

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αξιοποίηση Συστημάτων Διαχείρισης Περιεχομένου για
την Ανάπτυξη Εφαρμογής Αξιολόγησης Σημείων
Ενδιαφέροντος σε μια Πόλη.

Ταραντίλης Αντώνιος

Επιβλέπων Καθηγητής: Ταμπακάς Βασίλειος
Πρόεδρος Τμήματος Μηχανικών Πληροφορικής Τ.Ε.
του ΑΤΕΙ Δυτικής Ελλάδας

Αντίρριο
2017

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή
Αντίρριο, __ / __ / ____

Επιτροπή Αξιολόγησης

Όνοματεπώνυμο

Υπογραφή

1. _____

2. _____

3. _____

Ευχαριστίες

Ευχαριστώ τους επιβλέποντες καθηγητές Ταμπακά Βασίλειο και Τζήμα Ιωάννη για την συμβολή τους στο τελικό αποτέλεσμα της πτυχιακής άσκησης και για τις άμεσες και καίριες παρεμβάσεις, όποτε αυτές χρειάστηκαν. Ευχαριστώ επιπλέον τους καθηγητές Χριστοδούλου Σωτήριο για την μοναδική διδασκαλία του στις γλώσσες διαδικτυακού προγραμματισμού, και Αλεφραγκή Παναγιώτη που μέσα από τα εργαστήρια και τις παραδόσεις του γνώρισα τον μαγικό κόσμο του προγραμματισμού.

Ευχαριστώ όλους όσους ενδιαφέρθηκαν και με στήριξαν για την ολοκλήρωση του πρακτικού μέρους και κυρίως την συμφοιτήτρια μου Στόγιου Κατερίνα για την ανιδιοτελή βοήθεια της με τα κείμενα και τις εικόνες της εφαρμογής. Τέλος επιβάλλεται να ευχαριστήσω την οικογένεια μου, που λειτουργεί τόσα χρόνια σαν χορηγός των εφαρμογών μου και όχι μόνο.

Περιεχόμενα

Πρόλογος	6
Στόχος	7
Abstract	8
Κεφάλαιο 1 - Εισαγωγή στα CMS	9
1.1 - Τι είναι τα CMS	9
1.2 - Το γραφικό περιβάλλον	10
1.3 - Γνωριμία με τα CMS	13
1.3.1 - Τα δημοφιλέστερα	13
1.3.2 - Επιλογή κατάλληλου	20
1.4 - Περιεχόμενα των CMS	24
1.5 - Τα στάδια ζωής των CMS	29
Κεφάλαιο 2 - Εγκατάσταση CMS	30
2.1 - Εγκατάσταση και παραμετροποίηση	30
2.1.1 - Εγκατάσταση σε τοπικό Server	30
2.1.2 - Εγκατάσταση σε Web Server	32
2.2 - Το caferatras.gr στην διαδικασία εγκατάστασης	34
Κεφάλαιο 3 - Ο ρόλος των templates	37
3.1 - Επιλογή template	37
3.1.1 - Αναζήτηση από υπάρχοντα	37
3.1.2 - Κατασκευή βασικού template για Joomla	38
3.1.3 - Προαιρετικές λειτουργίες	42
3.2 - Εγκατάσταση template	50
3.3 - Επεξεργασία template	51
3.3.1 - Αλλαγές στην κατηγορία Styles	51
3.3.2 - Αλλαγές στην κατηγορία Templates	52
3.3.3 - Child Themes	52
Κεφάλαιο 4 - Δημιουργία διαδικτυακής εφαρμογής.....	54
4.1 - Ο κύκλος των Extensions	54
4.1.1 - Αναζήτηση	54
4.1.2 - Εγκατάσταση και ενεργοποίηση	56
4.1.3 - Παρουσίαση των Extensions του caferatras.gr	57
4.2 - Δημιουργία σελίδων εφαρμογής	61
4.2.1 - Διαμόρφωση του αρχείου index.php	61
4.2.2 - Στάδια δημιουργίας νέας σελίδας	62
4.2.3 - Κατηγορίες σελίδων	64
4.2.4 - Δημιουργία αρχικής σελίδας	65

4.3 - Παρουσίαση του caferpatras.gr	66
4.3.1 - Μενού και σελίδες	66
4.3.2 - Εικόνες και λογότυπα	67
4.3.3 - Προχωρημένος Προγραμματισμός	70
Κεφάλαιο 5 - Προώθηση και διαχείριση διαδικτυακής εφαρμογής	81
5.1 - Μηχανισμοί προώθησης στις μηχανές αναζήτησης	81
5.1.1 - Search Engine Optimization (SEO)	81
5.1.2 - Ο ρόλος των CMS στην προώθηση	82
5.1.3 - Χειραγώγηση των μηχανών αναζήτησης	86
5.2 - Διαχείριση εφαρμογής μέσω CMS	87
5.2.1 - Συντήρηση εφαρμογής	87
5.2.2 - Παρακολούθηση εφαρμογής	90
Μελλοντικοί στόχοι	98
Επίλογος	99
Βιβλιογραφία & σύνδεσμοι	100

Εικόνες & Πίνακες

Εικόνα 1.3-1 Οι θέσεις των CMS στην αγορά στις 9/10/16	14
Εικόνα 1.3-2 Οι θέσεις των CMS στην αγορά στις 4/2/17	15
Εικόνα 1.3-3 Οι θέσεις των CMS στην Ελληνική αγορά	16
Εικόνα 2.1-1 Το Control Panel του Champp	31
Εικόνα 2.2-1 Δημοφιλή Applications στο μενού του CPanel	35
Εικόνα 2.2-2 Η ανανεωμένη εγγραφή του caferpatras.gr στην λίστα των Domain	35
Εικόνα 3.1-1 Το template στην λίστα του Template Manager	43
Εικόνα 3.1-2 Το Advanced Tab στο γραφικό περιβάλλον	48
Εικόνα 4.3-1 Αρχικό και τελικό λογότυπο της εφαρμογής	68
Εικόνα 4.3-2 Εισαγωγική εικόνα στην σελίδα Περιοχές	69
Εικόνα 4.3-3 Εικονίδια που συμβολίζουν τις παροχές	69
Εικόνα 4.3-4 Παραδείγματα λογότυπων που δημιουργήθηκαν για την εφαρμογή	69
Εικόνα 4.3-5 Ο χάρτης αρχικά είναι κρυμμένος στην σελίδα	75
Εικόνα 4.3-6 Ο χάρτης μετά την αλλαγή κατάστασης της κλάσης του	76
Εικόνα 4.3-7 Τα κουτιά σε μεγάλες οθόνες	80
Εικόνα 4.3-8 Τα κουτιά σε οθόνες tablet και κινητού	80
Πίνακας 5.2-1 Front End Groups	90
Πίνακας 5.2-2 Back End Groups	90
Εικόνα 5.2-1 Ποσοστά φορητών συσκευών που επισκέπτονται την εφαρμογή	93
Εικόνα 5.2-2 Επισκέψεις μοναδικών χρηστών στην εφαρμογή ανά ημέρα	94
Εικόνα 5.2-3 Διαγράμματα επίσκεψης χρηστών στην εφαρμογή	94
Εικόνα 5.2-4 Πίνακας προέλευσης επισκεπτών στην εφαρμογή	95
Εικόνα 5.2-5 Προτιμήσεις browser των επισκεπτών της εφαρμογής	96
Εικόνα 5.2-6 Διαγράμματα προβολής σχολίων συγκριτικά με τις επισκέψεις	97

Πρόλογος

Με την πάροδο των χρόνων και την αλματώδη εξέλιξη της, η τεχνολογία, κατάφερε να γίνει αναπόσπαστο κομμάτι της ανθρώπινης ύπαρξης. Το κατάφερε για τον απλούστατο λόγο ότι ο άνθρωπος, πάνω απ' όλα, είναι κοινωνικό ον με αστείρευτη ανάγκη για επικοινωνία και ενημέρωση. Στο παρελθόν υπήρξαν αμέτρητες ιδέες για την διάδοση λόγου μέσα σε μία κοινωνία, αλλά ο μεγαλύτερος ύμνος της ανθρώπινης εφευρετικότητας σε αυτόν τον τομέα ήρθε περίπου το 1970 και λέγεται διαδίκτυο. Όπως τα περισσότερα θαύματα της τεχνολογίας, έτσι και το διαδίκτυο ξεκίνησε ως όπλο των στρατιωτικών δυνάμεων, σταδιακά όμως απλώθηκε σε όλες τις γωνιές του πλανήτη σε τέτοιο βαθμό που πλέον η επίσκεψη στο διαδίκτυο αποτελεί μέρος της καθημερινότητας όλων μας.

Η ανάγκη του ανθρώπου για πρόσβαση στο διαδίκτυο τείνει να εξελιχθεί σε εξάρτηση, κάτι που επιβεβαιώνεται από τους αριθμούς. Από τις αρχές του 2017 καθημερινά περίπου **3,400,000,000** μοναδικές συσκευές χρησιμοποιούν το διαδίκτυο, με το ποσοστό πληθυσμού της γης που έχει πρόσβαση στο διαδίκτυο να αγγίζει το **40%**. Σε αυτό συμβάλουν οι πολλές επιπλέον λειτουργίες του διαδικτύου, πέρα της επικοινωνίας. Άλλοι το βλέπουν σαν εργαλείο, άλλοι σαν τρόπο διασκέδασης, γενικότερα οι ανάγκες και οι απαιτήσεις των χρηστών ποικίλουν, αλλά από τεχνολογική σκοπιά όλοι ζητούν το ίδιο: διαδικτυακές εφαρμογές.

Με τον όρο διαδικτυακή εφαρμογή εννοούμε το πρόγραμμα που φτιάχτηκε σε ηλεκτρονικό υπολογιστή και η πρόσβαση από τον χρήστη γίνεται μέσω διαδικτύου. Το να φτιάξει κανείς μια απλή εφαρμογή είναι αρκετά εύκολο αρκεί να έχει τις βασικές γνώσεις προγραμματισμού. Κάπως έτσι έχουμε φτάσει να υπάρχουν αυτή τη στιγμή στο διαδίκτυο περισσότερες από **1,000,000,000** εφαρμογές. Ακόμη όμως και με τόση διαθέσιμη ποικιλία, οι χρήστες δεν σταματούν να ψάχνουν συνεχώς για κάτι καινούριο.

Για να εξυπηρετείται λοιπόν, η μεγάλη ανάγκη για εφαρμογές δημιουργήθηκαν τα Συστήματα Διαχείρισης Περιεχομένου, τα οποία μας επιτρέπουν να παράγουμε λειτουργικές εφαρμογές μέσα σε λίγα λεπτά. Επιπλέον είναι τόσο απλά στην χρήση τους, που δεν χρειάζονται ιδιαίτερες γνώσεις προγραμματισμού για να τα χρησιμοποιήσεις, αρκεί μόνο υπομονή και εξάσκηση.

Στόχος

Στόχος της πτυχιακής άσκησης είναι η αναλυτική περιγραφή της δημιουργίας μιας ολοκληρωμένης και λειτουργικής διαδικτυακής εφαρμογής, αξιοποιώντας τα Συστήματα Διαχείρισης Περιεχομένου ή αλλιώς τα CMS. Για μεγαλύτερη αντίληψη και βαθύτερη κατανόηση της διαδικασίας, η άσκηση περιλαμβάνει και πρακτικό μέρος, όπου εφαρμόζονται οι οδηγίες της θεωρίας. Οι οδηγίες από την επιλογή του κατάλληλου CMS μέχρι και την συντήρηση της εφαρμογής είναι γραμμένες σε απλή γλώσσα, χωρίς πολλές άγνωστες ορολογίες, ώστε να γίνονται κατανοητές ακόμα και από αρχάριους του τομέα.

Το πρώτο κεφάλαιο της πτυχιακής άσκησης είναι απόλυτα θεωρητικό και λειτουργεί σαν εισαγωγή στο περιβάλλον των Συστημάτων. Έπειτα κάθε κεφάλαιο αποτελεί ένα δημιουργικό βήμα όπου οι θεωρητικές οδηγίες δίνονται ταυτόχρονα με τα πρακτικά παραδείγματα. Σκοπός της αναλυτικής διατύπωσης είναι ο αναγνώστης, με το πέρας ή παράλληλα με την ανάγνωση, να είναι σε θέση να κατασκευάσει την δική του διαδικτυακή εφαρμογή.

Η εφαρμογή που κατασκευάστηκε ως πρακτικό κομμάτι της πτυχιακής άσκησης υπάρχει διαθέσιμη στον σύνδεσμο www.cafepatras.gr. Η βασική ιδέα της, είναι, να παρουσιάσει τα σημεία ενδιαφέροντος, στην πόλη της Πάτρας, επιτρέποντας σε χρήστες να τα βαθμολογούν και να τα σχολιάζουν. Για την δημιουργία της επέλεξα το CMS με όνομα Joomla!, αξιοποιώντας στο μέγιστο τις δυνατότητες και τις παροχές του Συστήματος. Επιπλέον η πτυχιακή άσκηση περιέχει οδηγίες για την συντήρηση και την αντιμετώπιση προβλημάτων μίας διαδικτυακής εφαρμογής στο περιβάλλον των Συστημάτων Διαχείρισης Περιεχομένου.

Abstract

The aim of this thesis is to present a detailed description for the creation of a complete and fully functional web application, using the Content Management Systems, also known as the CMS. For better understanding and of the process, the thesis includes a practical part where we can see the theory instructions in use. The instructions, from the beginning until the end, are written in simple language, without unknown terminologies, therefore, are easily understood even by beginners in the field.

The first chapter is totally theoretical and it works as an introduction into the environment of the Content Management Systems. After that, each chapter is a creative step as the theoretical instructions and the practical examples are given simultaneously. These steps are deeply analytic so the reader can easily follow them and build his own web application.

The application we build for the practical part of this thesis is available at the link: www.cafepatras.gr. The basic idea of the application is to present the coffee shops and some interesting sights in the Greek city of Patras, allowing users to rate and comment about them. I created the application with the use of a CMS called Joomla!, making the most out of the features and benefits of the System. Moreover, this thesis provides maintenance and troubleshooting instructions for web applications, using the advantages of the Content Management Systems.

1. Εισαγωγή στα CMS

Στόχος του κεφαλαίου αυτού, είναι ο αναγνώστης να γνωρίσει τις δυνατότητες που παρέχουν τα CMS για την κατασκευή ολοκληρωμένων εφαρμογών και να είναι σε θέση να επιλέξει το κατάλληλο σύστημα που ταιριάζει στις ανάγκες του. Παρουσιάζονται τα πλεονεκτήματα και μειονεκτήματα στα πιο διαδεδομένα CMS της αγοράς και προτάσεις για αρχάριους που αποφασίζουν να ασχοληθούν με το web development, αλλά και για έμπειρους προγραμματιστές που θέλουν να επεκτείνουν τις γνώσεις τους.

1.1 Τι είναι τα CMS

Με τα αρχικά CMS (Content Management System) εννοούμε τα Συστήματα Διαχείρισης Περιεχομένου, τα οποία είναι προγράμματα που όταν τα εγκαταστήσουμε στον Server, μας επιτρέπουν να κατασκευάζουμε και να διαχειριζόμαστε ιστοσελίδες μέσω γραφικού περιβάλλοντος. Τα περισσότερα CMS έχουν γραφτεί σε PHP, είναι open source άρα μπορούμε να τα βρούμε δωρεάν. Το βασικό τους πλεονέκτημα είναι πως δεν χρειάζεται να είναι κανείς προγραμματιστής για να τα χρησιμοποιήσει, καθώς υπάρχει το γραφικό περιβάλλον και οι αλλαγές στον κώδικα είναι προαιρετικές. Αυτό όμως δεν σημαίνει ότι δεν μπορούν να χρησιμοποιηθούν από προγραμματιστές ιστοσελίδων. Αντίθετα μάλιστα, τα CMS, λύνουν τα χέρια έμπειρων προγραμματιστών λόγω του εξελιγμένου editor για τον κώδικα, των επεκτάσεων και των configurations που διαθέτουν.

Για να καταλάβουμε καλύτερα το μέγεθος και την σημασία των Συστημάτων Διαχείρισης Περιεχομένου στην παγκόσμια αγορά αρκεί να ρίξουμε μια ματιά στα τελευταία στατιστικά. Σύμφωνα με πρόσφατη έρευνα του W3Tech το **45,4%** του παγκόσμιου ιστού έχει δημιουργηθεί με χρήση κάποιου CMS. Αυτό συμβαίνει όχι μόνο γιατί κάνουν απλή την δημιουργία μιας διαδικτυακής εφαρμογής, αλλά γιατί κάνουν απλή και την διαχείριση της.

Όταν ολοκληρώσουμε τον ιστότοπο το γραφικό περιβάλλον των CMS λειτουργεί ως μέσο διαχείρισης καθώς μπορούμε εύκολα να κάνουμε αλλαγές ή να προσθέσουμε καινούριες σελίδες. Μας βοηθάει και στην βελτίωση της διαδικτυακής εφαρμογής μας, ως προς τις ανάγκες των χρηστών, αφού παρέχει πολύτιμες πληροφορίες όπως για παράδειγμα, ποια σελίδα μας είχε περισσότερους επισκέπτες, πόσες σελίδες κοιτάει ο μέσος χρήστης πριν βγει από τον ιστότοπο, ποιο το ποσοστό χρηστών επέστρεψαν στον ιστότοπο μετά την πρώτη τους επίσκεψη και άλλα πολύ χρήσιμα στοιχεία.

Ουσιαστικά ένα Σύστημα Διαχείρισης Περιεχομένου λειτουργεί σαν εργαλείο από την αρχή μέχρι το τέλος μιας διαδικτυακής εφαρμογής, μέχρι να πάψουμε δηλαδή να την υποστηρίζουμε σαν διαχειριστές. Μας βοηθάει να ανανεώνουμε διαρκώς την εφαρμογή μας και να προσθέτουμε καινούριες σελίδες, χωρίς να χρειάζεται να επεξεργαζόμαστε κώδικα απευθείας στον Server. Όμως για να καταφέρουμε να κατανοήσουμε απόλυτα την λειτουργία των CMS πρέπει να αναλύσουμε πρώτα την δομή τους.

1.2 Το γραφικό περιβάλλον

Αν και υπάρχουν διαφορές ανάμεσα στα Συστήματα Διαχείρισης Περιεχομένου (θα τις αναλύσουμε αργότερα) η λογική που κρύβεται στην δομή τους είναι η ίδια. Στην ουσία εάν κάποιος αποκτήσει εμπειρία σε ένα CMS μπορεί με λίγη εξάσκηση να τα διαχειριστεί όλα. Με αυτό το σκεπτικό θα παρουσιαστούν τα σημαντικότερα κοινά τους στοιχεία από το γραφικό περιβάλλον.

Dashboard

Μεταφράζεται ως ταμπλό και είναι ουσιαστικά η αρχική σελίδα του εκάστοτε CMS. Από αυτή την σελίδα ο διαχειριστής ελέγχει την ομαλή λειτουργία της εφαρμογής καθώς (ανάλογα με τις ρυθμίσεις που έχουν γίνει) εκεί εμφανίζονται βασικές πληροφορίες όπως:

- Πληροφορίες του Server που είναι αποθηκευμένη η σελίδα μας
- Στοιχεία και διαγράμματα της βάσης δεδομένων
- Πληροφορίες εγγεγραμμένων χρηστών
- Παρακολούθηση επισκεπτών σε πραγματικό χρόνο
- Ενημέρωση αλληλεπίδρασης χρηστών στις σελίδες
- Βελτιστοποίηση SEO (Search Engine Optimization)
- Ενημέρωση για updates των επεκτάσεων

Template Manager

Εδώ επιλέγει ο δημιουργός του ιστότοπου την εμφάνιση που θα έχουν οι σελίδες που απευθύνονται στον χρήστη. Αν θέλουμε να το πούμε πιο σωστά, εδώ γίνεται το design για το user interface. Η λογική είναι απλή, ο δημιουργός αρχικά ψάχνει στο διαδίκτυο για κάποιο template (ή αλλιώς θέμα) που να του αρέσει. Στο template περιέχεται όλη η δομή της σελίδας, από τα χρώματα μέχρι το πού θα μπαίνει το κείμενο στα άρθρα. Υπάρχουν δεκάδες δωρεάν templates για κάθε CMS αλλά τα επαγγελματικά που είναι πραγματικά αξιόπιστα είναι επί πληρωμή. Όταν βρεθεί το κατάλληλο τότε το ανεβάζουμε στο template manager και το εφαρμόζουμε στην σελίδα μας. Αν η διαδικασία γίνει σωστά τότε κάθε καινούρια σελίδα που θα δημιουργούμε θα έχει συγκεκριμένο design και δομή. Περισσότερες πληροφορίες για τα templates υπάρχουν στο κεφάλαιο 3, “Ο ρόλος των templates”.

Article Manager

Βρίσκεται συνήθως ψηλά στο μενού των CMS καθώς χρησιμοποιείται συχνότερα από κάθε άλλη επιλογή. Εδώ είναι το μέρος όπου ελέγχουμε τις σελίδες μας και δημιουργούμε νέες με το πάτημα ενός κουμπιού. Όταν κάνουμε την επιλογή για δημιουργία νέου άρθρου (add new post / new article) τότε αυτόματα παράγεται μια νέα σελίδα στον server στην οποία διατηρείται η δομή που επιλέχθηκε με το template. Όμως το μόνο που βλέπει ο δημιουργός της σελίδας είναι τον επεξεργαστή κειμένου, στον

οποίο γράφουμε και επεξεργαζόμαστε το περιεχόμενο της σελίδας. Ανάλογα το CMS εδώ μπορεί να κρύβονται και άλλες λειτουργίες όπως κατηγοριοποιήσεις των σελίδων ή επιλογές για ομαδοποιήσεις με λέξεις κλειδιά (tags). Εδώ επίσης δηλώνονται και τα metadata για κάθε σελίδα που απευθύνονται στις μηχανές αναζήτησης.

Media Manager

Εδώ βρίσκονται όλες οι εικόνες που ανεβαίνουν σε οποιαδήποτε σελίδα του ιστότοπου. Αυτό διευκολύνει τον προγραμματιστή καταρχάς γιατί μπορεί εύκολα να βρει και να ομαδοποιήσει εικόνες (π.χ. για κάποιο photo gallery) αλλά κυρίως γιατί το path για να προστεθούν εικόνες με κώδικα στις σελίδες παραμένει σταθερό. Ανάλογα με το CMS, εδώ επιτρέπεται και μερική επεξεργασία εικόνων κάτι που μπορεί να φανεί χρήσιμο για αλλαγές τις τελευταίες στιγμής όπως το να ρίξουμε την ανάλυση σε μεγάλες εικόνες που καθυστερούν την φόρτωση της σελίδας μας.

Editor

Για τους προγραμματιστές πάντα υπάρχει και ο editor όπου βρίσκονται όλα τα php και html αρχεία του template ώστε να γίνεται επεξεργασία των λεπτομερειών. Ακόμα και στα έτοιμα templates, οτιδήποτε εμφανίζεται στον χρήστη έχει δημιουργηθεί με κώδικα και έχει αποθηκευτεί σε αυτό το σημείο, άρα μπορεί και να αλλαχθεί. Επίσης εδώ μπορούν και να ανέβουν έτοιμα αρχεία με κώδικα καθώς είναι τακτική πολλών προγραμματιστών να έχουν πακέτα που τα προσθέτουν σε κάθε ιστότοπο που δημιουργούν και βοηθούν σε κάποια λειτουργία. Αυτά τα πακέτα είναι συνήθως γραμμένα σε ccs ή javascript και η λειτουργία τους μπορεί να είναι απλή, όπως για παράδειγμα το κείμενο για το copyright της σελίδας ή σύνθετη, όπως κώδικας για την προστασία της σελίδας από κακόβουλους επισκέπτες. Γενικότερα παίρνουμε μέτρα ασφαλείας όταν επεξεργαζόμαστε στον editor και είμαστε ιδιαίτερα προσεκτικοί στις αλλαγές που πραγματοποιούμε διότι είναι μόνιμες. Είναι καλή τακτική να επεξεργαζόμαστε αντίγραφα των αρχείων και να βλέπουμε τα αποτελέσματα offline ή αλλιώς όταν κάνουμε αλλαγές να έχουμε κρατήσει back-up ώστε να είμαστε ασφαλείς σε περίπτωση λάθους.

