345 4338
Menus		Φάκελος αρχεί...	8/9/2014 12:50:13 μμ	0755	4345 4338
Pages		Φάκελος αρχεί...	11/1/2015 12:03:28 ...	0755	4345 4338
Photos		Φάκελος αρχεί...	27/9/2014 7:57:13 μμ	0755	4345 4338
.htaccess	642	Αρχείο HTACC...	27/8/2014 5:25:47 μμ	0644	4345 4338
error_log	52.816	Αρχείο	9/1/2015 2:21:06 πμ	0644	4345 4338
index.php	4.069	Αρχείο PHP	13/10/2014 10:40:16...	0644	4345 4338
style.css	45.518	Έγγραφο Casc...	28/9/2014 3:01:09 μμ	0644	4345 4338

Εικόνα 5 Δομή αρχείων

Κεφάλαιο 4. Πρόσθετες λειτουργίες του συστήματος

Πρόσθετο ηλεκτρονικού καταστήματος.

Για την ανάγκη υλοποίησης του συγκεκριμένου πρόσθετου χρειαζόμαστε νέα model που είναι τα εξής: products, product categories, orders, customer, order details. Το κάθε μοντέλο από αυτά συνοδεύεται από το αντίστοιχο controller που δίνει τις βασικές λειτουργίες CRUD. Ουσιαστικά ο μεγάλος φόρτος υλοποίησης του συγκεκριμένου πρόσθετου βρίσκεται στα front office αρχεία. Εκεί έχουν υλοποιηθεί τα αρχεία cart.php, order.php και orderdone.php. το αρχείο cart αναλαμβάνει να συλλέξει και να αποθηκεύσει στο session τις επιλογές προϊόντων του εκάστοτε επισκέπτη. Το αρχείο order αναλαμβάνει να συλλέξει την πληροφορία από το session για τα προϊόντα και ταυτόχρονα να συλλέξει τα στοιχεία του επισκέπτη-πελάτη. Τέλος αποστέλλει όλα τα παραπάνω στο αρχείο orderdone, το οποίο ελέγχει για την ορθότητα των δεδομένων, τα αποθηκεύει εάν αυτά είναι σωστά και επιστρέφει ένα μήνυμα επιτυχίας. Η υλοποίηση του πρόσθετου μπορεί να επεκταθεί και να τροποποιηθεί σύμφωνα με τις ανάγκες του εκάστοτε ηλεκτρονικού καταστήματος.

Πρόσθετο ηλεκτρονικών κρατήσεων ξενοδοχείων .

Στο συγκεκριμένο πρόσθετο έχουμε τα εξής μοντέλα: rooms, customers, availability, reservations και detail reservation. Κάθε ένα από τα μοντέλα αυτά συνοδεύεται από τον αντίστοιχο controller που υλοποιεί τις βασικές λειτουργίες CRUD. Ουσιαστικά ο μεγάλος φόρτος υλοποίησης του συγκεκριμένου πρόσθετου βρίσκεται στα front office αρχεία. Εκεί έχουν υλοποιηθεί το checkavailability.php το οποίο ελέγχει για τη διαθεσιμότητα του εκάστοτε δωματίου, το αρχείο reservation.php που αναλαμβάνει να τοποθετήσει στο session τις επιλογές του επισκέπτη-πελάτη καθώς και τα στοιχεία του. Τέλος τα προωθεί στο αρχείο reservationdone.php το οποίο κάνει έλεγχο της ορθότητας των δεδομένων, τα αποθηκεύει εάν αυτά είναι σωστά και επιστρέφει ένα μήνυμα επιτυχίας.

Interface - διεπαφή εφαρμογής.


Η διεπαφή του συστήματος διαχείρισης περιεχομένου Cactus CMS χωρίζεται σε τρία βασικά επίπεδα: το επίπεδο του κεντρικού μενού στο οποίο ο χρήστης επιλέγει τη λειτουργία που θα χρησιμοποιήσει. Το επίπεδο της λειτουργικότητας όπου είναι το σημείο εκείνο που εμφανίζονται όλες οι λειτουργίες που επέλεξε ο χρήστης από το κεντρικό μενού. Στο σημείο αυτό της κεντρικής λειτουργικότητας εμφανίζονται οι φόρμες καταχώρησης και διόρθωσης καθώς και οι πίνακες που παρουσιάζουν στον χρήστη με έναν απλό και ξεκούραστο τρόπο τα δεδομένα της εφαρμογής. Το τρίτο επίπεδο είναι η μπάρα ελέγχου που βρίσκεται στη κορυφή της εφαρμογής όπου εμφανίζεται το όνομα του χρήστη που συνδέθηκε στο σύστημα, καθώς δίνονται και κάποιες λειτουργίες όπως log out.