Settings

Στις ρυθμίσεις του CMS γίνεται το βασικό configuration του ιστότοπου για αυτό και εδώ χρειάζεται ιδιαίτερη προσοχή σε τυχόν αλλαγές. Ουσιαστικά τα settings είναι ο πυρήνας της εφαρμογής όπου παίρνονται οι σημαντικότερες αποφάσεις. Για παράδειγμα εδώ μπορούμε με ένα κουμπί να αποκλείσουμε την πρόσβαση στην εφαρμογή. Οι βασικότερες ρυθμίσεις που θα συναντήσει ο administrator είναι:

- Τίτλος και tagline του ιστότοπου
- URL των σελίδων
- Path για τον Server και για την βάση δεδομένων
- Metadata Settings (Κείμενα για τις μηχανές αναζήτησης)

- Γλώσσες εφαρμογής
- Online/Offline mode

Extension Manager

Τα extensions είναι κομμάτια κώδικα που τα ενεργοποιούμε στο γραφικό περιβάλλον και εκτελούν μία συγκεκριμένη και συνήθως απλή λειτουργία. Είτε μας τα παρέχει το περιβάλλον του CMS, είτε τα βρίσκουμε έτοιμα στο διαδίκτυο, για τον προγραμματιστή η χρήση τους είναι προαιρετική. Τα extensions μπορεί να έχουν διαφορετική ονομασία, ανάλογα το CMS, και κατά κύριο λόγο διαχωρίζονται σε δύο βασικές υποκατηγορίες, τα plug-ins και τα modules. Οι προ-εγκατεστημένες επεκτάσεις τις περισσότερες φορές είναι αρχικά απενεργοποιημένες, αφήνοντας την επιλογή στον διαχειριστή εάν θέλει να τις ενεργοποιήσει ή όχι. Μαζί με την ενεργοποίηση πρέπει να επιλεγεί και το σημείο που θα εμφανίζονται σε κάθε σελίδα. Γενικότερα τα extensions και οι υποκατηγορίες τους παρέχουν την λύση σε κάθε πιθανό πρόβλημα και θα είναι αδύνατον να περιγράψουμε όλη την λειτουργία τους σε μία παράγραφο. Όλο το φάσμα των δυνατοτήτων τους παρουσιάζεται αναλυτικά στο υποκεφάλαιο 4.1 “Ο κύκλος των extensions”.

Plug-ins & Modules

Η βασική διαφορά αυτών των δύο κρύβεται την πολυπλοκότητα και στην επίδραση που έχει στην εφαρμογή. Τα plug-ins είναι μικρά κομμάτια κώδικα, που κατασκευάστηκαν για να είναι απλά και να κάνουν την δουλεία τους χωρίς να ξοδεύουν περιττούς πόρους στον Server. Τα modules είναι σύνθετα, κάνουν περίπλοκες λειτουργίες, στην δομή τους συνήθως συναντάμε μεγάλα κομμάτια javascript και για να τα ελέγξουμε χρειαζόμαστε control panels, τα οποία τα βρίσκουμε στο Module Manager. Στο σύστημα βέβαια είναι απαραίτητος και ο Plug-in Manager έστω και για τις λίγες παραμετροποιήσεις που δέχονται τα plug-ins. Γενικότερα είναι η λύση για οτιδήποτε λείπει από την διαδικτυακή σου εφαρμογή και ανάλογα με το κάθε CMS, υπάρχουν εκατοντάδες από αυτά, είτε επί πληρωμή είτε όχι, στο διαδίκτυο. Δεν έχει σημασία εάν αυτό που χρειάζεσαι είναι κάτι μικρό όπως ένα φωτογραφικό άλμπουμ, είναι κάτι πιο περίπλοκο όπως ένας μηχανισμός για σχόλια χρηστών, ή ακόμα είναι κάτι εξαιρετικά σύνθετο όπως προσθήκη τρόπων πληρωμής στην εφαρμογή σου για πώληση προϊόντων, με μία αναζήτηση στο διαδίκτυο θα βρεις το plug-in ή το module που θα καλύψει την ανάγκη σου.

Components

Τα components είναι κάτι σαν μικρές εφαρμογές που τρέχουν για να παρουσιάσουν το αποτέλεσμα στα αιτήματα των χρηστών. Ουσιαστικά ενώνουν το περιβάλλον διαχείρισης όπου γράφεται η πληροφορία με το interface που επιδρά ο χρήστης που θέλει να την διαβάσει. Βρίσκονται σε κάθε σελίδα της εφαρμογής μας γιατί πέρα από την εμφάνιση της πληροφορίας, τα CMS τα χρησιμοποιούν για την λειτουργία του μενού, και γενικότερα για οποιοδήποτε αίτημα πραγματοποιείται στην εφαρμογή. Παρά την τεράστια σημασία τους δεν θα αναφερθούμε περισσότερο σε αυτά καθώς δεν υπάρχει λόγος ποτέ να επεξεργαστούμε την μέθοδο που το CMS καλεί την πληροφορία.

Users

Μια σελίδα μπορεί να την διαχειρίζονται από ένα άτομο μέχρι και μια ομάδα προγραμματιστών. Στις περισσότερες περιπτώσεις είναι απαραίτητο να μπει και ο πελάτης ως χρήστης για να προσθέτει νέα άρθρα ή σε ακόμα μεγαλύτερα project μια ομάδα αρθογράφων, ο καθένας με την δική του στήλη. Όταν δημιουργείται μια εφαρμογή σε CMS, πρώτος χρήστης με όλα τα δικαιώματα (read, write, execute) είναι ο administrator (ή super user ανάλογα το CMS), μέχρι να ορίσει αυτός άλλον, εάν το επιθυμεί. Στην κατηγορία users ο administrator μπορεί να δημιουργεί καινούριους χρήστες με συγκεκριμένα δικαιώματα για τον καθένα, ανάλογα τον ρόλο του στον ιστότοπο. Όταν για παράδειγμα στερείς το δικαίωμα execute από κάποιον χρήστη, το CMS αυτόματα όχι απλά δεν τον αφήνει να εκτελέσει εντολές αλλά του στερεί και την πρόσβαση σε συγκεκριμένα αρχεία. Με αυτή την τακτική αποφεύγονται σημαντικά λάθη που μπορούν προκύψουν από απειρία χρηστών. Αξίζει να σημειωθεί ότι η πλειοψηφία των CMS κρατάει και εμφανίζει το όνομα του user που επεξεργάστηκε τελευταίος κάποιο αρχείο καθώς και την ημερομηνία και ώρα της τελευταίας τροποποίησης.

1.3 Γνωριμία με τα CMS

Το να αποφασίσει κάποιος να εγκαταστήσει κάποιο Σύστημα Διαχείρισης Περιεχομένου είναι το πρώτο βήμα, καθώς μετά έρχεται αντιμέτωπος με μια δύσκολη απόφαση: ποιο CMS να επιλέξω; Η απάντηση δυστυχώς είναι περίπλοκη, καθώς δεν υπάρχει καλύτερο ή χειρότερο CMS, υπάρχει μόνο αυτό που ανταποκρίνεται ή δεν ανταποκρίνεται στις απαιτήσεις του προγραμματιστή. Πριν την επιλογή είναι υποχρεωτικό να συγκρίνει, να ψάξει για τα πλεονεκτήματα και μειονεκτήματα του κάθε CMS και να λάβει υπ' όψιν απόψεις άλλων προγραμματιστών. Τέλος, το σημαντικότερο είναι να πειραματιστεί και να δοκιμάσει, ώστε να δει τι του ταιριάζει και να έχει προσωπική άποψη.

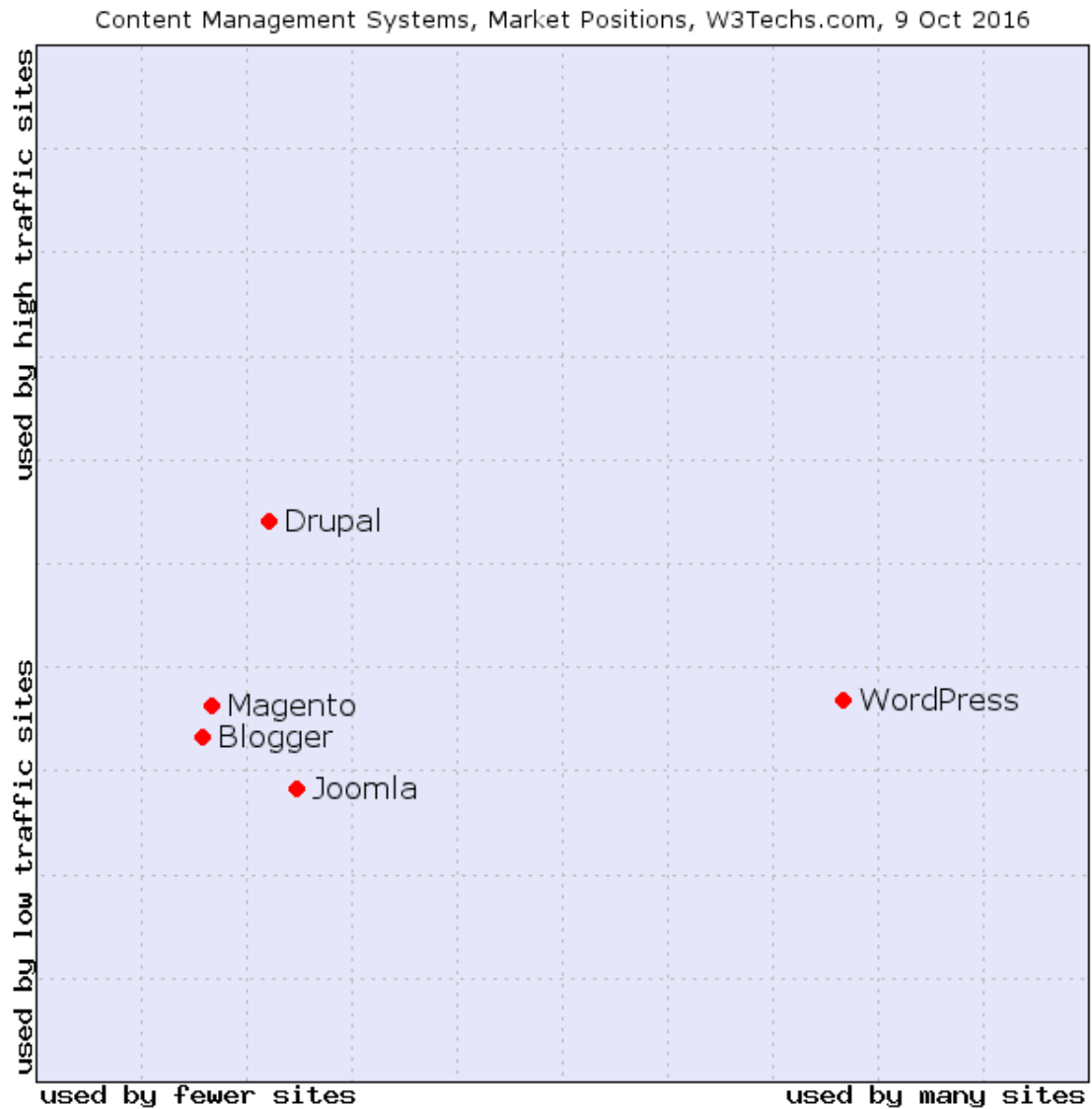
1.3.1 Τα δημοφιλέστερα

Υπάρχουν δυο κριτήρια για να αποφασίσουμε ποιο Σύστημα είναι το πιο δημοφιλές. Πρώτον, το σύστημα που είναι εγκατεστημένο στις περισσότερες ενεργές εφαρμογές και δεύτερον, το σύστημα που είναι εγκατεστημένο στις πιο δημοφιλείς εφαρμογές, αυτές δηλαδή που έχουν μεγάλη επισκεψιμότητα. Τα κριτήρια αυτά είναι εξίσου σημαντικά, καθώς το πρώτο δείχνει την προτίμηση που δείχνουν οι περισσότεροι προγραμματιστές σε κάποιο CMS, ενώ το δεύτερο υποδεικνύει πιο CMS εμπιστεύονται μεγάλες εφαρμογές με υψηλούς στόχους και απαιτήσεις.

Στους παρακάτω πίνακες μπορούμε να διακρίνουμε τα πέντε πρώτα CMS με βάση τα δύο κριτήρια που αναφέραμε. Πιο συγκεκριμένα όσο πιο δεξιά βρίσκεται κάποιο Σύστημα τόσο πιο πολλές εφαρμογές το λειτουργούν, ενώ το πόσο ψηλά βρίσκεται εξαρτάται από την συνολική επισκεψιμότητα που έχουν οι εφαρμογές.

Για να μελετήσουμε καλύτερα τις τάσεις της αγοράς ελέγχουμε τα δεδομένα και τις κινήσεις κατά την δημιουργία της πτυχιακής άσκησης. Αρχικά παρουσιάζεται ο πίνακας

με τα δεδομένα για τις 9 Οκτωβρίου 2016 (όπου ξεκίνησε η δημιουργία της πτυχιακής άσκησης):



Εικόνα 1.3-1 Οι θέσεις των CMS στην αγορά στις 9/10/16

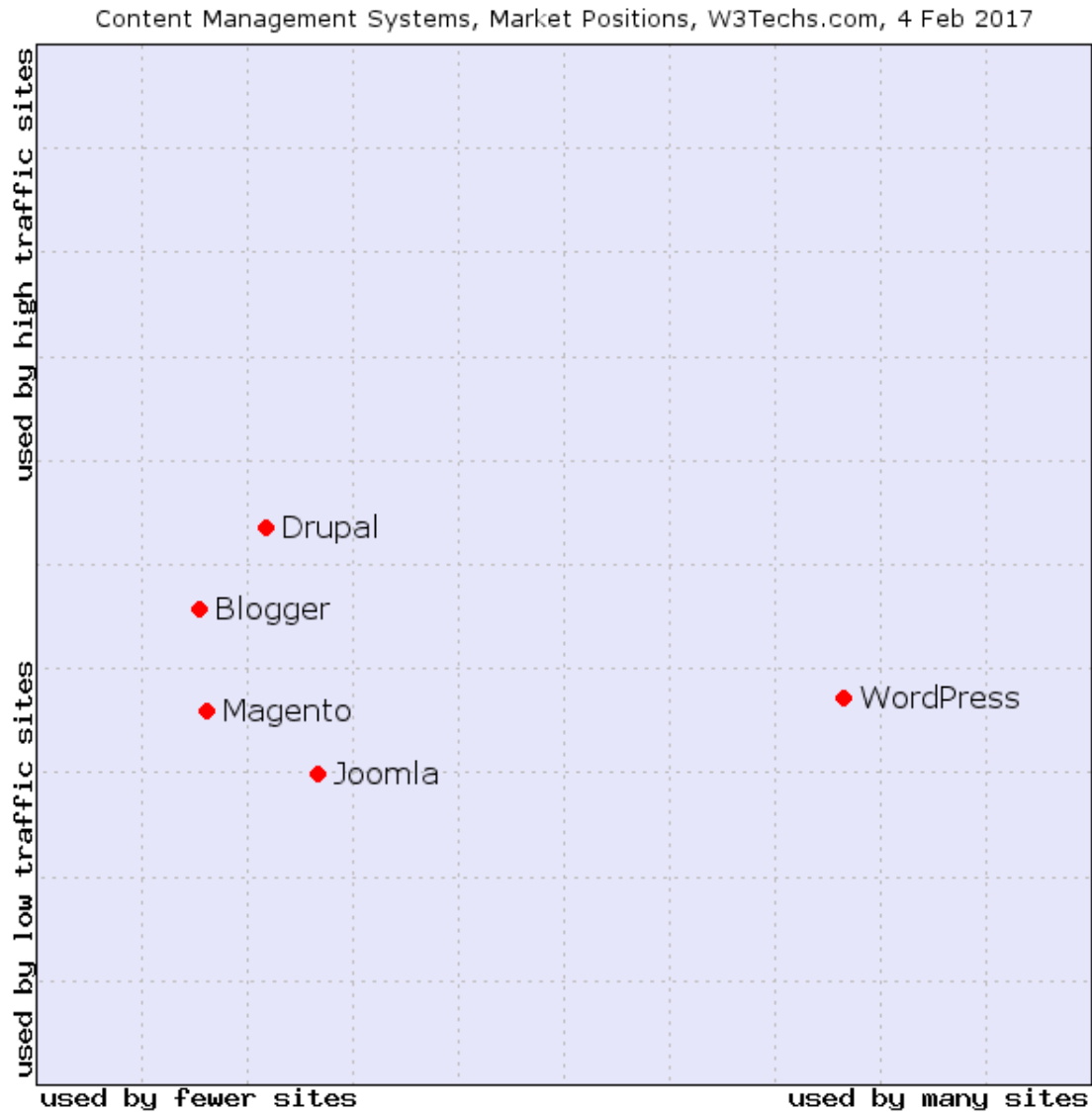
Αυτά ήταν τα πέντε πιο δημοφιλή CMS της αγοράς παγκοσμίως. Αν τα ταξινομήσουμε βάσει το ποσοστό χρήσης τους στο διαδίκτυο η σειρά είναι:

Wordpress / Joomla / Drupal / Magento / Blogger

Ενώ η σειρά με βάση τα στοιχεία επισκεψιμότητας είναι:

Drupal / Wordpress / Magento / Blogger / Joomla

Ενώ ο πίνακας για τις 4 Φεβρουαρίου 2017 (στο τέλος της πτυχιακής άσκησης):



Εικόνα 1.3-2 Οι θέσεις των CMS στην αγορά στις 4/2/17

Τα ταξινομούμε με βάση το ποσοστό χρήσης τους στο διαδίκτυο και παρατηρούμε πως δεν υπάρχει κάποια αλλαγή. Άρα το ποσοστό χρήσης δύσκολα αλλάζει:

Wordpress / Joomla / Drupal / Magento / Blogger

Αντίθετα τα στοιχεία επισκεψιμότητας έχουν τεράστια απόκλιση, καθώς το Blogger σκαρφάλωσε δύο θέσεις, ενώ και το Joomla ανέβηκε αισθητά:

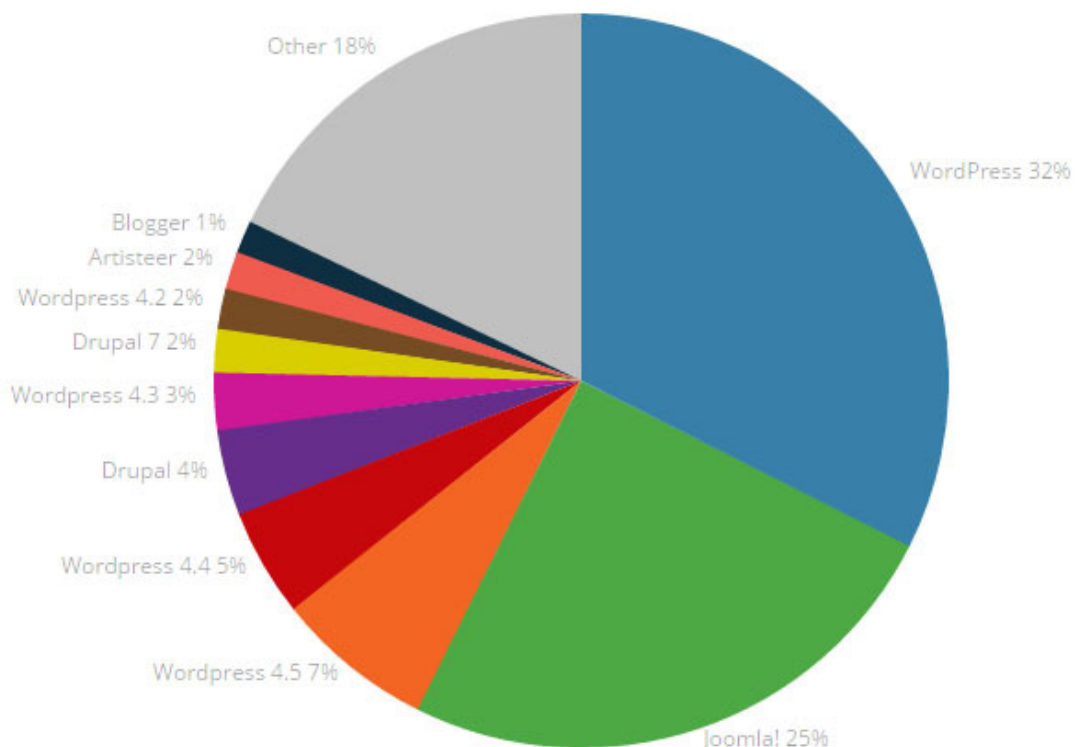
Drupal / Blogger / Wordpress / Magento / Joomla

Σημείωση:

Οι πίνακες λαμβάνουν υπ' όψιν τους πρώτους 10 εκατομμύρια ιστότοπους σύμφωνα με μετρήσεις επισκεψιμότητας της Alexa, ενώ το ποσοστό χρησιμοποίησης των CMS από ενεργές εφαρμογές είναι βασισμένο σε έρευνα της W3Techs.

Ας συγκρίνουμε τώρα τα δεδομένα της παγκόσμιας αγοράς που είδαμε με την εγχώρια κίνηση. Το παρακάτω διάγραμμα αφορά τις διαδικτυακές εφαρμογές που έχουν την κατάληξη gr στο domain name τους. Τα ποσοστά χρήσης των CMS αποκλειστικά για την χώρα μας δεν έχουν μεγάλες διαφορές με τα παγκόσμια ποσοστά, παρατηρούμε απλώς πως το magento κατέχει μερίδιο μικρότερο του 1% άρα απουσιάζει από την λίστα.

CMS usage in Greece



Εικόνα 1.3-3 Οι θέσεις των CMS στην Ελληνική αγορά

➤ Joomla!

Το Joomla! (το οποίο στο εξής για ευκολία θα το αναφέρουμε χωρίς θαυμαστικό) έχει κατοχυρώσει μια θέση στα δημοφιλέστερα CMS λόγω του support team που διαθέτει μαζί με τα αμέτρητα tutorials που λύνουν τις απορίες των προγραμματιστών. Στο γραφικό περιβάλλον διαθέτει τεράστιο μενού με ρυθμίσεις για την παραμικρή λεπτομέρεια ώστε να έχει ο administrator τον απόλυτο έλεγχο της εφαρμογής, ενώ φημίζεται για την διαχείριση των users. Παλιότερα θεωρούνταν ευάλωτο σε κακόβουλες επιθέσεις κάτι όμως που διορθώνεται στις πιο πρόσφατες εκδόσεις.



Το τεράστιο μενού όμως μπορεί να δημιουργήσει προβλήματα σε αρχάριους και να τους δυσκολέψει στον χειρισμό της εφαρμογής. Επίσης οι πολλές παράμετροι υποχρεωτικά δεσμεύουν μεγάλο χώρο στον Server με αποτέλεσμα την καθυστέρηση απόκρισης για περίπλοκες εφαρμογές.

Σημαντικότερες επιχειρήσεις και οργανισμοί που χρησιμοποιούν το Joomla σε ολόκληρη ή σε μέρος της διαδικτυακής τους εφαρμογής:

- Harvard University / gsas.harvard.edu
- United Nations Regional Information Centre / unric.org/en
- IT Wire / itwire.com
- Linux / linux.com
- Δικαιολογητικά / dikaiologitika.gr

➤ Wordpress

Είναι με διαφορά το δημοφιλέστερο στην λίστα καθώς τα ποσοστά χρήσης του, από προγραμματιστές που επιλέγουν κάποιο CMS, ξεπερνούν το 55%. Η επιτυχία του Wordpress κρύβεται στην απλότητα, καθώς είναι ταχύτατο στην εγκατάσταση, με πολύ άμεσο και κατανοητό μενού. Διαθέτει ένα τεράστιο community που συνεχώς επεκτείνει τις δυνατότητες του συστήματος και παρέχει ασταμάτητα καινούρια extensions και βελτιωμένες εκδόσεις.



Η απλότητα του συστήματος, όπως είναι λογικό βοηθάει τους αρχάριους προγραμματιστές, αλλά περιορίζει του έμπειρους που θέλουν να έχουν το απόλυτο έλεγχο της σελίδας. Γενικότερα τα εξειδικευμένα εργαλεία, που είναι δεδομένα σε άλλα

CMS, στο Wordpress απαιτούν ψάξιμο και πολλές φορές γνώσεις PHP για να προσαρμοστούν στην εφαρμογή.

Σημαντικότερες επιχειρήσεις που χρησιμοποιούν Wordpress σε ολόκληρη ή σε μέρος της διαδικτυακής τους εφαρμογής:

- Walt Disney Company / thewaltdisneycompany.com
- Microsoft News / news.microsoft.com
- Mercedes Benz / mercedes-benz.com/en
- New York Post / nypost.com
- Vogue / vogue.co.uk/

➤ Drupal

Το Drupal είναι το πιο ανεπτυγμένο CMS με αμέτρητες επιλογές ακόμα και για την πιο μικρή λεπτομέρεια. Από την στιγμή που θα ολοκληρωθεί η εγκατάσταση παρέχει τον απόλυτο έλεγχο στον administrator κάτι που είναι εξαιρετικά χρήσιμο σε επαγγελματικά projects. Ένα ακόμα μεγάλο πλεονέκτημα του είναι πως επιτρέπει την επεξεργασία των γραφικών της εφαρμογής δίνοντας την άνεση στον προγραμματιστή να πραγματοποιεί αλλαγές της τελευταίας στιγμής παρουσία του πελάτη.



Δυστυχώς οι πλούσιες παροχές του Drupal κοστίζουν σε χώρο και απλότητα. Είναι ένα βαρύ πρόγραμμα που καταλαμβάνει μεγάλο χώρο και επιβαρύνει σε μεγάλο βαθμό τον Server με αποτέλεσμα ακόμη και η εγκατάσταση του να απαιτεί ώρες και εξειδικευμένες γνώσεις κυρίως PHP.

Σημαντικότερες επιχειρήσεις και οργανισμοί που χρησιμοποιούν Drupal σε ολόκληρη ή σε μέρος της διαδικτυακής τους εφαρμογής:

- Warner Bros. Records / warnerbrosrecords.com
- Emmys / emmys.com
- French Government / gouvernement.fr
- Louvre / louvre.fr
- Tesla Motors / tesla.com

➤ Magento

Το Magento είναι ένα σχετικά καινούριο CMS καθώς έκανε την εμφάνιση του το 2008. Κυκλοφορεί σε τρεις εκδόσεις: την Enterprise, την Commerce Order Management και την Community edition, όπου από αυτές μόνο η τελευταία είναι open source και προσφέρεται δωρεάν. Είναι ένα CMS που στοχεύει αποκλειστικά στην κατασκευή και διαχείριση ηλεκτρονικών καταστημάτων. Όλες του οι ρυθμίσεις και τα extensions έχουν σκοπό να παρουσιάσουν καλύτερα το προϊόν μας και να αυξήσουν τις πωλήσεις μας. Οποιοδήποτε και αν είναι το αντικείμενο προς πώληση, σίγουρα θα βρεθεί ένα έτοιμο template που να ανταποκρίνεται στις απαιτήσεις.



Το Magento απευθύνεται σε προγραμματιστές με εμπειρία στο web development γιατί από την αρχή θα πρέπει να συνδέσουν το πρόγραμμα στην βάση δεδομένων και να ρυθμίσουν τους παραμέτρους του συστήματος στον Server. Όποια βοήθεια παρέχεται στον προγραμματιστή έρχεται με χρηματικό κόστος, μέσω του Community που είναι μόνο για εγγεγραμμένα μέλη.

Σημαντικότερα ηλεκτρονικά καταστήματα που χρησιμοποιούν magento στον ιστότοπο τους:

- Ford / accessories.ford.com
- Fox Connect / foxconnect.com
- Lenovo / outlet.lenovo.com/outlet_us/
- Olympus / getolympus.com
- Christian Louboutin / christianlouboutin.com/gr_en/

➤ Blogger

Το Blogger είναι ένα πολύ δημοφιλές CMS καθώς απευθύνεται σε απλούς χρήστες υπολογιστών χωρίς καμία γνώση προγραμματισμού. Έχει σαν στόχο να επιτρέπει στον καθένα να φτιάξει την σελίδα του στο διαδίκτυο με λίγα μόνο κουμπιά μέσα σε λεπτά. Κατασκευάστηκε από την Google άρα αυτό συνεπάγεται πως το τελικό αποτέλεσμα θα είναι απλό, mobile friendly, με καλό SEO, θα έχει καλό πλασάρισμα στις αναζητήσεις, και το κυριότερο, θα κρατάει αυτόματο backup.



Είναι λογικό πως έμπειροι προγραμματιστές δεν θα το προτιμήσουν καθώς έχει πολλούς περιορισμούς. Για παράδειγμα επιτρέπονται ελάχιστες αλλαγές στον κώδικα, δεν υπάρχει ποικιλία σε templates με αποτέλεσμα να χάνεται η μοναδικότητα, ενώ και οι σελίδες έχουν όριο ως προς το μέγεθος το 1MB.

Σημαντικά blogs που δημιουργήθηκαν με χρήση του Blogger:

- Google Blog / blog.google
- Twitter Blog / blog.twitter.com
- Post Secret / postsecret.com
- Blogger Buzz / blogger.googleblog.com
- Indexed / indexed.blogspot.gr

1.3.2 Επιλογή κατάλληλου

Κάθε εφαρμογή στο διαδίκτυο έχει ως σκοπό να παρουσιάζει κάτι καινούριο και διαφορετικό από τις προηγούμενες στον πελάτη, άρα αυτό συνεπάγεται πως η κατασκευή κάθε εφαρμογής χρειάζεται ειδική μεταχείριση. Οι στόχοι κάθε σελίδας είναι διαφορετικοί, για παράδειγμα, σελίδες που βγάζουν κέρδος από πωλήσεις αντικειμένων θέλουν σταθερή πελατεία άρα σταθερούς χρήστες, ενώ αντίθετα σελίδες που ζουν από διαφημίσεις θέλουν συνεχώς νέους χρήστες.

Καταλήγουμε, λοιπόν, στο συμπέρασμα πως η επιλογή του κατάλληλου CMS εξαρτάται από δύο παράγοντες: τις δυνατότητες του προγραμματιστή και τις απαιτήσεις της εφαρμογής. Αυτό σημαίνει πως για να προχωρήσουμε στο πρακτικό κομμάτι της εργασίας πρέπει να καταλαβαίνουμε τα γνωστικά μας όρια και να ορίσουμε τους στόχους που θα έχει η εφαρμογή μας, ώστε να ολοκληρωθεί με επιτυχία η κατασκευή. Παρακάτω αρχικά υπάρχουν τα συνήθη project που προκύπτουν σε έναν web developer μαζί με το κατάλληλο CMS για την κατασκευή τους. Έπειτα ακολουθούν οι πλατφόρμες που βασίζεται το κάθε CMS ώστε να γίνει επιλογή βάση της γλώσσας προγραμματισμού που το κατασκεύασε.

➤ Επιλογή με βάση το είδος της εφαρμογής

Υπάρχουν κάποιες κατηγορίες εφαρμογών στο διαδίκτυο οι οποίες διαφέρουν από την κλασική διαδικτυακή εφαρμογή και χρειάζονται ειδική μεταχείριση. Οι τέσσερις πιο δημοφιλείς κατηγορίες με γνώμονα την ζήτηση της αγοράς είναι τα blogs, τα portfolio sites, τα forums και τα e-shop. Παρακάτω αρχικά εξηγούμε τις διαφορές τους σε σχέση με τις υπόλοιπες διαδικτυακές εφαρμογές και έπειτα προτείνουμε το κατάλληλο CMS για κάθε περίπτωση.

Blog

Το blog είναι μία εφαρμογή, όπου στην αρχική της σελίδα παρουσιάζονται σε λίστα οι τίτλοι από προσωπικά κείμενα και απόψεις, σαν κάποιος να έγραφε στο ημερολόγιό του. Από εκεί προκύπτει και η λέξη blog, καθώς είναι συντομογραφία του web log το οποίο σημαίνει καταγραφή στο διαδίκτυο. Ξεκίνησε με την λογική, ο administrator να αποτυπώνει στιγμιότυπα της καθημερινότητας του, με τον καιρό όμως εξελίχθηκε, και πλέον υπάρχουν blogs που αφορούν κριτικές για ταινίες, αθλητικά, μουσική και ότι άλλο μπορεί να φανταστεί κανείς. Παραδοσιακά δεν έχουν μενού, είναι απλά η αρχική σελίδα και η σελίδα του κάθε κειμένου, δεν υπάρχουν ιδιαίτερες ρυθμίσεις και templates για να μοιάζει μοναδικό και η αναζήτηση κειμένων γίνεται είτε με τα tags είτε με την ημερομηνία δημοσίευσης.

Για αρχάριους που ενδιαφέρονται να κατασκευάσουν το δικό τους blog υπάρχει η προφανής λύση του **Blogger**, αλλά ένα ακόμη πολύ ενδιαφέρον και συνεχώς αναπτυσσόμενο CMS είναι το **Wix** το οποίο επιτρέπει γραφικά την δημιουργία περισσότερων components στην σελίδα και την κάνει να μοιάζει πιο επαγγελματική. Το μόνο που χρειάζεται να κάνει ο administrator είναι διπλό κλικ στο σημείο της σελίδας που θέλει να επεξεργαστεί κάτι, ενώ για να προσθέσει κάποιο component υπάρχει η επιλογή add στα αριστερά της οθόνης. Για τους προχωρημένους, το **Wordpress** είναι η σίγουρη λύση για δημιουργία blog καθώς δημιουργήθηκε με σκοπό να αλλάξει τα δεδομένα του blogging, άσχετα που με τα χρόνια εξελίχθηκε σε πρόταση για ολοκληρωμένες εφαρμογές. Λόγω της τεράστιας ποικιλίας σε extensions ο administrator έχει την δυνατότητα να δημιουργήσει κάτι ξεχωριστό.

Προβολή επιχείρησης / Portfolio

Η προβολή επιχείρησης στο διαδίκτυο είναι από τα πιο συνηθισμένα αιτήματα για προγραμματιστές στο αρχικό στάδιο του web development καθώς δεν νοείται επιχείρηση χωρίς ιστότοπο. Σε αυτήν την κατηγορία ανήκουν και τα portfolio sites, τα οποία είναι ιστοσελίδες όπου προβάλλονται τα προσόντα και παρουσιάζονται οι εργασίες ενός ατόμου ή μιας ομάδας. Αυτοί οι ιστότοποι δεν έχουν καμία σχέση με τα blogs καθώς εδώ δεν υπάρχουν συνεχώς νέα κείμενα, όσον αφορά την αρχική σελίδα τουλάχιστον, αλλά μια σταθερή πληροφορία. Εδώ μεγάλο ρόλο παίζει η αμεσότητα της πληροφορίας και η αισθητική της σελίδας, ενώ επίσης απαραίτητη είναι η δημιουργία μενού που θα καλύπτει όλες τις πτυχές της επιχείρησης. Για το τέλος αφήνουμε τα extensions των CMS τα οποία θα ολοκληρώσουν την εμπειρία του χρήστη με λειτουργίες όπως: φόρμα επικοινωνίας, φωτογραφικά άλμπουμ, υλικό από μέσα κοινωνικής δικτύωσης και πολλά άλλα.

Το κατάλληλο CMS για αρχάριο σε αυτή τη περίπτωση είναι σίγουρα το **Weebly**. Είναι σύστημα που εφαρμόζει την λογική του drag and drop στην σελίδα, δίνοντας στον χρήστη την δυνατότητα να πιάνει με το ποντίκι τα components που θέλει να προσθέσει, από ένα πίνακα στα αριστερά της οθόνης και να τα τοποθετεί εύκολα σε όποιο σημείο της σελίδας του επιθυμεί. Διαθέτει αρκετές επιλογές για animation των στοιχείων, με χρήση CSS3 και Javascript, τα οποία μπορούν να κάνουν ξεχωριστή μια σελίδα, ενώ επίσης στο κανάλι του Weebly στο YouTube, υπάρχουν αμέτρητα tutorials. Για χρήστες

όμως που ζητούν μεγαλύτερη ελευθερία σχεδίασης και επαγγελματικό αποτέλεσμα η επιλογή είναι ανάμεσα σε **Joomla** και **Drupal**, που αναλύσαμε προηγουμένως.

Forums

Το forum είναι πιο περίπλοκο από μια απλή σελίδα στο διαδίκτυο καθώς είναι απαραίτητο να έχει εγγραφή χρηστών και απόκρυψη ορισμένων σελίδων σε επισκέπτες. Η λογική των forum είναι η δημιουργία μιας κλειστής κοινωνίας, η οποία συνήθως ασχολείται με κάτι συγκεκριμένο (π.χ. τεχνολογία), και που οι χρήστες ξεκινούν συζητήσεις είτε για κάποιο πρόβλημα τους είτε για να εκφράσουν την άποψη τους. Οι υπόλοιποι χρήστες αλληλεπιδρούν κάτω από το συγκεκριμένο θέμα σχολιάζοντας δημιουργώντας έτσι κάτι σαν νήμα. Τους κανόνες στα forums, όπως θέματα για συζήτηση και πολιτική για τον τρόπο σχολιασμού, τους θέτει ο administrator και πρέπει να είναι σε θέση μέσω του συστήματος να επιβάλει κυρώσεις σε χρήστες που τους παραβαίνουν. Είναι κατανοητό πως εδώ χρειάζεται ένα CMS κατασκευασμένο για να δίνει στον administrator τον απόλυτο έλεγχο.

Τα forums δεν είναι εύκολος τομέας για να ξεκινήσει κάποιος το web development, αλλά αν είναι αποφασισμένος το πιο απλό σύστημα είναι το **Vanilla**. Στην δωρεάν έκδοση στερεί από τον χρήστη αρκετές δυνατότητες με αποτέλεσμα η τελική σελίδα να μοιάζει ξεπερασμένη, αλλά αν εγκατασταθεί σωστά ακόμα και με τις προεπιλεγμένες ρυθμίσεις θα είναι λειτουργικό. Για χρήστες που είναι εξοικειωμένοι με το Wordpress, το **bbPress** είναι η σίγουρη επιλογή, καθώς είναι δημιουργία της ίδιας προγραμματιστικής ομάδας και προσφέρουν ένα γνώριμο περιβάλλον μαζί με τα πολλά διαθέσιμα extensions του συστήματος. Μια ακόμα σταθερή πλατφόρμα είναι το **myBB**, το οποίο δημιουργήθηκε το 2002 και διαρκώς εξελίσσεται με δυνατό του σημείο την αναλυτική στατιστική απεικόνιση των χρηστών στις σελίδες. Παρουσιάζει γραφήματα που βοηθούν τον administrator να ελέγχει εύκολα συζητήσεις με χιλιάδες σχόλια και νήματα που τρέχουν.

E-shop

Σε αυτήν την κατηγορία οι απαιτήσεις αυξάνονται, διότι πέρα από εγγραφές χρηστών και διαρκή ενημέρωση των σελίδων, έχουμε να κάνουμε και με πληρωμές. Το σύστημα που θα επιλέξουμε πρέπει να είναι αξιόπιστο, ασφαλές και με μεγάλες δυνατότητες παραμετροποίησης. Οι χρήστες πρέπει να νιώθουν σιγουριά και άνεση για να εμπιστευτούν τα χρήματά τους σε έναν ιστότοπο, οπότε η εφαρμογή δεν πρέπει μόνο να είναι προστατευμένη, αλλά και να φαίνεται, κυρίως μέσα από τις λεπτομέρειες. Χρειάζονται αναλυτικά μενού για πρόσβαση στα προϊόντα, σελίδες με επεξήγηση όρων και προϋποθέσεων καθώς και διάφοροι τρόποι πληρωμής. Ο χρήστης πρέπει να απολαμβάνει την περιήγηση του στην σελίδα, να έχει άμεση πρόσβαση σε αυτό που ψάχνει και να του εμφανίζονται προτάσεις ανάλογα με το γούστο του.

Είναι πρακτικά αδύνατον ένας αρχάριος να κατασκευάσει ένα πετυχημένο e-shop με την χρήση CMS. Το πιο απλό σενάριο είναι η εγκατάσταση του **Wordpress** μαζί με το extension **WooCommerce** το οποίο είναι αξιόπιστο αλλά με ελάχιστες δυνατότητες παραμετροποίησης. Όπως αναφέραμε και νωρίτερα το κατάλληλο CMS που δημιουργήθηκε για τις απαιτήσεις τις διαδικτυακής πώλησης είναι το **Magento** με αμέτρητα templates που καλύπτουν σχεδόν ολόκληρο το εύρος των επιχειρήσεων. Για

παράδειγμα εάν το e-shop που ενδιαφερόμαστε να κατασκευάσουμε πουλάει ρούχα ή βιβλία, στην λίστα με τα διαθέσιμα templates σίγουρα θα βρούμε κάποια έτοιμα που απευθύνονται ακριβώς στις απαιτήσεις μας. Στο γραφικό περιβάλλον του Magento υπάρχει και ένα ολοκληρωμένο σύστημα υποστήριξης πελατών όπου ο επισκέπτης της σελίδας μπορεί να μιλάει προσωπικά με τον administrator μέσω online φόρμας. Καταλήγουμε πως για να ρυθμιστεί σωστά και να είναι λειτουργικό ένα e-shop χρειάζεται ομάδα προγραμματιστών ή έναν ικανότατο προγραμματιστή.

➤ Επιλογή με βάση την πλατφόρμα του CMS

Παραπάνω αναλύσαμε πιθανά project για εφαρμογές που προκύπτουν συχνά και προτείνουμε κάποια CMS ως λύση. Όλα τα CMS που προτείνουμε μέχρι στιγμής έχουν δημιουργηθεί σε PHP πλατφόρμα, κάτι που είναι λογικό διότι η λογική της PHP εξυπηρετεί απόλυτα τον τομέα του web development. Ο κάθε προγραμματιστής όμως έχει τον δικό του τομέα ειδίκευσης, πράγμα που σημαίνει πως είναι πιθανό να έχει μεγαλύτερη άνεση σε CMS άλλης πλατφόρμας. Για αυτό το ενδεχόμενο θα γνωρίσουμε τα πιο δημοφιλή και αξιόπιστα CMS της αγοράς, χωρισμένα με βάση την τεχνολογία που αναπτύχθηκαν.

Java

Η αξιοπιστία και η ασφάλεια της Java πλατφόρμας με τις αμέτρητες βιβλιοθήκες και το τεράστιο support community είναι ιδανική βάση για την κατασκευή διαδικτυακών εφαρμογών. Τα δυνατά τους σημεία είναι, πρώτον, η υποστήριξη από σχεδόν όλες τις βάσεις δεδομένων, όπως MySQL, Oracle, HSQL, SQL Server, Ingres, DB2 και Apache Derby, και δεύτερον, η σύνδεσή τους με την γλώσσα στην οποία βασίζονται οι περισσότερες εφαρμογές. Το δημοφιλέστερο CMS της πλατφόρμας είναι σταθερά το **Alfresco**, το οποίο εκμεταλλεύεται πλήρως Java τεχνολογίες όπως τα JavaServer Faces και το Hibernate και εγγυάται στον τελικό χρήστη μια συμπαγή και ασφαλή εφαρμογή. Για κάποιον που δεν είναι εξοικειωμένος με την java πλατφόρμα η καλύτερη επιλογή είναι το CMS **Mangolia** οποίο περιορίζει τις δυνατότητες του χρήστη, χωρίς όμως να υστερεί σε αξιοπιστία. Σύμφωνα με τους κατασκευαστές το Magnolia χρησιμοποιείται από επιχειρήσεις σε πάνω από 100 χώρες στον κόσμο.

Perl

Είναι όπως και η Java, μια αντικειμενοστραφής γλώσσα, με την διαφορά ότι η Perl θεωρείται οικογένεια δυναμικών τεχνολογιών καθώς δανείζεται στοιχεία από πολλές άλλες γλώσσες. Το δημοφιλέστερο CMS με βάση την Perl πλατφόρμα είναι το **Twiki**, το οποίο πέρα από το Σύστημα για την κατασκευή εφαρμογής προσφέρει επίσης υπηρεσίες hosting. Επιτυχία γνωρίζει και το **Movable Type** το οποίο ξεκίνησε ως λύση αποκλειστικά για την δημιουργία blog αλλά εξελίχτηκε και πλέον υποστηρίζει περισσότερα είδη εφαρμογών. Δυστυχώς οι δημιουργοί του ανακοίνωσαν πως με την καινούρια έκδοση του Movable Type σταματούν αυτόματα την υποστήριξη της δωρεάν έκδοσης ανοιχτού κώδικα.

Python

Η Python είναι μια γλώσσα που έχει σαν λογική να παράγει το τελικό αποτέλεσμα με όσο το δυνατόν λιγότερες γραμμές κώδικα ώστε να διευκολύνεται ο προγραμματιστής. Λογική που προφανώς θυμίζει αυτή των CMS με αποτέλεσμα να έχουν επιχειρήσει ομάδες προγραμματιστών να αξιοποιήσουν τις δυνατότητες της γλώσσας στον τομέα του web development. Για τον λόγο αυτόν δημιουργήθηκε το **Django framework** το οποίο δεν είναι CMS αλλά η εγκατάσταση του στον Server βοηθάει τον administrator να διαχειριστεί την σελίδα του, αρκεί να γνωρίζει τις βασικές αρχές της Python. Με βάση αυτό το framework δημιουργήθηκε, μόλις μέσα στο 2014, το **Wagtail** το οποίο εκμεταλλεύεται πλήρως την απλότητα και το μικρό μέγεθος του Django και παράγει για τον administrator ένα μοντέρνο και απλό στην χρήση του γραφικό περιβάλλον. Σε διαδικασία παραγωγής βρίσκεται το **Django CMS** το οποίο λογικά θα εξερευνεί βαθύτερα τις δυνατότητες της πλατφόρμας Python.

Ruby

Η Ruby είναι μια δυναμική, αντικειμενοστραφής γλώσσα προγραμματισμού που η συγγραφή της επηρεάστηκε από την Python. Όπως και προηγουμένως με το Django έτσι και εδώ υπάρχει το **Ruby on Rails** το οποίο είναι framework που τρέχει στον server και μας βοηθάει στην κατασκευή ιστοσελίδων. Με την χρήση του Ruby on Rails κατασκευάστηκαν αρκετά CMS, με πιο γνωστό το **Radiant** το οποίο μαζί με τις πλούσιες παροχές του, προσφέρει επίσης μεγάλη ποικιλία από templates. Το λιγότερο σύνθετο CMS που μπορεί να βρει κάποιος σε Ruby είναι το **Browser CMS**, που κατασκευάστηκε με την λογική να το χρησιμοποιεί ο οποιοσδήποτε και όχι μόνο οι γνώστες της γλώσσας.

Άλλες Πλατφόρμες

Οι υπόλοιπες πλατφόρμες υστερούν στον τομέα της δημιουργίας web applications. Για παράδειγμα έχουν κατασκευαστεί CMS σε Javascript πλατφόρμα όπως το **Ghost** χωρίς να καταφέρουν να καθιερωθούν, κυρίως λόγω της πολυπλοκότητας της γλώσσας. Άξια αναφοράς είναι τα CMS που γράφονται με την χρήση του ASP.NET fragment της Microsoft όπως το **Umbraco** και το **DotNetNuke**, τα οποία αυξάνουν συνεχώς το community τους.

1.4 Περιεχόμενα των CMS

Για να ολοκληρώσουμε την γνωριμία μας με τα CMS, πρέπει να κατανοήσουμε τους φακέλους και τα αρχεία που το απαρτίζουν. Τα περιεχόμενα για κάθε CMS διαφέρουν σε κάποιο βαθμό, οπότε θα επιλέξουμε το πιο γενικό παράδειγμα του Joomla. Επιλέγουμε το συγκεκριμένο CMS όχι μόνο γιατί σε αυτό βασίστηκε η κατασκευή του πρακτικού μέρους, αλλά επειδή είναι μεσαίας πολυπλοκότητας και ενώ είναι γραμμένο στην πιο διαδεδομένη πλατφόρμα για Web Applications, την PHP, περιέχει και Java κώδικα. Παρακάτω αναλύουμε την λειτουργία των φακέλων που υπάρχουν μέσα στο zip αρχείο και αποτελούν το CMS Joomla.

Φάκελος administrator

Είναι ο μεγαλύτερος φάκελος του συστήματος στο οποίο περιέχονται όλα τα αρχεία που διαμορφώνουν τη δομή του προγράμματος και το γραφικό του περιβάλλον. Όλες οι επιλογές του μενού, όλες οι δυνατότητες που παρουσιάζονται στον administrator και του κάνουν την ζωή εύκολη, υπάρχουν γιατί περιέχονται σε αυτόν τον φάκελο. Στον συγκεκριμένο φάκελο επεμβαίνουμε μόνο όταν είμαστε απολύτως σίγουροι για τις δυνατότητες μας και κυρίως για αλλαγές στην εμφάνιση του γραφικού περιβάλλοντος ή για να προσθέσουμε δυνατότητες στο σύστημα. Για παράδειγμα εδώ μπορούμε να προσθέσουμε χειροκίνητα μια νέα γλώσσα ή για να αλλάξουμε χρώμα στο θέμα του συστήματος. Ο φάκελος όσο πραγματοποιούμε δραστηριότητες συνεχώς επεκτείνεται, δημιουργώντας αυτόματα ακόμα και υποφακέλους εάν χρειαστεί. Ένας τρόπος για να κατανοήσουμε την λειτουργία αυτού του φακέλου είναι να τον σκεφτούμε ως τόπος αποθήκευσης των ενεργειών του administrator, οι οποίες δεν αφορούν τον κώδικα της εφαρμογής που δημιουργεί.