Κάθε οντότητα παρουσιάζεται στο view με τρεις βασικές διεπαφές: index.ctp, edit.ctp και add.ctp. Η διεπαφή index, σε όλους τους controller εκτός από τον controller gallery, παρουσιάζει στον χρήστη έναν πίνακα (html table) όπου σε κάθε σειρά εμφανίζονται ταξινομημένα τα δεδομένα της εκάστοτε οντότητας και δίνονται για κάθε γραμμή του εκάστοτε πίνακα οι λειτουργίες διαγραφής και διόρθωσης. Όπως γίνεται εύκολα αντιληπτό η λειτουργία της διαγραφής δεν συνοδεύεται από κάποια διεπαφή. Η λειτουργία της διόρθωσης (edit) παρουσιάζει κάθε φορά το αντίστοιχο αρχείο edit.ctp της οντότητας, εμπλουτισμένο με τα κατάλληλα δεδομένα. Η διεπαφή της διόρθωσης σε όλους τους controller έχει την ίδια βασική δομή και αυτή είναι μια φόρμα όπου περιέχονται πεδία input και text editors. Ο βασικός editor που χρησιμοποιούμε σε κάθε διεπαφή είναι ο ck editor. Η διεπαφή add έχει σε όλους τους controller την ίδια βασική δομή και αυτή είναι μια φόρμα όπου περιέχονται πεδία input και text editors. Ο βασικός editor που χρησιμοποιούμε σε κάθε διεπαφή είναι ο ck editor. Στο σημείο αυτό πρέπει να τονίσουμε ότι όλες μας οι διεπαφές ερχόμενες σε συνεργασία με το μοντέλο της εφαρμογής εμφανίζουν κατάλληλα μηνύματα λάθους όπου τα δεδομένα δεν συνάδουν με τους εκάστοτε περιορισμούς.

Η οντότητα των Users είναι μια εξαίρεση καθώς έχει μία ακόμα διεπαφή αυτήν του Login. Η διεπαφή του login έχει την ιδιαιτερότητα να χρησιμοποιεί

διαφορετικό Layout αυτό σημαίνει ότι είναι τελείως διαφορετικό το σύνολο την δομής και όχι μόνο το εσωτερικό αποσπασμένο κομμάτι.

Στο σημείο αυτό νομίζω είναι χρήσιμο να αναδείξουμε μερικά εργαλεία σύμφωνα με τα οποία στήθηκε η διεπαφή του συστήματος.



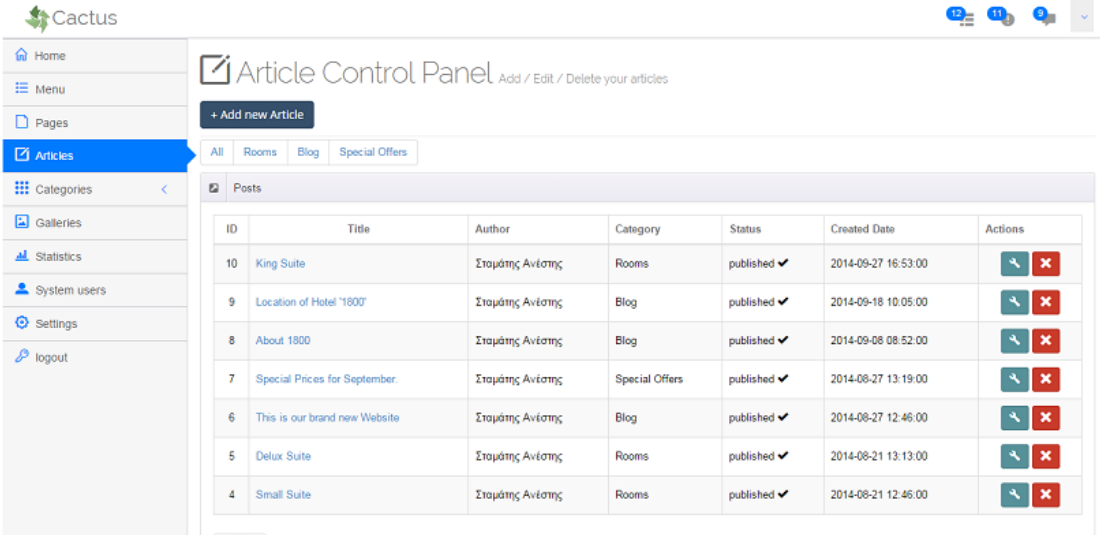
The image shows a login form for a system named 'Cactus'. At the top, there is a logo consisting of three green arrows forming a circle, followed by the word 'Cactus' in a grey sans-serif font. Below the logo, the main heading reads 'Sign in to your account' in a large, bold, grey font. Underneath this heading is a smaller line of text: 'Please enter your name and password to log in.' The form itself is a white rounded rectangle with a subtle drop shadow. It contains two input fields: the first is labeled 'Username' with a small person icon to its left, and the second is labeled 'Password' with a small lock icon to its left. To the right of the password field is a red button with the text 'Login' and a white right-pointing arrow. At the bottom of the form area, there is a small copyright notice: '2013 © Blog by Cactus.'