Φάκελος bin

Το μοναδικό αρχείο που περιέχεται στον φάκελο bin είναι το keychain.php, το οποίο περιέχει και συναρτήσεις γραμμένες σε Java. Ουσιαστικά η λειτουργία του είναι να ελέγχει και να προστατεύει τα στοιχεία του χρήστη κατά την διάρκεια της εγγραφής του και της εισόδου του στο σύστημα. Αυτός ο φάκελος δεν επεκτείνεται ποτέ αυτόματα και επιδέχεται αλλαγές μόνο από τον administrator όταν αυτός θέλει να επεξεργαστεί τις εγγραφές στο keychain.

Φάκελος cache

Ο cache φάκελος είναι όπως σε κάθε πρόγραμμα η προσωρινή μνήμη του συστήματος, απαραίτητος για την άμεση απόκριση της εφαρμογής και την ομαλή λειτουργία των extensions. Αρχικά ο φάκελος είναι άδειος μέχρι και να εκθέσουμε την εφαρμογή στο διαδίκτυο. Συνήθη τακτική για ελαχιστοποίηση παρενεργειών είναι πριν κάνουμε οποιαδήποτε αναβάθμιση στο σύστημα ή σε κάποιο ενεργό extension να αδειάζουμε εκ νέου τον cache φάκελο, κάτι που όμως δεν είναι και υποχρεωτικό.

Φάκελος cli

Περιέχει PHP αρχεία τα οποία είναι υπεύθυνα να παρουσιάζουν στον administrator σε μορφή ειδοποιήσεων όλες τις διαθέσιμες αναβαθμίσεις για το σύστημα και τα extensions. Παράλληλα περιέχει το αρχείο deletefiles.php στο οποίο βρίσκεται η απαραίτητη μέθοδος delete-Unexisting-Files που είναι υπεύθυνη για την διαγραφή περιττών αρχείων μετά την αναβάθμιση τους.

Φάκελος components

Αναφέραμε και νωρίτερα την σημασία των components για την λειτουργία του συστήματος. Στον φάκελο components βρίσκονται τα αρχεία που περιέχουν τον κώδικα

αυτών των μικρών εφαρμογών που καλεί η εφαρμογή μας. Εδώ μπορεί ο administrator να τα επεξεργαστεί τις εντολές τους κάτι που συστήνεται μόνο σε προχωρημένους προγραμματιστές.

Φάκελος images

Περιέχονται χωρισμένες σε φακέλους οι εικόνες που εμφανίζονται στην τελική εφαρμογή και όχι οι εικόνες που υπάρχουν στο γραφικό περιβάλλον. Από την στιγμή που το γραφικό περιβάλλον παρέχει τον media manager ο μόνος λόγος να επέμβει κανείς σε αυτόν τον φάκελο είναι για να βρει το ολοκληρωμένο μονοπάτι στον Server μιας εικόνας.

Φάκελος includes

Εδώ συναντούμε το αρχείο defines.php όπου ορίζονται τα global definitions, δηλαδή μεταβλητές οι οποίες θα είναι ορατές σε όλο το πρόγραμμα με το όνομα που τις δηλώσαμε εδώ. Σπανίως πραγματοποιείται αλλαγή εδώ καθώς είναι δύσκολο να ανακαλύψουμε ποια αρχεία θα επηρεαστούν από τις μεταβολές.

Φάκελος installation

Όπως είναι προφανές από το όνομα του, αυτός ο φάκελος περιέχει τα αρχεία για την εγκατάσταση του συστήματος στον Server. Όταν ολοκληρωθεί η διαδικασία είναι απαραίτητο να διαγραφεί αυτός ο φάκελος καθώς σε αντίθετη περίπτωση εγκυμονούν κίνδυνοι για την διαγραφή των αποθηκευμένων αρχείων μας. Το Joomla ανήκει στα CMS που δεν επιτρέπουν εκκίνηση του περιβάλλοντος όσο υπάρχει ο συγκεκριμένος φάκελος.

Φάκελος language

Σημαντικός φάκελος διότι δεν περιέχει μόνο την γλώσσα του συστήματος αλλά και τις γλώσσες των components και των extensions. Αυτό σημαίνει ότι εδώ μπορούμε χειροκίνητα να αλλάξουμε γλώσσα μιας εφαρμογής η οποία για παράδειγμα επιστρέφει εκφράσεις στα Αγγλικά. Μπορούμε ακόμα να φτιάξουμε αρχείο με τις εκφράσεις που επιθυμούμε εμείς να επιστρέφονται, αρκεί να αλλάξουμε την διεύθυνση του αρχείου γλώσσας που έχει το extension στον κώδικα του.

Φάκελος layouts

Περιέχονται οι ρυθμίσεις για τα layouts του συστήματος και των εφαρμογών. Η διαφορά στα layouts με τον φάκελο του administrator είναι ότι εδώ περιέχονται οι αλλαγές οι οποίες είναι διαθέσιμες και από το γραφικό περιβάλλον. Επομένως δεν έχει νόημα να επεξεργαζόμαστε τα συγκεκριμένα αρχεία ούτε να πραγματοποιούμε αλλαγές σε κώδικα και αρχεία, ενώ μπορούμε γραφικά από το μενού.

Φάκελος libraries

Εδώ αποθηκεύονται οι πραγματικά αμέτρητες βιβλιοθήκες για την αποτελεσματική και ομαλή λειτουργία του συστήματος. Περιέχονται βιβλιοθήκες και κανόνες που αφορούν σημαντικά αρχεία όπως τα δικαιώματα του administrator ως και λεπτομέρειες όπως το extension για τις παραλλαγές του μενού που παράγεται στην εφαρμογή. Κάθε αλλαγή σε αυτά τα αρχεία μπορεί να δημιουργήσει πολλαπλές επιπλοκές στο σύστημα καθώς οι βιβλιοθήκες ορίζουν global μεταβλητές.

Φάκελος media

Αναφέραμε νωρίτερα πως στον φάκελο images δεν περιέχονται οι εικόνες και τα εικονίδια του συστήματος καθώς οτιδήποτε είναι αρχείο εικόνας και αφορά το σύστημα βρίσκεται στον φάκελο media. Μπορεί να είναι ένα μικροσκοπικό βελάκι, μπορεί να είναι μια παύλα, ή ακόμα και το σήμα του Joomla, όλα τα αρχεία εικόνας οποιουδήποτε τύπου, αποθηκεύονται εδώ.

Φάκελος modules & Φάκελος plug-ins

Αναφέραμε και νωρίτερα πως τα modules και τα plug-ins είναι τύποι extensions και πως αποτελούν τεράστιας σημασίας εργαλεία που μας παρέχουν τα CMS για την κατασκευή εφαρμογών. Διαθέτουν ξεχωριστό φάκελο γιατί έχουν ουσιαστικές διαφορές μεταξύ τους, τις οποίες όμως θα τις δούμε στην πράξη στο τέταρτο κεφάλαιο, κατά την διαδικασία της εγκατάστασης και επεξεργασίας τους. Όταν εγκαθιστούμε κάποιο extension στο σύστημα, τα αρχεία του αποθηκεύονται σε αυτούς τους φακέλους, ανάλογα τον τύπο τους. Αυτούς τους φακέλους τους επισκέπτεται συχνά ένας web developer καθώς πολλές φορές η επεξεργασία τους είναι απαραίτητη για το τελικό αποτέλεσμα. Αξίζει να αναφέρουμε πως αρχικά οι φάκελοι δεν είναι κενοί, καθώς κάποια extensions περιέχονται από προεπιλογή και είναι ήδη εγκατεστημένα στο σύστημα. Μερικά παραδείγματα είναι τα image modules, η φόρμα για το login και extensions που αφορούν το μενού της εφαρμογής που παράγεται αυτόματα.

Φάκελος templates

Στον φάκελο templates αποθηκεύονται τα θέματα που έχουμε αποθηκεύσει στο CMS για την συγκεκριμένη εφαρμογή. Έχουμε την δυνατότητα να δοκιμάζουμε πολλά θέματα χωρίς να χρειάζεται να διαγράψουμε το παλιό. Τα κρατάμε απλώς όλα αποθηκευμένα σε αυτόν το φάκελο και μέσω του template manager επιλέγουμε πιο θα είναι σε κατάσταση activate. Για προσθήκη ή διαγραφή κάποιου θέματος πηγαίνουμε στο template manager, ενώ για επεξεργασία του κώδικα των θεμάτων προτιμούμε τον editor του μενού, με αποτέλεσμα να επισκεπτόμαστε αυτόν τον φάκελο μόνο σε δύο περιπτώσεις. Πρώτον, όταν υπάρχει κάποιο κόλλημα στο σύστημα και δεύτερον όταν υποψιαζόμαστε πως ένα έτοιμο template μας αποκρύπτει αρχεία στο γραφικό περιβάλλον, οπότε ερχόμαστε εδώ που είναι όλα φανερά.

Φάκελος tmp

Εδώ αποθηκεύονται αυτόματα καταστάσεις του template, ώστε σε περίπτωση προβλήματος του συστήματος ή προγραμματιστικού λάθους να υπάρχει μια δυνατότητα επαναφοράς προηγούμενης κατάστασης και αναίρεσης. Σε καμία περίπτωση αυτό δεν θεωρείται αξιόπιστη λύση back up.

htaccess.txt

Είναι αρχείο που τρέχει σε αποκλειστικά Apache Server και περιέχει εντολές που είτε ενεργοποιούν είτε αποκλείουν δυνατότητες του Server. Η κύρια λειτουργία του είναι η ανακατεύθυνση του χρήστη σε περιπτώσεις όπως 404 error, λάθος στην συμπλήρωση κωδικού και προστασία χρήστη από ευαίσθητο υλικό.

web.config.txt

Το ίδιο αρχείο με το htaccess που είδαμε πάνω, απλώς το συγκεκριμένο λειτουργεί αποκλειστικά στον Microsoft IIS (Internet Information Server).

index.php

Είναι η αρχική σελίδα της διαδικτυακής εφαρμογής, γραμμένη σε PHP γλώσσα, που συναντάμε την πρώτη φορά που ενεργοποιούμε το CMS. Εξελίσσεται αυτόματα ανάλογα με της επιλογές μας στο γραφικό περιβάλλον. Στα περισσότερα templates επιτρέπεται μικρή και επιφανειακή επεξεργασία του αρχείου, καθώς συνήθως είναι η βάση για τις καινούριες σελίδες που δημιουργούμε.

license.txt

Περιέχει την γραπτή νομική άδεια της ομάδας προγραμματιστών του Joomla, που μας επιτρέπει την αντιγραφή, διαμοίραση και επεξεργασία των αρχείων του φακέλου.

robots.txt.dist

Τα Web Robots αρχεία κρατούν στοιχεία της εφαρμογής μας στους φακέλους τους και εξαπλώνονται αυτόματα στο διαδίκτυο αφήνοντας ίχνη στις μηχανές αναζήτησης. Τα συγκεκριμένα αρχεία είναι δημοφιλή στους web developers και ακολουθούν το πρωτόκολλο REP (Robots Exclusion Protocol) που επιτρέπει στον administrator να θέτει ποιά τμήματα της εφαρμογής είναι εντός και εκτός ορίων των robots για λόγους ασφαλείας.

1.5 Τα στάδια ζωής των CMS

Τα CMS κατασκευάστηκαν με την λογική να είναι απλά και κατανοητά σε όλα τα στάδια της δημιουργίας μιας εφαρμογής, ώστε να διευκολύνουν τον web developer και να του επιτρέπουν να ασχοληθεί με την ουσία και όχι με τις λεπτομέρειες του σχεδιασμού. Δεν παύουν όμως να είναι προγράμματα μεγάλου μεγέθους τα οποία χρειάζονται ιδιαίτερη προσοχή κατά την εγκατάσταση και την λειτουργία τους.

Με την επεξήγηση κάθε υποφακέλου και αρχείου του CMS Joomla, ολοκληρώσαμε την έρευνα και την θεωρητική παρουσίαση των Συστημάτων Διαχείρισης Περιεχομένου. Ως γνωστών όμως, είναι αδύνατον να κατανοήσουμε απόλυτα οτιδήποτε στον τεχνολογικό τομέα αν δεν το δούμε στην πράξη. Εξετάζοντας τις πρακτικές λειτουργίες των CMS μπορούμε να διαχωρίσουμε την χρήση τους σε στάδια.

Ο κύκλος ζωής ενός CMS διαχωρίζεται σε τρία στάδια:

- Εγκατάσταση και παραμετροποίηση του Συστήματος
- Δημιουργία διαδικτυακής εφαρμογής
- Συντήρηση, και διαχείριση της εφαρμογής

Παρακάτω ακολουθούν οδηγίες για την κατασκευή εφαρμογών, πάντα με χρήση CMS, με κάθε κεφάλαιο να εστιάζει το αντίστοιχο στάδιο από τα παραπάνω. Ταυτόχρονα θα παρακολουθούμε την κατασκευή του πρακτικού μέρους, δηλαδή της εφαρμογής caferpatras.gr, για επιβεβαίωση της θεωρητικής διαδικασίας.

2. Εγκατάσταση CMS

Όταν έχουμε καταλήξει στον τύπο ιστοσελίδας που θέλουμε να κατασκευάσουμε και στο κατάλληλο CMS, το επόμενο στάδιο είναι να ξεκινήσουμε την εγκατάσταση στον Server της εφαρμογής μας. Σε αυτό το κεφάλαιο θα αναλύσουμε την εγκατάσταση των Συστημάτων Διαχείρισης Περιεχομένου ανάλογα με τον Server που επιλέξαμε, θα δούμε τα βήματα της παραμετροποίησης τους και θα δώσουμε οδηγίες ώστε όταν νοικιάσουμε κάποιο domain name να παρουσιαστεί εκεί το start page του CMS.

2.1 Εγκατάσταση και παραμετροποίηση

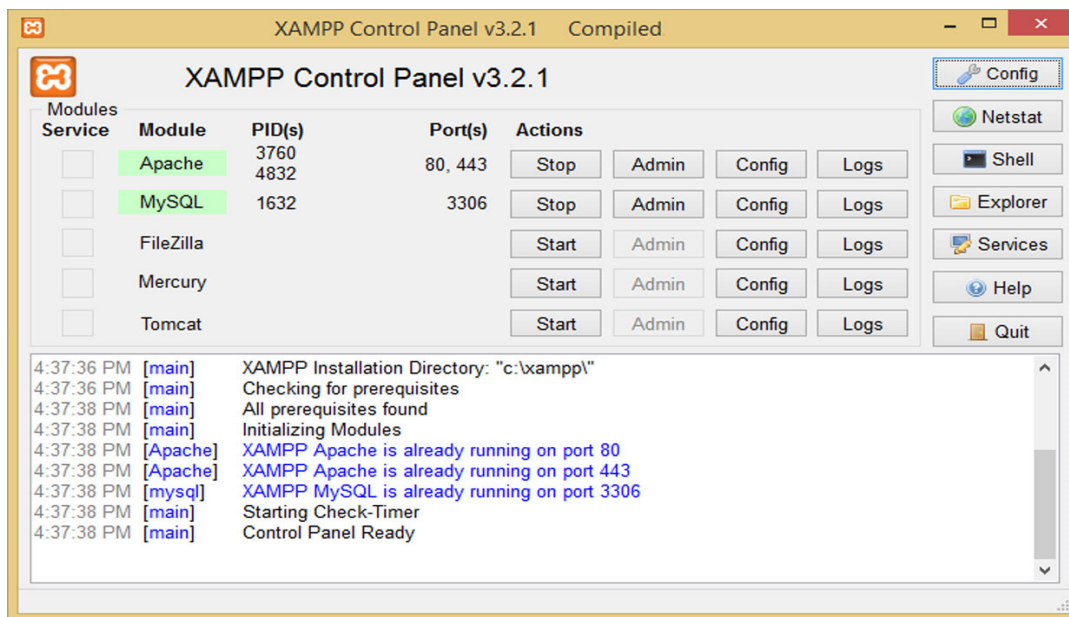
Όπως αναφέραμε και νωρίτερα η εγκατάσταση του CMS πρέπει να γίνει στον server της εφαρμογής. Ο administrator για το host οποιασδήποτε διαδικτυακής εφαρμογής έχει τρεις επιλογές. Η πρώτη είναι να χρησιμοποιήσει κάποιον δικό του υπολογιστή ως τοπικό server και να πραγματοποιήσει εκεί την εγκατάσταση του CMS, η δεύτερη επιλογή είναι να εγκαταστήσει τα αρχεία σε κάποιον δωρεάν server μικρής χωρητικότητας στο διαδίκτυο και η τρίτη επιλογή είναι η ενοικίαση server από κάποια εταιρία web hosting.

2.1.1 Εγκατάσταση σε τοπικό Server

Στην πρώτη περίπτωση το πρώτο βήμα είναι να εγκαταστήσουμε και ενεργοποιήσουμε τοπικά κάποιον server στον υπολογιστή μας. Η καλύτερη και δημοφιλέστερη επιλογή είναι ο Apache Server τον οποίο βρίσκουμε δωρεάν κατεβάζοντας το **Xampp project** από το διαδίκτυο. Το Xampp δημιουργήθηκε από το community apachefriends και προσφέρει μαζί με τον Server και άλλες χρήσιμες παροχές, όπως τις MySQL βάσεις δεδομένων που είναι απαραίτητες για να λειτουργήσει το CMS. Στο ιστότοπο του Xampp υπάρχουν αναλυτικές οδηγίες και video tutorials για την εγκατάσταση και διαχείριση του Server.

Επόμενο βήμα είναι να μπούμε στον ιστότοπο του CMS που επιλέξαμε και να κατεβάσουμε το zip αρχείο που περιέχει τα αρχεία του Συστήματος. Κάθε CMS ανοιχτού κώδικα έχει τον δικό του ιστότοπο και παρέχει δωρεάν το Σύστημα μαζί με οδηγίες εγκατάστασης. Για την εγκατάσταση των αρχείων πρέπει πρώτα να φτιάξουμε έναν καινούριο φάκελο, με το όνομα της εφαρμογής μας, μέσα στον φάκελο htdocs στο Xampp (PC/(C:)/xampp/htdocs). Σε αυτόν τον καινούριο φάκελο κάνουμε extract τον φάκελο του συστήματος που μόλις κατεβάσαμε. Με αυτόν τον τρόπο θα μπορούσαμε αργότερα γραφικά να κάνουμε την σύνδεση του CMS με τον Server και την βάση δεδομένων στην οποία θα αποθηκεύονται τα δεδομένα της εφαρμογής.

Με το Xampp είναι πολύ απλό να δημιουργήσουμε και να διαχειριστούμε μια βάση δεδομένων γραφικά. Για να το κάνουμε, αρχικά ενεργοποιούμε την MySQL από το control panel του Xampp.



Εικόνα 2.1-1 Το Control Panel του Xampp

Έπειτα ανοίγουμε τον browser, πληκτρολογούμε localhost και βρίσκουμε την επιλογή phpMyAdmin από το αριστερό μενού. Από εκεί δημιουργούμε καινούρια βάση δεδομένων μαζί με έναν τουλάχιστον καινούριο χρήστη που θα έχει όλα τα δικαιώματα. Δίνουμε προσοχή στους κωδικούς ασφαλείας για τους χρήστες της βάσης, καθώς σε αυτό το σημείο οι επιλογές μας επηρεάζουν το πόσο ευάλωτη θα είναι η εφαρμογή μας.

Όταν ολοκληρωθεί η δημιουργία της βάσης δεδομένων βρισκόμαστε στο σημείο που πρέπει να την συνδέσουμε με το CMS. Η πρόσβαση στο Σύστημα γίνεται πληκτρολογώντας στον browser localhost/ και κολλητά το όνομα του φακέλου που δημιουργήσαμε νωρίτερα στο htdocs. Εκεί γίνεται η αρχική παραμετροποίηση, συνεπώς είμαστε για άλλη μια φορά ιδιαίτερα προσεκτικοί. Από αυτό το σημείο και έπειτα η διαδικασία αλλάζει ανάλογα με το CMS που επιλέξαμε να εγκαταστήσουμε. Συνήθως ακολουθούν τρία βήματα που ορίζονται ως: configuration, database και overview της εφαρμογής.

Αρχικά στο configuration συνήθως ζητείται να συμπληρώσουμε τα στοιχεία της εφαρμογής μας, όπως όνομα και περιγραφή, και να επιλέξουμε κωδικούς και όνομα για τον administrator / super user. Το επόμενο βήμα είναι να επιλέξουμε την βάση δεδομένων που δημιουργήσαμε ως βάση για τις σελίδες του συστήματος, επιλέγοντας την και δίνοντας τους κωδικούς. Τα πιο περίπλοκα CMS εδώ θα εμφανίζουν σύνθετες ρυθμίσεις για την βάση, όπως για παράδειγμα να ορίσουμε ένα prefix για τους πίνακες που θα παράγονται αυτόματα. Στο τέλος εμφανίζεται το overview της σελίδας όπου ορίζουμε τις λεπτομέρειες για το start page που θα παραχθεί. Όλες οι ρυθμίσεις που θα βρούμε σε αυτό το σημείο μπορούν να αλλάξουν και αργότερα, μέσα από το μενού του CMS, απλώς υπάρχουν εδώ ως διευκόλυνση στους νέους χρήστες, ώστε η αρχική σελίδα που θα δημιουργηθεί να συναντά τις προσδοκίες τους.

Για το πέρας της διαδικασίας επιλέγουμε install, περιμένουμε το χρονικό διάστημα που ορίζει το κάθε CMS, και η εγκατάσταση έχει ολοκληρωθεί. Αν υπάρχει κάποιο σφάλμα

εδώ μας ενημερώσει το CMS, αν όχι μπορούμε πλέον πληκτρολογώντας στον browser localhost/ και το όνομα της εφαρμογής, να δούμε την αρχική σελίδα της εφαρμογής μας. Για να επεξεργαστούμε πλέον την εφαρμογή πρέπει να μπούμε στο γραφικό περιβάλλον πληκτρολογώντας το κατάλληλο path ανάλογα το CMS. Το path για το κάθε σύστημα μπορούμε να το μάθουμε από την σελίδα του στο διαδίκτυο, για παράδειγμα αν ονομάσουμε την εφαρμογή μας MyWebsite, στο Joomla το path που πρέπει να πληκτρολογήσουμε είναι το localhost/MyWebsite/administrator, στο Wordpress το localhost/MyWebsite/wp-login, ενώ στο drupal το localhost/MyWebsite/?q=user.

2.1.2 Εγκατάσταση σε Web Server

Το να έχουμε πάντα διαθέσιμο έναν υπολογιστή που να λειτουργεί σαν Server δεν είναι εύκολο, οπότε προχωράμε στην δεύτερη περίπτωση, όπου χρησιμοποιούμε κάποιον απομακρυσμένο Web Server. Εδώ συναντάμε το δίλλημα: να εγκαταστήσω την εφαρμογή μου σε δωρεάν Server ή να νοικιάσω δικό μου χώρο στον επαγγελματικό Server κάποιας εταιρίας Web Hosting; Η τελική επιλογή εξαρτάται από τις φιλοδοξίες και τις οικονομικές δυνατότητες του ιδιοκτήτη της εφαρμογής.

Για παράδειγμα εάν θέλουμε έναν μικρό ιστότοπο απλώς για να γράφουμε κάποιες απόψεις στο διαδίκτυο χωρίς ιδιαίτερες φιλοδοξίες και με γνώμονα το ελάχιστο κόστος τότε θα επιλέξουμε ένα δωρεάν Server. Η αναζήτηση δεν είναι εύκολη καθώς υπάρχουν αναξιόπιστες εταιρίες που παρέχουν δωρεάν υπηρεσίες web hosting χρησιμοποιώντας όμως την εφαρμογή μας για να παρουσιάζουν διαφημίσεις αγνώστου περιεχομένου. Υπάρχουν και CMS που παρέχουν φιλοξενία της εφαρμογής, αλλά επειδή δεν είναι αυτή η κύρια υπηρεσία τους, η υποστήριξη προβλημάτων αυτών των πελατών περνά σε δεύτερη μοίρα.

Όταν διαχειριζόμαστε μια φιλόδοξη εφαρμογή ή έχουμε να λογοδοτήσουμε σε κάποιον άλλο, η σίγουρη επιλογή είναι να νοικιάσουμε χώρο επί πληρωμή. Ο κύριος λόγος είναι ότι εάν συμβεί οποιαδήποτε δυσλειτουργία στους Servers, ή εμείς σαν administrator δημιουργήσουμε κάποιο πρόβλημα, υπάρχει αξιόπιστο tech support για να μας βοηθήσει. Για κάποιον που ανησυχεί πως θα έχει πρόβλημα επικοινωνίας, υπάρχουν επίσης Ελληνικές εταιρίες Web Hosting με τους Servers εντός Ελλάδας. Αξίζει εδώ να σημειώσουμε πως πολλές εταιρίες που παρέχουν free Web Hosting τοποθετούν ταυτόχρονα στο domain name της εφαρμογής τον τίτλο της εταιρίας τους. Αν θέλουμε να το αποφύγουμε πρέπει να αγοράσουμε το δικό μας domain name και να το συνδέσουμε στον Server.

Η διαδικασία και στις δύο περιπτώσεις είναι παρόμοια όπως θα την δούμε παρακάτω. Αρχικά κάνουμε μια έρευνα αγοράς στο διαδίκτυο ώστε να καταλήξουμε σε όποια εταιρία κρίνουμε κατάλληλη. Μπαίνουμε στον ιστότοπο της εταιρίας και υποχρεωτικά δημιουργούμε καινούριο λογαριασμό. Λίγα λεπτά μετά την δημιουργία λογαριασμού σε εταιρία Web Hosting, λαμβάνουμε ένα e-mail με όνομα χρήστη, κωδικούς και ένα σύνδεσμο. Ακολουθώντας τον σύνδεσμο και πληκτρολογώντας τους κωδικούς βρισκόμαστε στο control panel του Server μας. Εδώ αποθηκεύονται όλες οι πληροφορίες και τα αρχεία της εφαρμογής μας.

Μια επιλογή από τις πολλές που δίνει το μενού του control panel έχει πάντοτε το όνομα applications. Σε αυτή την κατηγορία βρίσκονται οι πιο συχνά εμφανιζόμενες εφαρμογές που πιθανόν θα θέλαμε να εγκαταστήσουμε. Σε αυτή την λίστα, και συνήθως στην υποκατηγορία Web applications, βρίσκουμε τα δημοφιλέστερα CMS έτοιμα για εγκατάσταση με το πάτημα ενός κουμπιού. Εάν υπάρχει το CMS που επιθυμούμε πατάμε την επιλογή εγκατάστασης και διαλέγουμε εάν θέλουμε προεπιλεγμένες ρυθμίσεις ή όχι. Η διαδικασία εγκατάστασης διαρκεί αρκετή ώρα, αλλά όταν ολοκληρωθεί, μένει μόνο η διευκρίνιση του domain name για το URL όπως παρουσιάζεται στις τρεις τελευταίες παραγράφους.

Στην περίπτωση όμως, που το CMS που επιθυμούμε δεν βρίσκεται στην λίστα, τότε η διαδικασία γίνεται πιο περίπλοκη. Θα πρέπει να ανεβάσουμε μόνοι μας τα αρχεία του CMS στον Server για να ολοκληρώσουμε την εγκατάσταση. Για να το πραγματοποιήσουμε αυτό, φεύγουμε από τα applications και βρίσκουμε στο μενού την κατηγορία File Manager. Φυσιολογικά την πρώτη φορά θα αντικρίσουμε μια άδεια λίστα καθώς δεν υπάρχει ακόμα κώδικας για κάποια εφαρμογή. Ταυτόχρονα μπαίνουμε στον ιστότοπο του CMS που έχουμε επιλέξει και κάνουμε download το zip φάκελο που περιέχει το σύστημα.

Πρώτο βήμα στον File Manager είναι να πατήσουμε file upload για να ανεβάσουμε τα αρχεία του συστήματος. Προσοχή όμως, γιατί τα αρχεία δεν πρέπει να είναι συμπιεσμένα, ούτε σε υποφάκελο. Άρα μπορούμε είτε να αποσυμπιέσουμε τοπικά το zip αρχείο και να ανεβάσουμε ξεχωριστά τους φακέλους που περιείχε, είτε καλύτερα, να ανεβάσουμε όλο το zip και να κάνουμε εκείνη τη στιγμή extract τα αρχεία με διεύθυνση προορισμού το File Manager και διαγράφοντας μετά την ολοκλήρωση της διαδικασίας το zip αρχείο. Έπειτα βρίσκουμε στο μενού την κατηγορία databases και δημιουργούμε μια νέα, άδεια βάση δεδομένων μαζί με ένα χρήστη που θα έχει όλα τα δικαιώματα.

Όταν ολοκληρωθεί η βάση δεδομένων προχωρούμε στο τελευταίο βήμα το οποίο είναι να ανοίξουμε το installation page. Για να γίνει αυτό θα πρέπει να γνωρίζουμε το domain name ή αλλιώς το URL που θα έχει η εφαρμογή μας στο διαδίκτυο. Αν αποφασίσαμε να διατηρήσουμε το δωρεάν domain name που μας παρέιχε η εταιρία Web Hosting τότε το link θα το βρούμε στο e-mail που μας στάλθηκε μετά την δημιουργία λογαριασμού. Σε αντίθετη περίπτωση, όταν δηλαδή έχουμε αγοράσει δικό μας domain name, τώρα είναι η κατάλληλη στιγμή να το ενεργοποιήσουμε στον Server.

Βρίσκουμε την κατηγορία που αφορά τα domain names, η οποία συνήθως βρίσκεται αν όχι πρώτη, ψηλά στην λίστα του μενού και επιλέγουμε το add domain. Φτάνουμε έτσι στο τελικό στάδιο της εγκατάστασης, όπου όπως και στην περίπτωση της εγκατάστασης σε τοπικό Server, θα πρέπει να γνωρίζουμε ποιο είναι το installation page του CMS που επιλέξαμε. Συνήθως πρέπει να πληκτρολογήσουμε στον browser το domain name / install. Αλλά δεν είναι δεδομένο ότι αυτό θα είναι το installation page για κάθε CMS, οπότε είναι προτιμότερο να κάνουμε μια αναζήτηση στον ιστότοπο του και να το βρούμε. Όταν πληκτρολογήσουμε το σωστό link μας εμφανίζεται η αρχική σελίδα εγκατάστασης όπου μας ζητούνται ρυθμίσεις ανάλογα την πολυπλοκότητα του κάθε συστήματος. Συμπληρώνουμε βάση των προτιμήσεων μας και, όταν μας ζητηθεί, συνδέουμε το CMS με την βάση δεδομένων που δημιουργήσαμε προηγουμένως.

Με την ολοκλήρωση της εγκατάστασης και έχοντας συνδέσει το domain name με τον Server της εφαρμογής το μόνο που μένει είναι να εισέλθουμε στο γραφικό περιβάλλον του συστήματος. Όπως είπαμε και για την τοπική εγκατάσταση το κάθε σύστημα έχει το δικό του μονοπάτι εισόδου, το οποίο βρίσκουμε στα documentations του. Ενδεικτικά αναφέρουμε πως το path που πρέπει να πληκτρολογήσουμε για είσοδο στο γραφικό περιβάλλον του Joomla είναι το `www.URL.com/administrator`, στο Wordpress το `www.URL.com/wp-login`, ενώ στο drupal το `www.URL.com/?q=user`, με την λέξη URL να αντικαθιστάται ανάλογα το παράδειγμα.

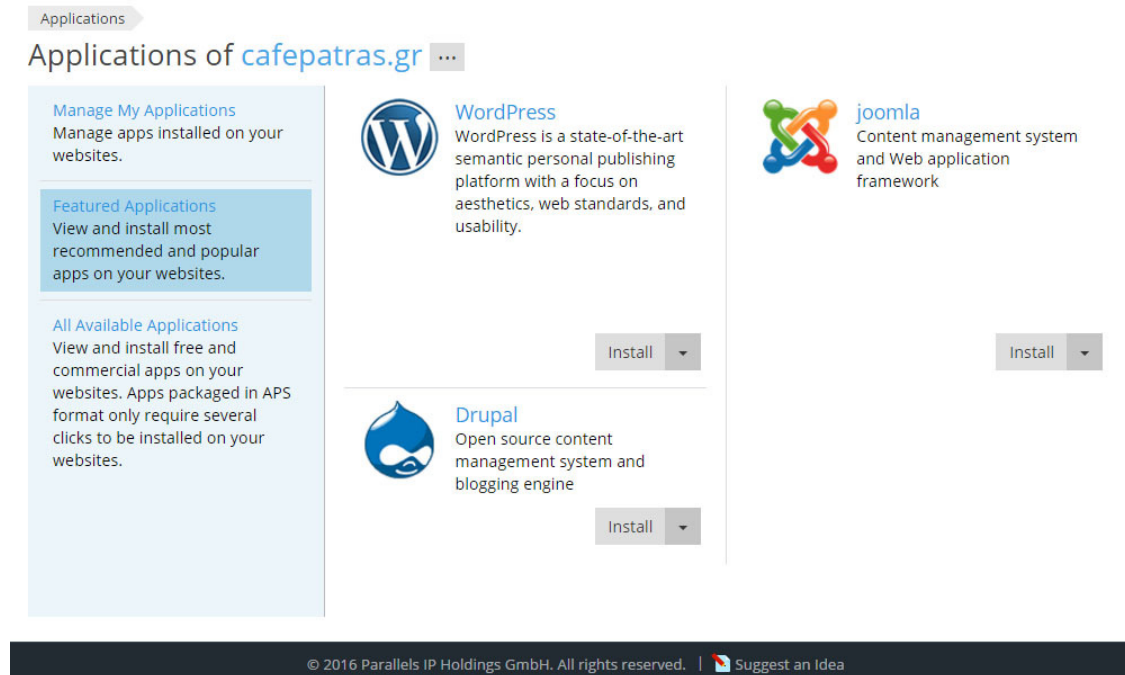
2.2 Το `caferpatras.gr` στην διαδικασία εγκατάστασης

Ξεκινώντας από το μηδέν, πρώτη απόφαση που παίρνουμε για την εφαρμογή είναι το όνομα. Δεν φτάνει μόνο να το επιλέξουμε, πρέπει να είναι και διαθέσιμο, να μην ανήκει δηλαδή ήδη σε άλλον χρήστη. Στην περίπτωση μου, το `caferpatras` ήταν διαθέσιμο με κάθε κατάληξη (.com, .gr, .uk κτλ.) και επέλεξα την Ελληνική καθώς, αρχικά τουλάχιστον, η εφαρμογή θα είναι διαθέσιμη μόνο στα Ελληνικά. Υπάρχουν πολλές εταιρίες που παρέχουν domain name σε διαφορετικές τιμές. Προσωπικά επέλεξα να νοικιάσω το όνομα της εφαρμογής μαζί με χώρο σε Server από την Ελληνική εταιρία Top.Host.

Επιλέγουμε το domain name που θέλουμε να νοικιάσουμε και περιμένουμε το e-mail με κωδικούς που θα χρειαστούν για να το ενεργοποιήσουμε σε κάποιον Server. Από την στιγμή που θα λάβουμε το e-mail χρειάζονται περίπου τρεις ώρες για να ενεργοποιηθεί το domain. Καταλαβαίνουμε ότι ενεργοποιήθηκε, όταν πατώντας το URL δεν βγάλει μήνυμα `site cannot be reached`, αλλά εμφανίσει ένα κείμενο, που απευθύνεται στον administrator, και εξηγεί πως δεν υπάρχει σύνδεση σε κάποιον Server.

Όσο περιμένουμε για ενεργοποίηση του domain name μπορούμε να προχωρήσουμε την διαδικασία του Hosting, η οποία είναι πιο περίπλοκη από άποψη δυνατοτήτων. Έχουμε να επιλέξουμε ανάμεσα σε πολλές προσφορές ανάλογα με τις υπηρεσίες που χρειαζόμαστε. Κάνουμε έρευνα αγοράς και καταλήγουμε σε κάποια εταιρία ανάλογα πάλι με τις απαιτήσεις και τις οικονομικές δυνατότητές μας. Για το `caferpatras.gr` νοικιάσαμε ένα φθινό χώρο που να αντέχει μέτρια κίνηση με δυνατότητες επέκτασης στο μέλλον. Μετά την ολοκλήρωση της ενοικίασης περιμένουμε ξανά e-mail αυτή την φορά με link το οποίο θα μας οδηγήσει στο control panel του Server μας. Ένα από τα πιο διαδεδομένα control panel είναι το CPanel, το οποίο το χρησιμοποιεί και η εταιρία που επιλέξαμε για το πρακτικό μέρος της άσκησης.

Όταν οι υπηρεσίες μας ενεργοποιηθούν, μπαίνω στο control panel επιλέγω το Add Domain που βρίσκεται ψηλά στο μενού και πληκτρολογώ τα στοιχεία του domain name που μόλις νοίκιασα. Ακολουθώντας τις οδηγίες του e-mail και βάζοντας προσεκτικά τους κωδικούς, μέσα σε λίγα δευτερόλεπτα θα υπάρχει νέα καταχώρηση στην λίστα των Domains. Από κάτω της βρίσκονται πολλές επιλογές για διαμόρφωση του domain name. Βρίσκουμε και επιλέγουμε την κατηγορία applications. Εκεί με την σειρά τους, υπάρχουν αμέτρητες βοηθητικές εφαρμογές, όμως το Joomla θα βρίσκεται πάντα στην κατηγορία με τις πιο δημοφιλή.

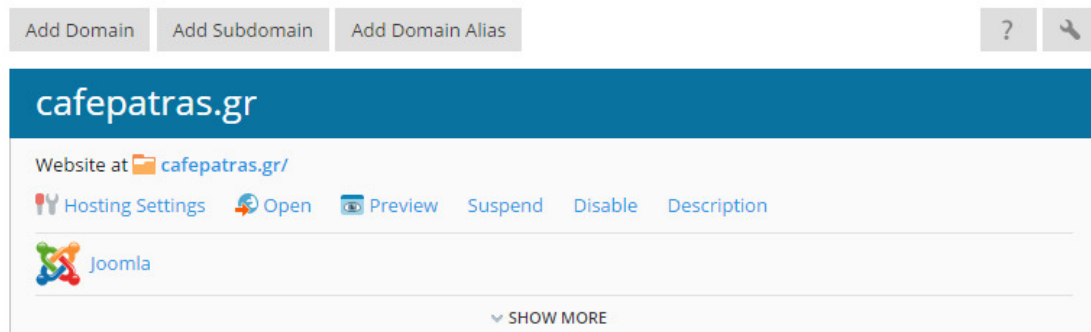


Εικόνα 2.2-1 Δημοφιλή Applications στο μενού του CPanel

Έχοντας νωρίτερα καταλήξει στην απόφαση να εγκαταστήσουμε το Joomla, επιλέγουμε το βελάκι δίπλα στο Install ώστε να κάνουμε Custom Installation με τις δικές μας ρυθμίσεις και όχι τις προεπιλογές. Ένα από τα πλεονεκτήματα του web hosting όταν χρησιμοποιούμε CMS είναι πως η εγκατάσταση είναι πανεύκολη και διαδικασίες που μπορεί να παιδένουν κάποιον αρχάριο γίνονται αυτόματα. Σημαντικότερο παράδειγμα η αυτόματη δημιουργία βάσης δεδομένων στον Server από το Joomla. Όταν η διαδικασία ολοκληρωθεί, στην λίστα των domain names, η εγγραφή θα έχει ενημερωθεί με τα νέα δεδομένα.

Websites & Domains

This is where you set up and manage websites.



Εικόνα 2.2-2 Η ανανεωμένη εγγραφή του cafepatras.gr στην λίστα των Domains

Η επιβεβαίωση πως η εγκατάσταση του Συστήματος ήταν επιτυχής έρχεται όταν πάμε να ανοίξουμε το γραφικό περιβάλλον. Η πρόσβαση στο περιβάλλον του Joomla γίνεται πληκτρολογώντας στον browser το URL της εφαρμογής μας, και κολλητά μαζί /administrator. Αν η διαδικασία πραγματοποιήθηκε σωστά, στην οθόνη μας βλέπουμε την φόρμα εισόδου του Joomla που ζητάει username και password. Για μεγαλύτερη ασφάλεια, με την χρήση ενός plug-in, έχουμε την δυνατότητα να εξελίξουμε την φόρμα εισόδου με περισσότερες επιλογές, όπως επίσης να αλλάξουμε το path με οποιοδήποτε της επιλογής μας.

Πληκτρολογούμε τα στοιχεία μας και πραγματοποιούμε την πρώτη μας είσοδο στο dashboard του CMS. Είμαστε πλέον σε θέση να πούμε πως η διαδικασία της εγκατάστασης πραγματοποιήθηκε και μπορούμε να προχωρήσουμε στην κατασκευή της εφαρμογής μας.

3. Ο ρόλος των templates

Είναι απαραίτητο να γνωρίζουμε καλά το περιβάλλον του CMS, πριν ξεκινήσουμε την κατασκευή της εφαρμογής, γιατί αλλιώς κάποιες σημαντικές δυνατότητες θα μείνουν ανεξερεύνητες. Ο κάθε προγραμματιστής θέλει τον δικό του χρόνο προσαρμογής οπότε δεν υπάρχει συγκεκριμένη διάρκεια περιήγησης πριν ξεκινήσουμε. Αναφέραμε και νωρίτερα πως το Joomla διαθέτει διπλό μενού, κάθετο και οριζόντιο, με αποτέλεσμα να χρειάζεται μεγάλη εξερεύνηση ώστε να νιώθουμε άνετα στο περιβάλλον. Κατά την διάρκεια της περιήγησης μας θα έρθει η στιγμή που θα ανοίξουμε την αρχική σελίδα της άδειας εφαρμογής μας και θα δούμε τι έχει κατασκευάσει αυτόματα το CMS. Υπεύθυνο για το αποτέλεσμα είναι το προεγκατεστημένο template που είναι ενεργό εκείνη την στιγμή.

3.1 Επιλογή template

Το πρώτο τεχνικό βήμα κατά την διαδικασία της εξερεύνησής μας είναι να γυρίσουμε την εφαρμογή μας σε offline mode από τις ρυθμίσεις, ενώ το πρώτο δημιουργικό βήμα είναι η επιλογή του κατάλληλου θέματος για την εφαρμογή μας. Αν η δουλειά κάποιου σταματά με την κατασκευή της εφαρμογής, και δεν τον αφορούν τα κείμενα και η συντήρηση, τότε με την σωστή εγκατάσταση και παραμετροποίηση του template ξεκινά και ολοκληρώνει την εργασία του.

3.1.1 Αναζήτηση από υπάρχοντα

Υπάρχουν CMS, με καλύτερο παράδειγμα το Wordpress, όπου στον Template Manager υπάρχει δυνατότητα αναζήτησης θεμάτων σε μια λίστα που περιέχει όλα όσα είναι διαθέσιμα, είτε επί πληρωμή, είτε όχι. Στο Joomla αντίθετα, η αναζήτηση θεμάτων γίνεται στις μηχανές αναζήτησης, όπου επισκεπτόμαστε τις σελίδες των προγραμματιστών που τα κατασκευάζουν και κατεβάζουμε το συμπιεσμένο αρχείο με το template. Για να δούμε το θέμα που επιλέξαμε στην εφαρμογή μας, πρέπει να κάνουμε upload το αρχείο από τον Template Manager και μετά να το ενεργοποιήσουμε από την λίστα των templates. Το τελικό template που θα επιλέξουμε εξαρτάται σε μεγάλο βαθμό από τις δυνατότητες μας στις γλώσσες προγραμματισμού. Για παράδειγμα, ένας αρχάριος που κατασκευάζει μια ιστοσελίδα για προώθηση ενός ξενοδοχείου θα αναζητήσει στον ιστό τα καλύτερα θέματα για ξενοδοχείο. Έπειτα μέσα από δεκάδες αποτελέσματα, θα πρέπει να επιλέξει πιο θέμα ταιριάζει καλύτερα στα γούστα του και έχει θετικά σχόλια, ώστε να μην προκύψουν αργότερα προβλήματα.

Σε περίπτωση που εμπιστευόμαστε τις δυνατότητες μας για επεξεργασία κυρίως PHP κώδικα σε template, μπορούμε να κάνουμε γενικότερες αναζητήσεις όπως τα καλύτερα θέματα για επιχειρήσεις, και να τα προσαρμόσουμε αργότερα. Άλλωστε δεν πρέπει να ξεχνάμε την δύναμη των extensions και τις παροχές που μας προσφέρουν. Με αυτή την λογική όμως, μπορούμε να το πάμε και ένα βήμα παραπέρα και να κατεβάσουμε ένα

blank template. Τα συγκεκριμένα templates περιέχουν τα αρχεία που είναι απαραίτητα για την λειτουργία και την προστασία της εφαρμογής, αλλά τίποτα παραπάνω. Το CSS αρχείο, το οποίο είναι υπεύθυνο για το design της σελίδας, είναι άδειο, ενώ δεν υπάρχει κανένας σχεδιασμός γραμμένος σε HTML5. Αυτό δίνει στον administrator την ελευθερία να απελευθερώσει την δημιουργικότητα του απερίσπαστα, όπως θα έκανε τοπικά σε έναν editor, αλλά χωρίς να χάνει τα πλεονεκτήματα που το προσφέρει το CMS στον σχεδιασμό.

Με τα δωρεάν blank templates όμως, προκύπτουν κάποια ερωτήματα. Δεν περιέχουν design με αποτέλεσμα να μην μπορούμε να δούμε κάποιο demo τους, άρα πως μπορούμε εξαρχής να γνωρίζουμε εάν είναι αξιόπιστα; Είναι όντως απαραίτητοι οι τόσσοι φάκελοι που περιέχουν, απλώς για να διατηρείται η δομή της σελίδας; Και κυρίως, από την στιγμή που είμαστε σε θέση να πραγματοποιήσουμε το design της σελίδας, είναι δύσκολο για εμάς να κατασκευάσουμε και την δομή του template;

3.1.2 Κατασκευή βασικού template για Joomla

Θέλουμε να εξετάσουμε ποιες είναι οι ελάχιστες απαιτήσεις του Joomla, ώστε να αποδεχτεί και να αναγνωρίσει ένα δικό μας, σχεδιαστικά άδειο, template. Για να το κάνουμε αυτό, δημιουργήσαμε ένα template, για το caferpatras.gr, που περιέχει μόνο τους βασικούς φακέλους και τα υποχρεωτικά αρχεία. Η δομή των αρχείων πρέπει να είναι όπως φαίνεται παρακάτω, όπου οι φάκελοι παρουσιάζονται με έντονη γραμματοσειρά.

```
taza
├── css
│   └── mystyle.css
├── images
│   └── logo.png
├── index.php
└── templateDetails.xml
```

Ο αρχικός φάκελος θα διαθέτει και το όνομα του template που θα δημιουργήσουμε, όπου για την περίπτωση του caferpatras.gr είναι taza, (στα Ισπανικά σημαίνει κούπα του καφέ). Μέσα δημιουργήσαμε δύο φακέλους, τον `css` και τον `images`, κάτι που δεν είναι απαραίτητο για να δουλέψει το template, αλλά βοηθάει και εμάς στην ταξινόμηση αρχείων και το σύστημα στην ταχύτητα που θα απαντάει στα αιτήματα. Και οι δύο φάκελοι μπορούν είτε να παραμείνουν κενοί, είτε να περιέχουν πολλά αρχεία με `css` κώδικα και εικόνες που αφορούν την εφαρμογή μας αντίστοιχα.