Εικόνα 6 Φόρμα σύνδεσης στο σύστημα.

Responsive Layout .

Responsive Layout είναι εκείνο το layout που μας επιτρέπει να προσαρμόζουμε την διεπαφή της εφαρμογής μας σε φορητές συσκευές όπως είναι τα κινητά και τα tablets. Είναι ιδιαίτερα χρήσιμο σε περιπτώσεις που ο χρήστης του συστήματος πρέπει να παρεμβαίνει ανά πάσα στιγμή και από οποιαδήποτε συσκευή στο διαχειριστικό κομμάτι του συστήματος. Είναι αλήθεια βέβαια το γεγονός ότι δεν είναι πάντα λειτουργικό και εύχρηστο κάτι τέτοιο μέσα από μια web based εφαρμογή για τον λόγο αυτό τα δύο βασικά modules συνοδεύονται και από τις αντίστοιχες android εφαρμογές τους.

Τα βασικά εργαλεία που μπορούν να συντελέσουν στην επιτυχία ενός σύγχρονου διαδικτυακού συστήματος σαφώς και είναι τελευταία έκδοση της HTML και του CSS είναι δύο πολύ δυνατά εργαλεία που μας επιτρέπουν να σχεδιάσουμε εύχρηστες και όμορφες διεπαφές για τις εφαρμογές μας. Η σχεδίαση κάθε διεπαφής δεν είναι εύκολη υπόθεση πρέπει να λαμβάνονται σοβαρά υπό όψιν σημαντικές κατευθυντήριες γραμμές όπως είναι η προσβασιμότητα από ΑμέΑ καθώς και από μη έμπυρους χρήστες.

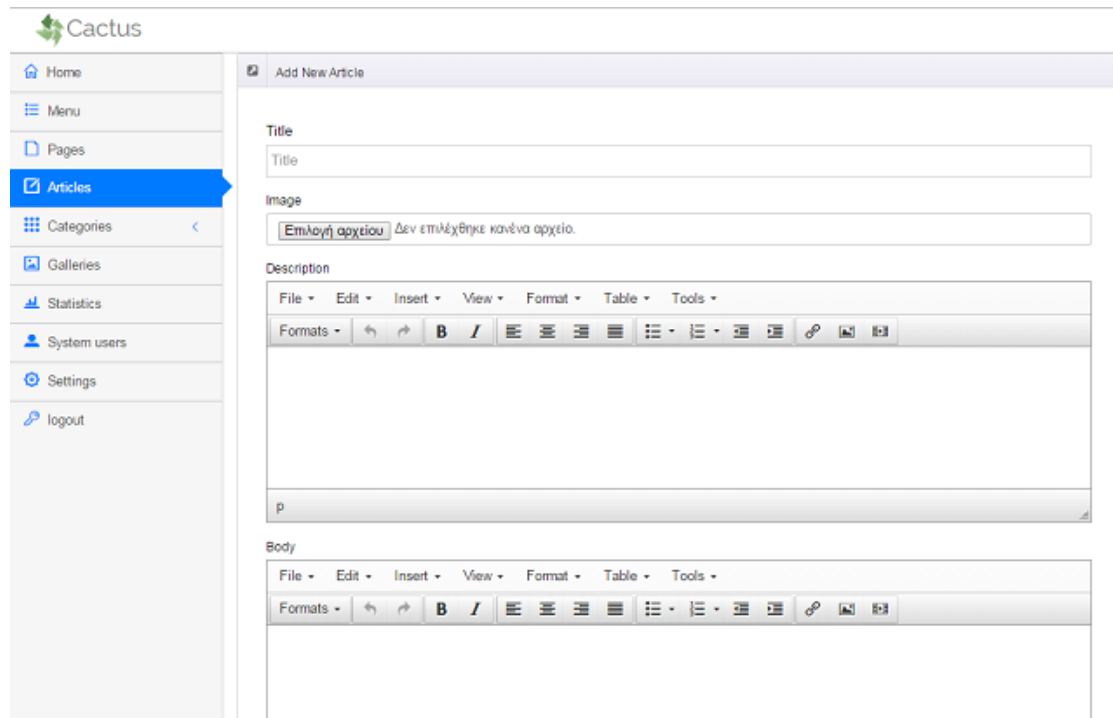


The screenshot shows the 'Article Control Panel' interface. On the left is a sidebar menu with options: Home, Menu, Pages, Articles (selected), Categories, Galleries, Statistics, System users, Settings, and logout. The main content area displays a table of articles under the 'Posts' section. The table has columns for ID, Title, Author, Category, Status, Created Date, and Actions. There are 10 rows of data, all with a 'published' status and a checkmark icon. Each row also has a magnifying glass icon and a red 'X' icon in the Actions column.

ID	Title	Author	Category	Status	Created Date	Actions
10	King Suite	Σταμάτης Ανέστης	Rooms	published ✓	2014-09-27 16:53:00	🔍 ✖
9	Location of Hotel '1800'	Σταμάτης Ανέστης	Blog	published ✓	2014-09-18 10:05:00	🔍 ✖
8	About 1800	Σταμάτης Ανέστης	Blog	published ✓	2014-09-08 08:52:00	🔍 ✖
7	Special Prices for September	Σταμάτης Ανέστης	Special Offers	published ✓	2014-08-27 13:19:00	🔍 ✖
6	This is our brand new Website	Σταμάτης Ανέστης	Blog	published ✓	2014-08-27 12:46:00	🔍 ✖
5	Delux Suite	Σταμάτης Ανέστης	Rooms	published ✓	2014-08-21 13:13:00	🔍 ✖
4	Small Suite	Σταμάτης Ανέστης	Rooms	published ✓	2014-08-21 12:46:00	🔍 ✖

Εικόνα 7 Βασική φόρμα εφαρμογής

Πτυχιακή εργασία του φοιτητή Ανέστη Σταμάτης



Εικόνα 8 Διεπαφή εισαγωγής δεδομένων

Κεφάλαιο 5. Συμπεράσματα.

Τα συμπεράσματα που προκύπτουν από την εκπόνηση της συγκεκριμένης εργασίας είναι αρκετά και σε διάφορους τομείς ας μελετήσουμε όμως τα σημαντικότερα. Είναι γεγονός ότι αν θέλουμε να δομήσουμε σωστές διαδικτυακές εφαρμογές θα πρέπει να εξοικειωθούμε με κάποιο Framework και να αποφύγουμε τον δρόμο του κώδικα από το μηδέν. Είναι γεγονός ότι συστήματα που φιλοδοξούν να παρέχουν καινοτόμες λύσεις πολλές φορές πρέπει να συνδυάσουν κομμάτια που ήδη υπάρχουν.

Τα συμπεράσματα μας όσος προς την διαχείριση και κατασκευή διαδικτυακών τόπων είναι ότι σίγουρα στην αγορά υπάρχουν πολλές εναλλακτικές λύσεις κατασκευής και ενημέρωσης μιας ιστοσελίδας είναι όμως γεγονός ότι ακόμα δεν έχει βρεθεί η χρυσή τομή που θα συνδυάζει απεριόριστη ευελιξία του συστήματος σε συνδυασμό με απλές καθημερινές εργασίες που δεν πρέπει να απαιτούν γνώσεις προγραμματισμού. Είναι επίσης ξεκάθαρο πως οι ανάγκες των ανθρώπων για την προβολή του έργου τους στο διαδίκτυο είναι τόσο ποικιλόμορφες που είναι σχεδόν αδύνατον ένα σύστημα διαχείρισης περιεχομένου να τις εκπληρώσει όλες το ίδιο καλά.

Τέλος να συμπληρώσουμε ότι η επιτυχία μια ιστοσελίδας τουλάχιστον από την σκοπιά της πληροφορικής είναι η χρήση τομή ανάμεσα στην σταθερότητα των δομών του κώδικα και στον εμπλουτισμό του με χρήσιμο κάθε φορά περιεχόμενο. Είναι επίσης στοίχημα των καιρών μας όχι απλά η διαχείριση δεδομένων αλλά η διαχείριση και αξιοποίηση πραγματικά πολύ μεγάλου όγκου διαδικτυακού περιεχομένου.

Βιβλιογραφία

1. www.wordpress.org
2. www.joomla.org
3. www.drupal.org
4. Βιβλίο «Αντικειμενοστρεφής σχεδίαση» του Αλέξανδρου Χατζηγεωργίου.
5. Βιβλίο «Τεχνολογία λογισμικού» των Γιακουμάκης και Διαμαντίδης.
6. www.cakephp.org
7. <http://book.cakephp.org/>
8. www.w3schools.com
9. <https://github.com/>
10. <http://php.net/>
11. Βιβλίο «Ανάπτυξη Web εφαρμογών με PHP και MySQL» του Welling Thomson
12. <http://dev.mysql.com/doc/refman/5.6/en/>