Από την στιγμή που οι δύο φάκελοι είναι προαιρετικοί, καταλαβαίνουμε πως ολόκληρο το template στηρίζεται σε δύο και μόνο αρχεία, το `index.php` και το `templateDetails.xml`. Ας δούμε αρχικά το δεύτερο αρχείο, το οποίο έχει κατάληξη `xml` (Extensible Markup Language) η οποία είναι μια γλώσσα κατανοητή και από ανθρώπους και από υπολογιστές. Είναι το αρχείο το οποίο παρουσιάζει στο Joomla το template και τα

περιεχόμενα του. Η σύνταξη του για τις εκδόσεις 2.5 και πάνω του Joomla πρέπει να είναι όπως εμφανίζεται παρακάτω:

➤ Το αρχείο templateDetails.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <extension version="2.5" type="template">
3     <name>taza</name>
4     <creationDate>03-11-2016</creationDate>
5     <author>Tarantilis Antonis</author>
6     <authorEmail>antonis@tarantilis.com</authorEmail>
7     <authorUrl>http://www.tarantilis.com</authorUrl>
8     <copyright> Copyright © 2015 tarantilis.com </copyright>
9     <license>GNU/GPL</license>
10    <version>1.0.0</version>
11    <description> <![CDATA[
12        <h1>taza!</h1>
13        <p>Created by <a href="http://www.tarantilis.com"
14    target="_blank">Antonis Tarantilis | tarantilis.com</a>.</p>
15    ]]></description>
16    <files>
17        <filename>index.php</filename>
18        <filename>templateDetails.xml</filename>
19        <folder>images</folder>
20        <folder>css</folder>
21    </files>
22    <positions>
23        <position>breadcrumb</position>
24        <position>left</position>
25        <position>right</position>
26        <position>top</position>
27        <position>user1</position>
28        <position>user2</position>
29        <position>user3</position>
30        <position>user4</position>
31        <position>footer</position>
32    </positions>
33 </extension>
```

Για την επεξήγηση του κώδικα τον χωρίσαμε σε τέσσερα νοητά κομμάτια από όπου ξεκινάει η κόκκινη αρίθμηση. Αρχικά, στην πρώτη γραμμή, έχουμε την δήλωση του xml αρχείου που περιέχει την έκδοση της γλώσσας και την μέθοδο κωδικοποίησης του αρχείου. Στη δεύτερη γραμμή, δηλώνουμε την έκδοση του Joomla, και το τύπο του

αρχείου που κατασκευάζουμε. Αυτές οι δύο εντολές παραμένουν σταθερές για κάθε template.

Στην συνέχεια από την γραμμή 3 έως την 15, ακολουθεί η περιγραφή του θέματος. Δεν είναι απαραίτητες όλες αυτές οι δηλώσεις, παρόλα αυτά τις έβαλα για να δείξω την ολοκληρωτική περιγραφή. Γράφουμε xml κώδικα οπότε δεν χρειάζεται κάποιο tag για τα κείμενα, εκτός από το description που περιέχει tags, όπως το **<h1>** και παρουσιάζεται με διαφορετικό χρώμα από τον editor. Στο description τοποθέτησα HTML κώδικα, καθώς ήθελα, πρώτον, να δηλώσω ως κεφαλίδα το όνομα του template, αλλά κυρίως, δεύτερον, να προσθέσω ένα link στην πρόταση. Αυτό το επιτυγχάνουμε με την χρήση της εντολής **<![CDATA [...]]>** η οποία δηλώνει πως ο κώδικας μέσα στις αγκύλες δεν πρέπει να μεταφραστεί από τον browser ως xml.

Στην γραμμή 16 βρίσκεται το tag **<files>** στο οποίο πρέπει να δηλώσουμε κάθε φάκελο και κάθε αρχείο που θα περιέχεται στο template μας. Όταν τα αρχεία βρίσκονται μέσα σε φάκελο αρκεί να δηλώσουμε μόνο τον φάκελο ως **<folder>**, όπως για παράδειγμα στην περίπτωση του images όπου δεν δηλώσαμε το αρχείο εικόνας png. Τα αρχεία, ασχέτως τύπου, δηλώνονται με το tag **<filename>**. Υπάρχουν πολλά ακόμα αρχεία που μπορούμε να ανεβάσουμε για να εμπλουτίσουμε το template μας, αλλά θα τα προσθέσουμε αργότερα, στο γραφικό περιβάλλον για να ελαχιστοποιήσουμε την πιθανότητα λάθους. Σε περίπτωση που κάνουμε λάθος είτε προγραμματιστικό στον κώδικα, είτε γραμματικό στην δήλωση ονομάτων, το template θα απορριφθεί με error από το Joomla και πολλές φορές η περιγραφή του λάθους δεν μας βοηθάει να το εντοπίσουμε. Οπότε σε αυτό το σημείο, στόχος μας είναι να δημιουργήσουμε ένα μικρό και συμπαγές template.

Το τελευταίο κομμάτι του αρχείου, από την γραμμή 22 και κάτω, δηλώνει τα **positions**, τα οποία είναι σημεία στην σελίδα ώστε να ορίζουμε αυστηρά πλαίσια σε ένα extension ή σε ένα κομμάτι της εφαρμογής μέσω του κώδικα. Για παράδειγμα, το πιο προφανές position, που είναι το footer, μας εξασφαλίζει πως όταν σε οποιοδήποτε component, plug-in ή module ορίσουμε την συγκεκριμένη θέση, αυτό θα πάει αυτόματα εκεί, στο κάτω μέρος της σελίδας. Τα positions δεν είναι απαραίτητα για να λειτουργήσει το template μας, αλλά τα προσθέτω για την ομαλή μετάβαση από το υπάρχων template στο δικό μας.

Από την προεπιλογή, κατά πάσα πιθανότητα, θα υπάρχουν κάποια extensions ή κάποιες φόρμες στην εφαρμογή μας, στις οποίες θα έχουν δηλωθεί positions. Για να μην απορυθμιστεί τελείως η δομή της σελίδας τα δηλώνουμε και στο δικό μας template, έστω προσωρινά, εκτός και αν έχουμε σκοπό να απενεργοποιήσουμε όλα τα προϋπάρχοντα components. Πρέπει επίσης να αναφέρουμε πως τα positions δεν λειτουργούν απλά επειδή δηλώθηκαν στο xml αρχείο. Είναι υποχρεωτικό να τους δηλωθεί θέση, όπως θα δούμε και παρακάτω, μέσα στο body του index.php.

Κλείνοντας το tag extension έχουμε ολοκληρώσει την δήλωση των περιεχόμενων του template μας. Το αποθηκεύουμε με την κατάλληλη xml και προχωρούμε στην κατασκευή του index.php.

➤ Το αρχείο index.php

Το index αρχείο είναι ο πυρήνας για κάθε καινούρια σελίδα που θα δημιουργούμε μέσω του γραφικού περιβάλλοντος στο CMS. Με απλά λόγια, κάθε εντολή που χρησιμοποιούμε εδώ θα αφορά ολόκληρη την εφαρμογή, είτε το αποτέλεσμα της είναι φανερό στον χρήστη, όπως μια εικόνα, είτε είναι αντιληπτό μόνο από τους browsers, όπως η προσθήκη της βιβλιοθήκης jquery. Είναι αντιληπτό πως σε αυτό το σημείο ο κώδικας διαφέρει ανάλογα την εφαρμογή που θέλουμε να κατασκευάσουμε. Υπάρχουν όμως κάποιες εντολές που πάντα παραμένουν σταθερές και αποτελούν την ραχοκοκαλιά του αρχείου, όπως παρουσιάζονται παρακάτω:

```
1 <?php defined('_JEXEC') or die('Restricted access');?>
2 <!DOCTYPE html>
3 <html xml:lang="<?php echo $this->language; ?>" lang="<?php
4     echo $this->language; ?>" >
5     <head>
6         <jdoc:include type="head" />
7         <link rel="stylesheet" href="<?php echo $this->baseurl
8             ?>/templates/<?php echo $this->template
9             ?>/css/template.css" type="text/css" />
10    </head>
11    <body>
12        <jdoc:include type="modules" name="top" />
13        <jdoc:include type="component" />
14        <jdoc:include type="modules" name="footer" />
15    </body>
16 </html>
```

Ξανά χωρίζουμε νοητά τον κώδικα μας σε τρία μέρη, όπου υπάρχει κόκκινη αρίθμηση, για ευκολότερη επεξήγηση. Σε αυτή την περίπτωση, ο διαχωρισμός είναι ακόμα πιο προφανής, καθώς οι αρχικές δηλώσεις, το head και το body είναι τελείως ανεξάρτητα κομμάτια κώδικα.

Ξεκινάμε με μια βασική και δοκιμασμένη, PHP εντολή, για την προστασία του αρχείου από κακόβουλους χρήστες. Ακολουθεί, στην δεύτερη σειρά, το **DOCTYPE** (Document Type Declaration) με το οποίο δηλώνουμε στον browser και στους web crawlers, την έκδοση της HTML γλώσσας που χρησιμοποιούμε. Γράφοντας δίπλα στο DOCTYPE σκέτο html, δηλώνουμε πως το αρχείο περιέχει κώδικα HTML5, με όποιες συνέπειες έχει αυτό σε παλιότερες εκδόσεις browser (κυρίως σε explorer 8 και κάτω), εάν δεν πάρουμε προφυλάξεις. Στο τέλος ακολουθεί το html tag, που θα περιέχει τον κώδικα, και μέσα του δηλώσαμε την γλώσσα που χρησιμοποιεί η σελίδα. Χρησιμοποιήσαμε μια PHP εντολή για να τραβήξουμε την επιλογή της γλώσσας από την επιλογή που θα γίνει στο γραφικό περιβάλλον.

Επόμενο κομμάτι είναι το head της σελίδας το οποίο ξεκινά με ένα jdoc statement. Αυτό είναι πάγια τακτική στα Joomla templates, που χρησιμοποιούν συχνά την εντολή `<jdoc:include />` για να δηλώσουν την θέση κάποιου extension ή κάποιας ρύθμισης μέσω του γραφικού περιβάλλοντος. Στην δική μας περίπτωση, με το **type="head"**, δηλώνουμε πως εδώ θα εμφανιστούν οι ρυθμίσεις για το head που επιλέξαμε, όπως για παράδειγμα, τίτλος σελίδας, metadata ή κάποιο javascript. Έπειτα στο head βλέπουμε μια απλή εντολή, η οποία καλεί το css αρχείο σχεδίασης για τις σελίδες της εφαρμογής. Για να είναι γενικότερο το παράδειγμα, δεν γράψαμε το path για το css αρχείο που χρησιμοποιήσαμε στο caferatras.gr, αλλά βάλαμε δύο PHP echo εντολές, οι οποίες σε σημεία του path καλούν το URL της εφαρμογής και το όνομα του template αντίστοιχα. Με αυτόν τον τρόπο το head που δημιουργήσαμε, με τις συγκεκριμένες εντολές, λειτουργεί σε όλα τα Joomla templates.

Στο τέλος βλέπουμε το body της σελίδας, όπου ουσιαστικά περιέχει το output της σελίδας και το τι θα βλέπει ο τελικός χρήστης, όταν ανοίγει την εφαρμογή. Δεν πρέπει να ξεχνάμε πως το index αρχείο, θα είναι η βάση για όλες τις σελίδες, άρα θέλουμε μια εντολή που να περιέχει τα κείμενα που θα βάζουμε σε κάθε καινούρια σελίδα που θα δημιουργούμε. Αυτή η εντολή είναι το `<jdoc:include type="component" />` το οποίο, πολύ απλά, θα εμφανίσει ότι αποθηκεύσαμε στον editor. Αναφέραμε και νωρίτερα τον τεράστιο ρόλο των components στα CMS και αυτή εδώ η εντολή αποτελεί την καλύτερη απόδειξη. Για επιπλέον περιεχόμενο συμπεριέλαβα και άλλα δύο jdoc, αυτή την φορά τύπου module με όνομα top και footer, με σκοπό να λειτουργούν ως header και footer, και να περιέχουν ότι τους δηλώσουμε γραφικά. Για παράδειγμα, ένα λογικό βήμα είναι να δημιουργήσουμε το μενού για την εφαρμογή μας και να του δηλώσουμε position=top. Έτσι εύκολα θα εμφανιστεί στο σημείο κώδικα που δηλώσαμε.

Κλείνοντας το html tag, το αρχείο μας έχει πλέον τελειώσει και είναι σε θέση να παρουσιάσει νέες σελίδες που δημιουργούμε. Προσπάθησα να το κρατήσω εξαιρετικά απλό και κατανοητό σε απλούς χρήστες, παρόλα αυτά υπάρχουν κάποια προαιρετικές λεπτομέρειες, που αφού αποφασίσαμε να κατασκευάσουμε το θέμα από το μηδέν, είναι ευκαιρία να τις συμπεριλάβουμε.

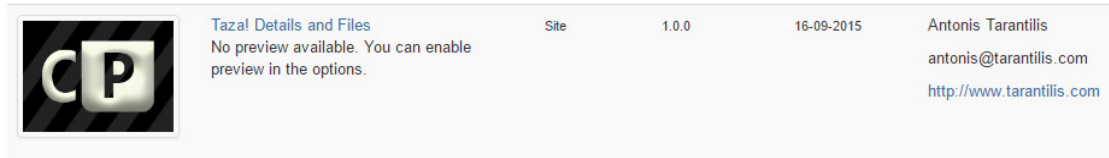
3.1.3 Προαιρετικές λειτουργίες

Όπως αναφέραμε και νωρίτερα, ότι κατασκευάσαμε μέχρι στιγμής λειτουργεί, απλώς έχουμε την δυνατότητα με κάποιες απλές κινήσεις να εμπλουτίσουμε το template και να το κάνουμε πιο επαγγελματικό. Για αρχή θα μιλήσουμε για την προσθήκη κάποιων αρχείων. Πέρα από το index και το templateDetails υπάρχουν και άλλα αρχεία που δεν είναι απαραίτητα για την εκκίνηση της εφαρμογής, αλλά είναι συνεισφέρουν στην ομαλή λειτουργία της.

➤ Εικόνες παρουσίασης του template

Ας ξεκινήσουμε από τις πιο απλές λεπτομέρειες. Για την σωστή εμφάνιση του θέματος μας στο περιβάλλον του Joomla, μπορούμε να προσθέσουμε δύο εικόνες οι οποίες θα εμφανίζονται ως περιγραφή του θέματος μας, στην λίστα με τα διαθέσιμα templates. Οι

δύο αυτές εικόνες είναι απαραίτητο να ονομαστούν `template_preview` και `template_thumbnail` και συνηθίζεται να έχουν διαστάσεις 800x600 και 206x150 αντίστοιχα. Δεν χρειάζεται κάποια σύνδεση, οι εικόνες θα τοποθετηθούν αυτόματα και το θέμα μας στη λίστα θα μοιάζει κάπως έτσι:



Εικόνα 3.1-1 Το template στην λίστα του Template Manager

Η εικόνα που φαίνεται είναι το `template_thumbnail` που επιλέξαμε, και πατώντας πάνω του ανοίγει ένα το παράθυρο με τα details της σελίδας, όπου βρίσκεται και η άλλη εικόνα που φτιάξαμε, με τις μεγάλες διαστάσεις. Να αναφέρουμε επίσης πως τα στοιχεία που εμφανίζονται στο δεξί μέρος της λίστας, είναι τα στοιχεία που δηλώσαμε στο `templateDetails.xml`.

➤ Βιβλιοθήκη JQuery

Κατά την διάρκεια της κατασκευής, είναι πιθανό να κάνουμε χρήση Javascript κώδικα για να κάνουμε την εφαρμογή μας δυναμική και να αλληλεπιδρά με τον χρήστη. Με αυτή τη λογική είναι απαραίτητο να συμπεριλάβουμε στο template μας την βιβλιοθήκη JQuery. Η συγκεκριμένη βιβλιοθήκη περιέχει συναρτήσεις που απλοποιούν σε τεράστιο βαθμό την δημιουργία μιας ολοκληρωμένης και δυναμικής διαδικτυακής εφαρμογής. Τα σημαντικότερα πλεονεκτήματα της είναι: αμέτρητες επιλογές για animation των στοιχείων της εφαρμογής, εύκολο χειρισμό συναρτήσεων AJAX, ευκολία στην responsive μετατροπή της εφαρμογής σε διαφορετικές οθόνες, και τέλος, το απαραίτητο event handling των περιεχομένων.

Για να μπορούμε να χρησιμοποιούμε τις συναρτήσεις της βιβλιοθήκης έχουμε δύο επιλογές. Μπορούμε είτε να τοποθετούμε σε κάθε script που περιέχει Javascript ένα link που να δείχνει τον ιστότοπο που να περιέχει τον κώδικα, είτε, καλύτερα, να αποθηκεύσουμε ολόκληρη την βιβλιοθήκη με την τελευταία έκδοση στο template μας. Το δεύτερο μας εγγυάται γρήγορη απόκριση του δυναμικού κώδικα, χωρίς την ανασφάλεια που περιέχεται σε οποιοδήποτε link δείχνει εκτός των φακέλων στον server μας.

Δημιουργούμε, λοιπόν, έναν νέο φάκελο στο template μας, με όνομα Javascript ή οτιδήποτε παραπέμπει σε αυτήν, όπως για παράδειγμα το `js`. Κατεβάζουμε την βιβλιοθήκη από τον ιστότοπο της στο `jquery.com` και μέσω του editor αποθηκεύουμε τα περιεχόμενα σε ένα νέο αρχείο με όνομα `jquery.js`. Το νέο αρχείο το τοποθετούμε στον `js` φάκελο, ο οποίος θα περιέχει αυτό και οποιοδήποτε άλλο Javascript αρχείο χρειαστούμε στο μέλλον.

➤ 404 error page

Ένα άλλο αρχείο που είναι χρήσιμο για την τελική εμφάνιση της εφαρμογής μας είναι το error.php. Παρακάτω ακολουθεί το error.php που προτείνεται από το community του Joomla για την διαχείριση του σφάλματος 404 στον ιστότοπο μας.

```
/*1*****  
<?php defined ('_JEXEC') or die;  
if (!isset($this->error))  
{  
    $this->error = JError::raiseWarning(404,  
JText::_('JERROR_ALERTNOAUTHOR')); $this->debug = false;  
}  
$doc          = JFactory::getDocument();  
$app          = JFactory::getApplication();  
$this->language = $doc->language;  
$this->direction = $doc->direction; ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml: lang="<?php echo  
$this->language; ?>" lang="<?php echo $this->language; ?>"  
dir="<?php echo $this->direction; ?>">  
  
/*2*****  
<head>  
    <meta http-equiv="content-type" content="text/html;  
charset=utf-8" />  
    <title> <?php echo $this->error->getCode(); ?> - <?php echo  
htmlspecialchars ($this->error->getMessage(), ENT_QUOTES, 'UTF-  
8'); ?> </title>  
    <link rel="stylesheet" href="<?php echo $this-> baseUrl;  
?>/templates/css/mystyle.css" type="text/css" />  
</head>  
  
/*3*****  
<body>  
    <div class="error">  
        <div id="outline">  
            <div id="errorboxoutline">
```

```

        <div id="errorboxheader"> <?php echo $this-
>error->getCode(); ?> - <?php echo htmlspecialchars($this->error-
>getMessage(), ENT_QUOTES, 'UTF-8'); ?> </div>
        <div id="errorboxbody">
            <p> <strong> <?php echo
JText::_('JERROR_LAYOUT_NOT_ABLE_TO_VISIT'); ?></strong> </p>
            <ol>
                <li> <?php echo
JText::_('JERROR_LAYOUT_AN_OUT_OF_DATE_BOOKMARK_FAVOURITE'); ?>
</li>
                <li> <?php echo
JText::_('JERROR_LAYOUT_SEARCH_ENGINE_OUT_OF_DATE_LISTING'); ?>
</li>
                <li> <?php echo
JText::_('JERROR_LAYOUT_MIS_TYPED_ADDRESS'); ?> </li>
                <li> <?php echo
JText::_('JERROR_LAYOUT_YOU_HAVE_NO_ACCESS_TO_THIS_PAGE'); ?>
</li>
                <li> <?php echo
JText::_('JERROR_LAYOUT_REQUESTED_RESOURCE_WAS_NOT_FOUND'); ?>
</li>
                <li> <?php echo
JText::_('JERROR_LAYOUT_ERROR_HAS_OCCURRED_WHILE_PROCESSING_YOUR_
REQUEST'); ?> </li>
            </ol>
            <p><strong> <?php echo
JText::_('JERROR_LAYOUT_PLEASE_TRY_ONE_OF_THE_FOLLOWING_PAGES');
?> </strong></p>
            <ul>
                <li> <a href="<?php echo $this->
baseurl;?>/index.php" title="<?php echo
JText::_('JERROR_LAYOUT_GO_TO_THE_HOME_PAGE'); ?>"><?php echo
JText::_('JERROR_LAYOUT_HOME_PAGE'); ?></a> </li>
            </ul>
            <p> <?php echo
JText::_('JERROR_LAYOUT_PLEASE_CONTACT_THE_SYSTEM_ADMINISTRATOR')
; ?> </p>
            <div id="techinfo">
                <p> <?php echo htmlspecialchars($this->error-
>getMessage(), ENT_QUOTES, 'UTF-8'); ?> </p>
                <p> <?php if ($this->debug) : ?>
                    <?php echo $this->renderBacktrace(); ?>
                    <?php endif; ?>
                </p>

```



```

        </div>
    </div>
</div>
</div>
</div>
</body>
</html>

```

Παρατηρούμε πως το αρχείο αποτελείται σχεδόν αποκλειστικά από εντολές `php echo` ώστε οι ρυθμίσεις να γίνονται από το γραφικό περιβάλλον και να μπορεί να τοποθετηθεί σε οποιοδήποτε `template`. Μόνο στο `head` περιέχεται ένα `path`, για το `css` αρχείο μας, το οποίο πρέπει να σιγουρευτούμε πως έχει το ίδιο όνομα με αυτό που δηλώσαμε. Για καλύτερη κατανόηση, χώρισα το αρχείο σε τρία κομμάτια, με χρήση σχολίων, εκεί όπου αλλάζει η λειτουργία των εντολών. Επειδή όμως πρόκειται για προαιρετικό αρχείο ας κάνουμε μια βασική επεξήγηση των εντολών.

Στο πρώτο κομμάτι του αρχείου βλέπουμε κάποιες εντολές που δεν έχουμε ξανασυναντήσει, οπότε θα ασχοληθούμε αποκλειστικά με αυτές. Μέσα σε μια συνάρτηση `if` παρουσιάζεται η εντολή `isset` η οποία υποστηρίζεται από την νέα έκδοση `PHP 7`, και η λειτουργία της είναι να καθορίζει εάν μια μεταβλητή έχει τιμή, άρα δεν είναι `NULL`. Στην δική μας περίπτωση, χρησιμοποιώντας το θαυμαστικό πριν την εντολή `isset`, εάν δεν υπάρχει `error` θα επιστρέψει `false` και ο κώδικας της σελίδας δεν θα εμφανιστεί. Έπειτα βλέπουμε κάποιες δηλώσεις μεταβλητών, οι οποίες παίρνουν τιμές από το **JFactory**, το οποίο μαζί με το **JError** που συναντήσαμε μέσα στην `if`, είναι κάτι σαν αποθήκες αντικειμένων. Αυτό σημαίνει πως τα `$doc` και `$app` είναι πλέον αντικείμενα, τα οποία έχουν τις δικές τους μεταβλητές και μπορούμε να χρησιμοποιήσουμε, αν το επιθυμούμε, κατά την διάρκεια του κώδικά μας.

Στο δεύτερο κομμάτι έχουμε μόλις τρεις εντολές μέσα στο `head`. Αρχικά το `http-equiv="content-type"`, το οποίο χρησιμοποιείται πολύ συχνά για τον καθορισμό κωδικοποίησης των αρχείων. Έπειτα ακολουθεί ο τίτλος της σελίδας ο οποίος θα εμφανίζεται ψηλά στο παράθυρο του `browser`, και περιγράφει το περιεχόμενο της σελίδας, ανάλογα με το τι επιλέξαμε να εμφανίζεται μέσα από το γραφικό περιβάλλον. Τέλος προφανώς ακολουθεί η σύνδεση του `error.php` με το `css` αρχείο σχεδιασμού των σελίδων. Εδώ πρέπει να είμαστε προσεχτικοί να υπάρχει σωστή σύνδεση και να μην κάνουμε λάθος στο `path`. Κάποιοι προγραμματιστές προτιμούν να κατασκευάζουν ξεχωριστό `css` αρχείο για τον σχεδιασμό του `error page`, κάτι που είναι λογικό όταν το αρχικό μας αρχείο παίρνει μεγάλες διαστάσεις.

Ακολουθεί το μεγαλύτερο, αλλά και απλούστερο, κομμάτι κώδικα όπου περιέχει ότι τελικά εμφανίζεται στον χρήστη. Έχουμε κάποια `div` τα οποία περιέχουν μια λίστα με επεξηγήσεις για πιθανά προβλήματα που οδήγησαν στο `404 error`, τα οποία πρέπει και να σχεδιαστούν μέσω του `css`. Μέσα στην λίστα συναντάμε πολλά **JText**, τα οποία εύκολα καταλαβαίνουμε πως είναι υπεύθυνα να μεταφέρει τα κείμενα που έχουμε δηλώσει γραφικά για κάθε περίπτωση που περιγράφεται στις παρενθέσεις.

Προφανώς αυτό είναι το ενδεικτικό `error page` που προτείνει το `Joomla`. Εγώ προσωπικά δεν το χρησιμοποιώ για την εφαρμογή καθώς σπάνια ο χρήστης ικανοποιείται με την παρουσίαση της λίστας προβλημάτων. Μπορούμε να διαγράψουμε τα πάντα από το τρίτο

κομμάτι κώδικα και να σχεδιάσουμε αυτό που θέλουμε εμείς να εμφανιστεί, ώστε ακόμα και αυτό το σημείο της εφαρμογής να έχει την δική μας, ξεχωριστή προσέγγιση. Σπάνια θα βρούμε πλέον μεγάλη εφαρμογή, που να μην κρύβει κάτι ενδιαφέρον στο 404 error page.

➤ Fields

Προηγουμένως, στο `templateDetails.xml` προσθέσαμε τα `positions`, διότι παρότι είναι προαιρετικά, είναι πολύ απλά στην γραφή τους και βοηθούν στην ομαλή μετάβαση από το προηγούμενο `template` στο δικό μας. Υπάρχει όμως, μια ακόμα δυνατότητα για προσθήκη επιπλέον επιλογών, μέσω αυτού του αρχείου, η οποία κάνει το θέμα μας πιο δυναμικό. Το Joomla μας δίνει την επιλογή να κατασκευάσουμε ένα καινούριο `tab`, στο γραφικό περιβάλλον, το οποίο θα διαθέτει κάποιες βασικές σχεδιαστικές ρυθμίσεις του θέματος και θα μας επιτρέπει να αλλάζουμε ολόκληρο το `design` της εφαρμογής με το πάτημα λίγων κουμπιών.

Για να το κάνουμε αυτό, πρέπει να τοποθετήσουμε στο `xml` αρχείο μας, κάτω από τα `positions`, τον παρακάτω κώδικα:

```
1 <config>
2   <fields name="params">
3     <fieldset name="advanced">
4
5       <field name="templateColor" class="" type="color"
6         default="#08C" label="Χρώμα θέματος"
7         description="Βασικό χρώμα που θα έχει το
8         template" />
9
10      <field name="templateBackgroundColor" class=""
11        type="color" default="#F4F6F7" label="Χρώμα
12        Background" description="Χρώμα για το Background
13        του template" />
14
15      <field name="logoFile" class="" type="media" default=""
16        label="Λογότυπο"
17        description="Εικόνα Λογότυπου" />
18
19      <field name="sitetitle" type="text" default=""
20        label="Τίτλος"
21        description="Ο τίτλος της εφαρμογής"
22        filter="string" />
23
24      <field name="sitedescription" type="text" default=""
25        label="Περιγραφή Εφαρμογής"
26        description="Γράψε μια Περιγραφή για την
27        εφαρμογή" filter="string" />
```

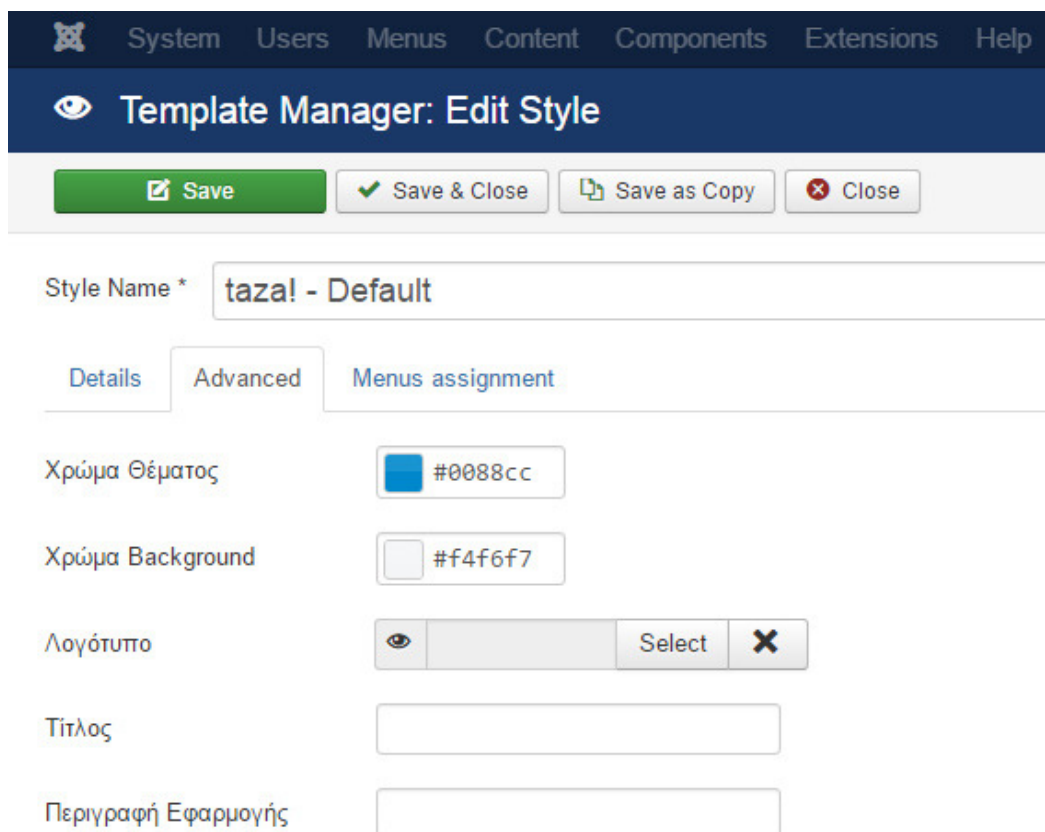


```
28     </fieldset>
29 </fields>
30 </config>
```

Αυτά είναι κάποια fields που ενδεικτικά επιλέξαμε να χρησιμοποιήσουμε για αυτό το παράδειγμα. Ανάλογα με τις προτιμήσεις του, ο καθένας μπορεί να προσθέσει πολλά ακόμα fields, δίνοντας **όνομα**, **κλάση** για πιθανή ομαδική σχεδίαση στο css, **τύπο**, προεπιλεγμένη τιμή με το default, ένα **label** για το κείμενο που θα εμφανίζεται στο tab, **περιγραφή** του field και αν είναι κείμενο το δηλώνουμε σαν string μέσω ενός **filter**, όπως φαίνεται στα δυο τελευταία.

Είναι υποχρεωτικό οι παράμετροι που θα δηλώσουμε να περιέχονται μέσα σε ένα **fields** tag, το οποίο έχει όνομα **params** και περικλείεται από ένα **config** tag. Έπειτα με το **fieldset** δημιουργούμε το νέο tab στο γραφικό περιβάλλον με το όνομα advanced. Μπορούμε να δημιουργήσουμε όσα tabs επιθυμούμε, ομαδοποιώντας έτσι τις ρυθμίσεις που ορίζουμε. Τα ονόματα, τα επιλέγουμε εμείς, και θα εμφανίζονται στο πάνω μέρος του tag, οπότε καλό είναι να είναι περιγραφικά ως προς το είδος των ρυθμίσεων.

Στο παράδειγμά μου, όρισα πέντε fields, δύο τύπου χρώματος, δύο τύπου κειμένου και ένα τύπου media. Αυτά τα fields παράγουν το ακόλουθο αποτέλεσμα στο γραφικό περιβάλλον:



Εικόνα 3.1-2 Το Advanced tab στο γραφικό περιβάλλον

Παρατηρούμε τις αλλαγές που υπάρχουν στα fields ανάλογα τον τύπο και τον εάν δηλώσαμε προεπιλογή. Πρέπει, λοιπόν, να είμαστε ιδιαίτερα προσεκτικοί κατά τον ορισμό των fields ώστε να παρουσιαστούν τα αναμενόμενα αποτελέσματα. Όπως φαίνεται από την παραπάνω εικόνα υπάρχουν άλλα δύο tabs είναι που περιέχονται στο template μου βρίσκονται εκεί από την αρχική εγκατάσταση. Το πρώτο με το όνομα Details περιέχει τις πληροφορίες του θέματος που θέσαμε στο xml αρχείο και το άλλο με το όνομα Menus Assignment περιέχει όλες τις σελίδες που υπάρχουν στην εφαρμογή ώστε να μπορούμε να τις ομαδοποιήσουμε εύκολα, εάν το επιθυμούμε.

➤ Html φάκελος

Στα θέματα που κατασκευάζω συνηθίζω να προσθέτω έναν html φάκελο για περιπτώσεις που θέλω να προσθέσω ένα html αρχείο που έφτιαξα τοπικά. Για παράδειγμα υπάρχουν εφαρμογές που όταν του δηλώνουμε ευαίσθητα στοιχεία ανοίγουν ένα νέο παράθυρο μέσω https, δηλαδή secure hypertext transfer protocol, το οποίο δεν έχει ιδιαίτερη σχεδίαση, αλλά προστατεύει με αξιοπιστία τα στοιχεία μας.

Αν ήθελα να κάνω κάτι παρόμοιο, θα κατασκεύαζα στον τοπικό μου editor την σελίδα με όλο τον κώδικα, και μετά θα ανέβαζα το αρχείο απευθείας στον server. Για περιπτώσεις σαν και αυτήν θέλω να υπάρχει ένας άδειος html φάκελος ώστε όταν χρειαστεί να είμαι έτοιμος και οργανωμένος.

Η οργάνωση αποτελεί σημαντικό παράγοντα για την τελική μορφή που θα πάρει το template μου. Αν η δομή και τα ονόματα των αρχείων δεν είναι σωστά, το Joomla θα απορρίψει το θέμα μας και κατά την διάρκεια του upload θα εμφανίσει σφάλμα.

Ολοκληρώνοντας και αυτό το κομμάτι το τελικό μου θέμα μοιάζει κάπως έτσι:

```
taza
├── css
│   ├── mystyle.css
│   └── error.css
├── images
│   └── logo.png
├── html
├── js
│   └── jquery.js
├── index.php
├── templateDetails.xml
├── template_preview.png
├── template_thumbnail.png
└── error.php
```

Υπάρχουν κάποια αρχεία τα οποία δεν επιδέχονται αλλαγής, αλλά αν κατά την διάρκεια της κατασκευής κατανοήσαμε την λογική που λειτουργεί το Joomla είμαστε σε θέση να πειραματιστούμε με όλα τα υπόλοιπα αρχεία. Γενικά πρέπει πάντα να θυμόμαστε πως το

σημαντικό δεν είναι να μην εμφανιστούν λάθη, αλλά όταν εμφανιστούν, να νιώθουμε πως είμαστε σε θέση να τα αντιμετωπίσουμε.

3.2 Εγκατάσταση template

Όταν η διαδικασία εύρεσης ή κατασκευής ολοκληρωθεί επόμενο βήμα είναι να εγκαταστήσουμε το template στο περιβάλλον του Joomla. Προχωρούμε στην εγκατάσταση ακόμα και αν δεν είμαστε σίγουροι για την λειτουργία του θέματος, καθώς εάν υπάρχει κάποιο βασικό λάθος στα αρχεία, το σύστημα κατά την διάρκεια του upload θα εμφανίσει μήνυμα που θα το υποδεικνύει, άρα θα ξέρουμε τι να διορθώσουμε. Αν υπάρχει σχεδιαστικό λάθος, τότε κανένα πρόβλημα, αφού μπορούμε να επεξεργαστούμε τον κώδικά μας γραφικά στον editor του Joomla.

Η εγκατάσταση πραγματοποιείται από το οριζόντιο μενού στο πάνω μέρος και τις επιλογές Extensions / Extensions Manager. Εκεί υπάρχει το κουμπί upload όπου μπορούμε να ανεβάσουμε το zip αρχείο. Υπάρχουν και άλλα δύο tab όπου παρέχουν την δυνατότητα εγκατάστασης απευθείας μέσω URL, για αρχεία στο διαδίκτυο και μέσω directory για αρχεία που βρίσκονται στον Server.

Αφού ανέβει το θέμα μας, πρέπει να το ενεργοποιήσουμε, ή έστω να ελέγξουμε αν ενεργοποιήθηκε αυτόματα. Την ενεργοποίηση την πραγματοποιούμε ξανά από το οριζόντιο μενού και συγκεκριμένα το μονοπάτι Extensions / Template Manager. Εκεί μας εμφανίζει μια λίστα με τα διαθέσιμα θέματα που αφορούν είτε για το σύστημα είτε για το γραφικό περιβάλλον. Εάν στην στήλη **location** υπάρχει η περιγραφή “Administrator” αυτό το θέμα αφορά το γραφικό περιβάλλον του συστήματος και δεν επηρεάζει την εφαρμογή μας. Αντίθετα όταν υπάρχει η περιγραφή “Site” τότε το θέμα αφορά αποκλειστικά την εφαρμογή. Για να δούμε ποια θέματα είναι ενεργοποιημένα αρκεί να παρατηρήσουμε την στήλη **default** όπου υπάρχουν κάθετα αστέρια. Όταν το αστέρι του θέματος είναι χρυσό, αυτό το θέμα είναι ενεργοποιημένο με αποτέλεσμα όλα τα υπόλοιπα θέματα, τα οποία έχουν το ίδιο location να αποκτούν αυτόματα γκρι αστέρι.

Για κάποιον καινούριο στο Joomla, λογικά αυτή είναι η πρώτη στιγμή που συνειδητοποιεί την ευχρηστία του συγκεκριμένου CMS. Μπορούμε να έχουμε αμέτρητα templates, αλλά επαγγελματικά με πολλές χιλιάδες γραμμές κώδικα, αλλά με μηδαμινό σχεδιασμό, όλα όμως θα βρίσκονται τακτοποιημένα σε αυτή την λίστα, και η ολική μετατροπή της εφαρμογής μας διαρκεί ελάχιστα δευτερόλεπτα. Το μόνο που απαιτείται είναι το πάτημα ενός κουμπιού.

Βρίσκουμε το θέμα που ανεβάσαμε στην λίστα και ελέγχουμε το χρώμα του αστεριού στην διπλανή στήλη. Εάν είναι χρυσό το αφήνουμε ως έχει, καθώς ενεργοποιήθηκε αυτόματα, σε αντίθετη περίπτωση επιλέγουμε το γκρι αστέρι ώστε να γίνει η αλλαγή του ενεργοποιημένου θέματος. Για να επαληθεύσουμε την επιτυχία της αλλαγής κοιτάμε την αρχική σελίδα της εφαρμογής μέσω ενός browser. Εάν έχουμε θέσει την εφαρμογή σε offline mode τότε θα εμφανιστεί απλώς μια φόρμα όπου θα χρειαστεί να βάλουμε τα στοιχεία μας ως administrator. Εισάγουμε τα στοιχεία μας και η εφαρμογή εμφανίζεται, αλλά θα είναι ορατή μόνο σε εμάς.

3.3 Επεξεργασία template

Τελευταίο αλλά εξίσου σημαντικό κομμάτι είναι η επεξεργασία των αρχείων του template από την στιγμή που βρίσκεται ήδη στο σύστημα του CMS. Όταν κατασκευάζουμε σελίδα για κάποιον πελάτη τότε είναι απαραίτητο να είμαστε απόλυτα εξοικειωμένοι με τον editor του Joomla καθώς σε αντίθετη περίπτωση θα εκτεθούμε όταν μας ζητηθούν αλλαγές. Δεν αρκεί να ξέρουμε πως αλλάζει κάτι, είναι υποχρέωση μας να ξέρουμε και που βρίσκεται το σημείο της αλλαγής, ώστε να την πραγματοποιήσουμε άμεσα. Το Joomla, όπως πάντα, προβλέπει τις ανάγκες του δημιουργού και μας βοηθά στο να έχουμε τον απόλυτο έλεγχο προσθέτοντας ρυθμίσεις που όμως προϋποθέτουν την μελέτη και εξοικείωση μας με το σύστημα.

Σε μεγάλα project η ευθύνη θα είναι σαφώς μεγαλύτερη, καθώς λογικά θα υπάρχει υπεύθυνος για το design της εφαρμογής όπου θα πρέπει να του παρουσιάσουμε ακριβώς αυτό που μας υπέδειξε και να είμαστε έτοιμοι για πολλές ώρες αλλαγών. Ακόμα όμως και σε περιπτώσεις όπου δουλεύουμε μόνοι, ακόμα και στην απίθανη περίπτωση ο πελάτης να μας παρέχει απόλυτη ελευθερία και εμπιστοσύνη στον σχεδιασμό, είναι σπάνιο να μην ζητηθεί κάποια αλλαγή, τουλάχιστον σε ένα χρώμα, έστω και δοκιμαστικά. Το Joomla μας προσφέρει δύο τρόπους επεξεργασίας του θέματος.

3.3.1 Αλλαγές στην κατηγορία Styles

Όταν μπούμε στο Extension Manager στα αριστερά της οθόνης μας εμφανίζεται ένα νέο μενού με δύο επιλογές. Από προεπιλογή βρισκόμαστε ήδη στην πρώτη κατηγορία η οποία λέγεται Styles. Νωρίτερα από εδώ ενεργοποιήσαμε το θέμα μας, πράγμα που σημαίνει πως ήδη πραγματοποιήσαμε την πρώτη μας αλλαγή. Η λογική σε αυτήν την κατηγορία είναι οι αλλαγές να γίνονται γραφικά, χωρίς καμία αλλαγή στον κώδικα. Αυτό σημαίνει πως ο κώδικας του θέματος πρέπει να είναι κατασκευασμένος με PHP όπου σε κάποια σημεία θα καλεί με την εντολή echo μεταβλητές οι οποίες θα παίρνουν τιμές γραφικά.

Οι ρυθμίσεις στο Styles είναι ξεχωριστές για κάθε θέμα, αφού πατήσουμε το όνομα του από την λίστα, και είναι ιδανικές για αρχάριους προγραμματιστές που δεν τολμούν αλλαγές στο κώδικα καθώς και για πελάτες που θέλουν να έχουν τις βασικές δυνατότητες επεξεργασίας της εφαρμογής τους. Εδώ αξίζει να υπενθυμίσουμε όμως πως ο administrator ελέγχει τα δικαιώματα των χρηστών. Αν ο πελάτης δεν είναι εξοικειωμένος με το περιβάλλον, τότε είναι καλύτερα να του στερείται η πρόσβαση ακόμα και σε βασικές ρυθμίσεις.

Στο υποκεφάλαιο 3.1.3, είδαμε πως να προσθέσουμε fields τα οποία δημιουργούν γραφικές ρυθμίσεις. Μπορούμε εκεί με δημιουργία νέου field να προσθέσουμε ρυθμίσεις που μας ζητήθηκαν ή απλώς ρυθμίσεις που χρειάζεται να πειραματιστούμε και θέλουμε γρήγορη πρόσβαση. Σε περίπτωση που χρησιμοποιούμε ένα έτοιμο και προσεγμένο θέμα, στην κατηγορία Styles θα βρούμε έτοιμα πολλά tabs που μας κάνουν την ζωή εύκολη. Για παράδειγμα, οι συχνότερες δυνατότητες που μας δίνονται είναι ανέβασμα λογότυπου, metadata για την εφαρμογή, κάποια ρύθμιση για το layout των σελίδων,

παλέτα χρωμάτων και σε επαγγελματικά θέματα θα υπάρχουν σίγουρα ξεχωριστά tabs για styling, SEO και για την διαρρύθμιση του μενού.

3.3.2 Αλλαγές στην κατηγορία Templates

Κάτω από το Styles, η δεύτερη επιλογή λέγεται Templates και εκεί μας δίνεται η δυνατότητα να πραγματοποιήσουμε αλλαγές όπως θα τις κάναμε απευθείας στον Server. Βρίσκοντας το θέμα που θέλουμε να επεξεργαστούμε και πατώντας στο όνομα του ανοίγονται μπροστά μας όλοι οι φάκελοι που το αποτελούν μαζί με τον περίφημο editor του Joomla που τον αναφέρω από το πρώτο κεφάλαιο. Εδώ μπορούμε να κάνουμε δραστικές αλλαγές και να τροποποιήσουμε την εμφάνιση ακόμα και ενός έτοιμου θέματος πάνω στις απαιτήσεις μας.

Ένα προσεγμένο θέμα, όπως το βασικό που δημιουργήσαμε νωρίτερα, θα έχει τακτοποιημένα τα αρχεία σε φακέλους και τα ονόματα πάντα θα επεξηγούν το περιεχόμενο. Το τελευταίο είναι απαραίτητο για περιπτώσεις όπου μας ενοχλεί κάτι μικρό, όπως για παράδειγμα ένα label που υποχρεωτικά τοποθετεί κείμενο στο slider της εφαρμογής μας. Αν τα αρχεία δεν έχουν σωστές ονομασίες θα είναι τρομερά δύσκολο να βρούμε τον κώδικα του slider και να κρύψουμε το κείμενο, με μοναδική ελπίδα να ανατρέξουμε σε βοήθεια από τα Inspect του browser και να τα εντοπίσουμε.

Για κάθε αρχείο που θα επιλέξουμε, ο editor παρουσιάζει τα περιεχόμενα του στο δεξί μέρος της σελίδας και δεν ολοκληρώνει ποτέ τις αλλαγές εάν δεν αποθηκεύσουμε και βγούμε από την επεξεργασία του αρχείου. Προφανώς οι αλλαγές εδώ γίνονται με μεγάλη προσοχή και υπευθυνότητα καθώς μπορούν να αποβούν μοιραίες.

Οι περισσότεροι templates developers συμβουλεύουν του χρήστες των θεμάτων τους να μην πραγματοποιούν αλλαγές μέσω του editor. Αυτό συμβαίνει για δύο λόγους. Πρώτον γιατί δημιουργούν προβλήματα στην εφαρμογή και μετά επικοινωνούν με τον developer του θέματος, ζητώντας βοήθεια, κάτι που προφανώς είναι παράλογο, και δεύτερον γιατί πολλοί χρήστες που περιεργάζονται τον κώδικα έχουν την τάση να σβήνουν τα στοιχεία του δημιουργού από τα σχόλια στον κώδικα. Τα θέματα που μας παρέχονται δωρεάν, περιορίζουν τα έσοδα τους μόνο από την διαφήμιση και το ελάχιστο δείγμα ευγνωμοσύνης από εμάς, είναι να αφήνουμε τις πληροφορίες του δημιουργού για όποιον κοιτάξει τον κώδικα των σελίδων μας.

3.3.3 Child themes

Συζητώντας την επεξεργασία των templates και μετά από την γνωριμία με τον editor, είναι απαραίτητο να αναφέρουμε τα child themes. Όπως γνωρίζουμε, το Joomla είναι μια πλατφόρμα που εξελίσσεται διαρκώς. Από τις αρχές του 2017 η έκδοση 3.6 είναι διαθέσιμη, με ουσιαστικές αλλαγές κυρίως στην ασφάλεια και την προστασία των χρηστών, αλλά και με μικρές εξελίξεις στο μενού. Αυτό σημαίνει πως οι χρήστες πρέπει να κάνουν update το σύστημα τους. Και σε λίγες βδομάδες θα χρειαστεί ξανά update για την έκδοση 3.6.1. Και αυτό δεν θα σταματήσει ποτέ όσο η κοινότητα εξελίσσεται.

Έτσι και με τα templates, όταν γίνεται επαγγελματική δουλειά και τα downloads αγγίζουν πενταψήφιους αριθμούς, ο κατασκευαστής δεν μπορεί να επαναπαυτεί, αλλά πρέπει να εξελίξει και να αναβαθμίσει το προϊόν σε επικοινωνία με τους χρήστες. Το πρόβλημα όμως είναι πως το σύστημα Joomla είναι σε θέση να κάνει αναβαθμίσεις χωρίς να επηρεάζει τις ρυθμίσεις του χρήστη. Στα θέματα αυτό είναι αδύνατο, εκτός και αν ο χρήστης προσθέσει μόνος του στους φακέλους τα νέα αλλαγμένα αρχεία, κάτι που είναι σχεδόν αδύνατο. Άρα πρέπει να είμαστε προσεχτικοί με τις αλλαγές που πραγματοποιούμε στον editor όχι μόνο για τυχόν λάθη, αλλά γιατί αν πραγματοποιήσουμε update του θέματος είναι πιθανό να σβηστούν και η πρόοδος μας να χαθεί.

Το Wordpress για να εγγυηθεί την ασφάλεια των αρχείων, επιτρέπει την δημιουργία child themes. Μέσα από μια σχετικά απλή διαδικασία ο χρήστης μπορεί να δημιουργήσει ένα νέο πανομοιότυπο θέμα, κάτι σαν κλώνο του αρχικού θέματος, με τις αλλαγές που έχει αυτός πραγματοποιήσει, το οποίο σε αυτή την περίπτωση αποκαλείται parent theme. Έτσι όταν προκύψουν updates ο χρήστης τα πραγματοποιεί στο parent theme και κρατά ενεργοποιημένο το child theme στην εφαρμογή. Ελέγχει τις αλλαγές, που προέκυψαν με την αναβάθμιση και έπειτα ενεργεί, σίγουρος πως η εφαρμογή δεν τροποποιήθηκε χωρίς την έγκρισή του.

Τα child themes είναι ένα χρήσιμο εργαλείο που βέβαια προϋποθέτει αλληλεπίδραση με τον Server, αλλά σε πολύ βασικό στάδιο. Στο Joomla όπως και στα περισσότερα CMS, προς το παρόν υπάρχουν παρόμοιες λειτουργίες, όπως τα overrides τα οποία προστατεύουν αρχεία από αλλαγές. Ενώ είναι χρήσιμα, καθώς αποκλείεται να αλλάξει το αρχείο σου, αχρηστεύουν τις μεγάλες αναβαθμίσεις. Για παράδειγμα, πάντα θα προστατεύουμε το αρχείο mystyle.css από αλλαγές, αλλά είναι δύσκολο η αναβάθμιση να μην περιέχει αρχεία εκεί, άρα δεν θα λειτουργήσει σωστά.

Προς το παρόν, τουλάχιστον για αυτούς που είναι ικανοποιημένοι με το θέμα τους και δεν έχουν πρόβλημα ασφαλείας, η καλύτερη συμβουλή είναι να ελέγχουν τι προσφέρουν τα νέα updates από την περιγραφή τους, και αν δεν είναι κάτι απαραίτητο να τα αποφεύγουν.

4. Δημιουργία διαδικτυακής εφαρμογής

Ολοκληρώνοντας το κεφάλαιο με τον ρόλο των templates, μπορούμε πλέον να ξεκινήσουμε την ουσιαστική κατασκευή της εφαρμογής μας. Ανοίγεται μπροστά μας ένα τελείως διαφορετικό κεφάλαιο που ο ρόλος του προγραμματιστή μπαίνει σε δεύτερη μοίρα και περιορίζεται στην επεξεργασία του CSS αρχείου και στην περιγραφή νέων κλάσεων για τα αντικείμενα που θα προσθέσουμε στις σελίδες μας. Την σκυτάλη τώρα παίρνουν οι αρθογράφοι και οι συντάκτες. Εδώ κρύβεται η πραγματική δύναμη των CMS, στην ευκολία που μας παρέχει να μετατρέψουμε μια κενή εφαρμογή σε πλούσια και ολοκληρωμένη μέσα σε λίγες ώρες. Σε αυτό το κεφάλαιο θα εξερευνήσουμε τις δημιουργικές ιδιότητες του Joomla, παρατηρώντας κυρίως, την διαδικασία κατασκευής της εφαρμογής caferpatras.gr.

4.1 Ο κύκλος των extensions

Αναφέραμε και στην θεωρία πόσο χρήσιμα είναι τα extensions στην ολοκλήρωση μιας διαδικτυακής εφαρμογής. Κατά την προσωπική μου άποψη είναι το σημαντικότερο εργαλείο των CMS. Ειδικότερα σε μια μεγάλη και ενεργή κοινότητα όπως αυτή του Joomla, οι δυνατότητες για αναβάθμιση της εφαρμογής μας αυξάνονται λογαριθμικά. Ένας ικανός προγραμματιστής θα μπορούσε φυσικά με την βοήθεια από εγχειρίδια και παραδείγματα στο διαδίκτυο να κατασκευάσει μόνος του όλα τα πρόσθετα που χρειάζεται η εφαρμογή, αλλά η χρήση των extensions τον γλυτώνει από περιττό χρόνο και πειράματα. Είναι γνωστό άλλωστε πως ο χρόνος, και στον τομέα της τεχνολογίας, είναι χρήμα.

4.1.1 Αναζήτηση

Πριν ξεκινήσουμε λοιπόν την δημιουργία σελίδων, είναι φρόνιμο να καταλήξουμε στο τι χρειάζεται η εφαρμογή μας, το οποίο εμείς δυσκολευόμαστε να κατασκευάσουμε. Για παράδειγμα, οι περισσότερες μοντέρνες εφαρμογές χρησιμοποιούν περισσότερες εικόνες και λιγότερα κείμενα. Αυτό αυτομάτως δημιουργεί σε πολλούς την επιθυμία για ένα slider με εικόνες στην εφαρμογή τους. Κάνοντας ανάλογους συλλογισμούς, καταλήγω στις βασικές ανάγκες της εφαρμογής και ξεκινώ μια έρευνα αγοράς.

Για κάθε CMS ο τρόπος αναζήτησης των extensions διαφέρει. Στην περίπτωση του Joomla όλα τα διαθέσιμα extensions μαζί με περιγραφές, demo, documentations και κριτικές πελατών, βρίσκονται οργανωμένα στον ιστότοπο extensions.joomla.org. Από τις αρχές του 2017 ο συνολικός αριθμός extensions για το Joomla αγγίζει τα 8,000. Γίνεται εύκολα κατανοητό πως σε μια κοινότητα σαν και αυτή, είναι πολύ δύσκολο να μην βρούμε αυτό που ψάχνουμε.

Για να μείνουμε στο παράδειγμα με το image slider, η αναζήτηση για extensions στο Joomla μπορεί να πραγματοποιηθεί με δυο τρόπους. Ο πρώτος είναι αναζήτηση βάση κατηγορίας, που είναι ο γενικός τρόπος. Τον χρησιμοποιούμε όταν δεν είμαστε σίγουροι για το τι χρειαζόμαστε και θέλουμε να δούμε τις επιλογές που έχουμε. Αναγκαστικά θα επιλέξουμε την κατηγορία photos & images και θα αναζητήσουμε στην γενική λίστα αυτό που ψάχνουμε.

Ο άλλος τρόπος αναζήτησης, και ο περισσότερο ορθολογικός στην δική μας περίπτωση, είναι να συμπληρώσουμε την φόρμα στο πάνω μέρος. Εκεί μας δίνεται η δυνατότητα να δώσουμε λέξεις κλειδιά, και άλλες δυνατότητες φιλτραρίσματος όπως κατηγορία, τύπος (plug-in / module), συμβατότητα (με εκδόσεις του Joomla), τιμή, και βαθμολογία. Σε αυτήν την περίπτωση θα γράψουμε τις λέξεις image slider και θα επιλέξουμε συμβατότητα με Joomla 3 και πάνω. Το αν θα διαθέσουμε χρήματα για ένα extension είναι στην κρίση του καθενός, αλλά προσωπικά δεν πιστεύω πως αξίζει να επενδύσουμε χρήματα σε κάποιο slider. Με μια επίσκεψη στην σελίδα του δημιουργού και μια καλή κριτική, πολλές φορές του προσφέρουμε έμμεσα ανταλλάγματα για τον κόπο του.

Σε κάθε περίπτωση, το σημαντικό είναι να από την λίστα που θα μας εμφανίσει, να δίνουμε βαρύτητα στα score και reviews. Για να κάνουμε την αναζήτηση πιο εύκολη, μπορούμε να ταξινομήσουμε τα αποτελέσματα με βάση τα ratings. Το score έχει για άριστα το 100 ενώ τα reviews είναι κριτικές χρηστών που σχολιάζουν το extension. Όταν το score ξεπερνά το 90 και δούμε τουλάχιστον 50 reviews σημαίνει πως το extension είναι πετυχημένο άρα και λειτουργικό. Προφανώς όμως, η βαθμολογία που δίνεται εξαρτάται και από τις απαιτήσεις του κάθε χρήστη.

Είναι απαραίτητο λοιπόν, όταν δούμε μια εφαρμογή που μας ικανοποιεί να πατήσουμε πάνω της και να διαβάσουμε την περιγραφή από τον δημιουργό της και τις κριτικές των χρηστών. Το Joomla για να κάνει τις κριτικές πιο εμπεριστατωμένες επιτρέπει στον χρήστη να βαθμολογήσει την εφαρμογή με βάση πέντε παράγοντες. Το support του δημιουργού, τα documentation που προσφέρει, την ευκολία χρήσης, την λειτουργικότητα και την συνολική βαθμολογία της εφαρμογής. Με αυτό τον τρόπο, ξέρουμε ακριβώς που υπερτερεί και που υστερεί ένα extension πριν μπούμε στην διαδικασία να το εγκαταστήσουμε στο σύστημα.

Όπως αναφέραμε και νωρίτερα, τεράστιας σημασίας είναι και τα demo που προσφέρονται, καθώς βλέπεις το extension σε λειτουργία και αποφασίζεις ευκολότερα αν βρήκες αυτό που ψάχνεις. Αξίζει όμως να αναφέρουμε πως χρειάζεται ιδιαίτερη προσοχή καθώς τα περισσότερα δωρεάν extensions παρουσιάζουν στο demo τους λειτουργίες οι οποίες είναι επί πληρωμή. Αυτό έχει αποτέλεσμα να ξοδεύουμε άσκοπα χρόνο και ενέργεια.

Μια τελευταία συμβουλή για τον τομέα αναζήτησης των extension, είναι πως ακόμα και όταν πιστεύεις πως δεν χρειάζεσαι κάτι, ένα πέραςμα από συγκεκριμένες κατηγορίες μπορεί να φανεί πολύ χρήσιμο. Κατηγορίες όπως access & security και administration παρέχουν προτάσεις που δεν φαίνονται στο τελικό αποτέλεσμα, αλλά βοηθούν έμμεσα με το να νιώθεις ασφαλή την εφαρμογή σου και για να προσαρμόσεις το περιβάλλον του συστήματος πάνω στις δικές σου ανάγκες.

4.1.2 Εγκατάσταση και ενεργοποίηση

Στο πρώτο κεφάλαιο κάναμε μια μικρή αναφορά για τους τύπους των extensions και αναφέραμε πως οι δύο κυριότεροι είναι το module και τα plug-ins. Συνήθως πριν το κατεβάσουμε δεν μας απασχολεί τι τύπο θα έχει το extension εκτός και αν ενδιαφερόμαστε υπερβολικά για την ταχύτητα απόκρισης της σελίδας, άρα θα αποφύγουμε τα περίπλοκα modules. Κατά την διάρκεια της εγκατάστασης όμως πρέπει να γνωρίζουμε τον τύπο για να ολοκληρώσουμε την διαδικασία.

Ας πάρουμε όμως τα πράγματα από την αρχή. Βρήκαμε νωρίτερα το extension που αναζητούσαμε και θέλουμε να προχωρήσουμε με την εγκατάσταση. Πρώτο βήμα είναι να κατεβάσουμε το zip αρχείο του extension στον υπολογιστή μας. Όπως είπαμε νωρίτερα, το upload μπορεί να γίνει και μέσω URL, αλλά δεν το συνηθίζουμε χωρίς να περάσει το αρχείο από ένα antivirus πρόγραμμα. Έπειτα πηγαίνουμε στην επιλογή Extensions / Extension Manager του οριζόντιου μενού, κάνουμε upload το αρχείο και περιμένουμε το πράσινο μήνυμα πως η διαδικασία ολοκληρώθηκε με επιτυχία.

Στην περίπτωση που το extension είναι τύπου plug-in επόμενο βήμα είναι να πάμε στον Plugin Manager που βρίσκεται μία θέση κάτω στο από το Extensions Manager. Εκεί εμφανίζεται μια λίστα με όλα τα plug-ins που περιέχει η σελίδα μας. Αναζητούμε το καινούριο που ανεβάσαμε και ελέγχουμε από την στήλη Status εάν είναι ενεργοποιημένο, ώστε γρήγορα να το απενεργοποιήσουμε μέχρι να τελειώσουμε την παραμετροποίηση. Μπαίνοντας μέσα βλέπουμε τις ρυθμίσεις που μας παρέχει ο δημιουργός και εκτελούμε ανάλογα με τις οδηγίες των documentation που υπάρχουν στην σελίδα του. Για παράδειγμα κάποιο slider θα έχει την επιλογή σε εκείνο το σημείο να φορτώσουμε τις εικόνες μας, ενώ άλλο μπορεί να απαιτεί να φτιάξουμε ένα φάκελο με αυτές στο Media Manager. Όταν έχουμε τελειώσει με τις ρυθμίσεις, ενεργοποιούμε το plug-in πατώντας το activate και πλέον είμαστε θέση να το καλέσουμε στην εφαρμογή. Για το πώς θα γίνει αυτό, συμβουλευόμαστε για άλλη μια φορά τα documentation καθώς είναι πάλι ανάλογα τον δημιουργό. Τα περισσότερα plug-ins απαιτούν να τα καλέσουμε σε σημείο του κώδικα της σελίδας με μια συγκεκριμένη εντολή, αλλά υπάρχουν περιπτώσεις όπου ορίζουμε το που και ποτέ θα εμφανιστούν από τις ρυθμίσεις.

Η δεύτερη περίπτωση είναι το extension να είναι τύπου module. Όπως και νωρίτερα στο υπομενού του Extensions βρίσκουμε το Module Manager και ανοίγουμε την λίστα με τα modules. Παρατηρούμε πως σε σχέση με την λίστα των plug-ins υπάρχει μια βασική διαφορά. Υπάρχει μια στήλη που λέγεται position και ορίζει την θέση που θα πάρει το module μέσα στην εφαρμογή μας. Τις θέσεις όπως είδαμε κατά την κατασκευή του αρχείου templateDetails.xml, τις ορίζει ο δημιουργός του θέματος με τα positions. Σε αντίθεση με πριν, τα modules δεν τα καλούμε στην εφαρμογή, αλλά τα τοποθετούμε σε κάποιο από τα δηλωμένα σημεία. Για τις επιπλέον ρυθμίσεις δεν μπορούμε να κάνουμε κάποια περιγραφή, γιατί για άλλη μια φορά ακολουθούμε πιστά τις οδηγίες των documentation. Υπενθυμίζουμε ξανά όμως πως τα modules είναι πιο δυναμικά κομμάτια κώδικα και κατά πάσα πιθανότητα θα διαθέτουν αρκετά tabs με διαθέσιμες παραμετροποιήσεις. Τελειώνοντας για να δούμε τα αποτελέσματα κάνουμε active το extension και ελέγχουμε την σελίδα μας.

4.1.3 Παρουσίαση των extensions του cafepatras.gr

Εδώ παρουσιάζονται όλα τα extensions που χρησιμοποιήθηκαν για να ολοκληρωθεί και να συντηρηθεί το cafepatras.gr. Τα προτιμήσαμε έπειτα από αναλυτική έρευνα αγοράς και εμπειρίες του παρελθόντος. Δεν αναφέρονται πουθενά τιμές γιατί όλα τα παρακάτω extensions διατίθενται δωρεάν στο extensions.joomla.org. Στο πρώτο κεφάλαιο της εργασίας αναφερθήκαμε και στα components τα οποία μοιάζουν πολύ με τα extensions. Το τελευταίο αντικείμενο της λίστας είναι ένα component και θα μας βοηθήσει να ξεδιαλύνουμε τις διαφορές τους στην πράξη, γιατί στην θεωρία είναι περίπλοκο.

➤ ExtraVote

Το πρώτο είναι ένα μικρό plug-in που χρησιμεύει για τις αξιολογήσεις των καφετεριών της εφαρμογής. Είναι εξαιρετικά απλό και αποτελείται από μόλις τρία αρχεία. Καταφέρνει όμως αισθητικά να είναι το πιο κομψό στο είδος του, να επιτρέπει να το χρησιμοποιήσεις παραπάνω από μια φορές στην σελίδα και να κρατάει την IP χρηστών που έχουν ψηφίσει, ώστε να μην επαναλαμβάνονται ψήφοι. Όταν το ExtraVote είναι ενεργοποιημένο, το καλούμε στην σελίδα μας με την εντολή {extravote #}. Όπου η δίσωση, συμβολίζει το ID του κάθε rating. Σύμφωνα με τον δημιουργό, ξεκινώντας από το {extravote 1} και ανεβαίνοντας σταδιακά, σε μια σελίδα μπορούν να λειτουργήσουν ταυτόχρονα και χωρίς προβλήματα έως 80 βαθμολογίες. Είναι παρόλα αυτά, ένα από τα αναλώσιμα plug-ins καθώς στο μέλλον είναι πιθανό να αντικατασταθεί στην εφαρμογή με κάποιο καινούριο, που θα προσφέρει ακόμα περισσότερες δυνατότητες.

➤ SigPlus Image Gallery

Το Sigplus είναι ένα πολύ δυνατό και αξιόπιστο extension το οποίο το εμπιστεύομαι διαχρονικά όταν χρειάζομαι κάποιο Image Gallery. Το Score του από χρήστες στην λίστα είναι 100/100 και τα documentation που παρέχονται είναι αναλυτικά εξηγώντας ακόμα και τις λεπτομέρειες που απαιτούνται για τον χειρισμό του. Το Sigplus γενικά, κατασκευάστηκε ως module για Joomla, αλλά για την διευκόλυνση των χρηστών και με περιορισμό κάποιων δυνατοτήτων του, μπορεί να λειτουργήσει και σαν plug-in. Προστίθεται δηλαδή, και στις δύο λίστες επιτρέποντας στον administrator να το εκμεταλλευτεί ανάλογα τις ανάγκες του. Ως module ζητά το path από κάποιον φάκελο εικόνων που έχουμε δημιουργήσει στο Media Manager και μας δίνει την δυνατότητα προβολής των εικόνων σε συγκεκριμένο σημείο της εφαρμογής, παρέχοντας αμέτρητες ρυθμίσεις για σχεδιασμό του παραγόμενου αποτελέσματος. Στην περίπτωση του cafepatras.gr όμως, αυτό που χρειαζόμαστε είναι η κάθε καφετέρια να διαθέτει το δικό της Image Gallery, στην δική της σελίδα. Αυτό σημαίνει πως δεν ενεργοποιήσαμε το module, αλλά το plug-in. Για να λειτουργήσει όμως, θέλει όπως έχουμε ξαναδεί, μια συγκεκριμένη εντολή που θα τοποθετούμε στο σημείο της σελίδας που θέλουμε να το καλέσουμε. Αυτή η εντολή είναι {gallery}folder{/gallery}, με το folder να ορίζεται ανάλογα με το όνομα του φακέλου που διαθέτει τις εικόνες. Για παράδειγμα, στο cafepatras.gr θέλουμε να φτιάξουμε την σελίδα για μια καφετέρια στην Πάτρα που λέγεται Bodegas. Καταρχάς πηγαίνω στον Media Manager και δημιουργώ ένα φάκελο

που τον ονομάζω cafe και θα περιέχει όλες τις φωτογραφίες που αφορούν καφετέριες, Μέσα εκεί δημιουργώ έναν νέο φάκελο που τον ονομάζω bodegas. Κάνω σε εκείνο το σημείο upload τις φωτογραφίες που έχω συλλέξει για το bodegas και κλείνω τον φάκελο. Κατά την δημιουργία της σελίδας για το bodegas (την δημιουργία σελίδων την αναλύουμε αργότερα στο κεφάλαιο) στο σημείο όπου θέλω να προσθέσω το Image Gallery της καφετέριας, γράφω την εντολή `{gallery}cafe/bodegas{/gallery}`. Αν τα ονόματα δεν περιέχουν λάθη και οι φάκελοι βρίσκονται στην θέση τους, τότε στην εφαρμογή μου θα εμφανιστεί το Gallery όπως το ρύθμισα από τον Plugin Manager. Για κάθε καφετέρια θα χρειαστεί να επαναλάβω ακριβώς την ίδια διαδικασία.

➤ Disqus

Το `discus` δεν χρειάζεται ιδιαίτερες συστάσεις, καθώς είναι η πιο διαδεδομένη πλατφόρμα για την διαχείριση σχολιασμού των χρηστών. Είναι μια κοινότητα όπου οι χρήστες μπορούν να αλληλεπιδρούν μεταξύ τους και να παρακολουθούν σχόλια από όλες τις διαδικτυακές εφαρμογές που το έχουν εγκατεστημένο. Με αποτέλεσμα όταν κάποιος χρήστης του `discus` σχολιάσει στην εφαρμογή μας, αυτομάτως κάνει διαφήμιση σε όποιον παρακολουθεί τα σχόλια του. Ο σχεδιασμός του είναι μοντέρνος και λιτός με αποτέλεσμα να ταιριάζει αρμονικά με την εφαρμογή μας, κάτι που δεν συναντά κανείς εύκολα στις υπόλοιπες πλατφόρμες σχολιασμού. Το δυνατό του στοιχείο όμως είναι πως δίνει στον administrator τον απόλυτο έλεγχο των μηνυμάτων καθώς και των χρηστών που επιτρέπεται να σχολιάσουν. Για παράδειγμα, στο `cafepatras.gr` ρύθμισα το `discus` ώστε να επιτρέπεται ο σχολιασμός μέσω προφίλ του `discus`, του `twitter`, του `facebook`, του `google`, αλλά ακόμα και απλών επισκεπτών ως `guest`. Ο έλεγχος παρέχεται από την πλατφόρμα `discus` σε δική τους σελίδα, όπου ως administrator έχεις το δικαίωμα να επεξεργάζεσαι τα σχόλια σύμφωνα με τους δικούς σου κανόνες. Για να γίνουν όλα αυτά όμως πρέπει πρώτα να κατεβάσουμε και να ενεργοποιήσουμε το `plug-in` του `discus` για Joomla. Τα αρχεία που θα κατέβουν είναι λίγα, αλλά αυτό γιατί όταν τα ενεργοποιήσουμε, καλούν κώδικα από τον Server του `discus` για την ομαλή τους λειτουργία. Αυτό μας εμποδίζει από το να επεξεργαστούμε την εμφάνιση του `plug-in` μέσω κώδικα, και μας περιορίζει αποκλειστικά στις ρυθμίσεις του Plugin Manager. Σε αντίθεση με τα προηγούμενα, εδώ το `discus` δεν το καλούμε, αλλά ορίζουμε γραφικά σε ποιες σελίδες ή σε ποιες κατηγορίες σελίδων θα εμφανίζεται (περισσότερα για τις κατηγορίες υπάρχουν αργότερα στο κεφάλαιο).

➤ POWr Survey

Το `POWr Survey` δεν είχε πολλές κριτικές από χρήστες, παρόλα αυτά βλέποντας το demo του καταλάβαμε ότι ανταποκρίνεται στην ανάγκες της εφαρμογής. Η βασική του λειτουργία είναι να δημιουργεί ερωτηματολόγια σε σελίδες ώστε να βοηθά σε έρευνες. Στο `cafepatras.gr` με τροποποιημένες ερωτήσεις και `label`, χρησιμοποιείται ώστε να βοηθάει χρήστες της εφαρμογής να στέλνουν καφετέριες οι οποίες δεν περιέχονται στην εφαρμογή. Το extension δημιουργεί μια φόρμα όπου οι χρήστες, επιλέγοντας την ιδιότητα τους ως προς την καφετέρια (ιδιοκτήτης, πελάτης, περαστικός) και δίνοντας τα δικά τους βασικά στοιχεία μαζί με της καφετέριας, στέλνουν ένα μήνυμα απευθείας στο

e-mail της εφαρμογής. Όπως και με το Sigplus νωρίτερα, έτσι και εδώ, το POWr Survey είναι ένα module που στην βασική και δωρεάν έκδοση του, το χρησιμοποιούμε σαν plug-in για ευκολία. Η κλήση του στην σελίδα ως module γίνεται από τον Module Manager όπου δηλώνουμε απλώς το position και τις σελίδες που επιθυμούμε να εμφανίζεται και το ενεργοποιούμε. Ως plug-in η κλήση του στην σελίδα γίνεται με την εντολή [powr-survey] στο σημείου όπου θέλουμε να εμφανιστεί η φόρμα.

➤ Google Analytics

Τα Google Analytics είναι με μεγάλη διαφορά η καλύτερη πλατφόρμα tracking για διαδικτυακές εφαρμογές. Παρακολουθεί όλες τις κινήσεις που γίνονται στην εφαρμογή, παρουσιάζει διαγράμματα με τις δημοφιλέστερες σελίδες, ομαδοποιεί τους επισκέπτες ανάλογα τις συνήθειες τους, και γενικότερα μας αναφέρει τα πάντα που συμβαίνουν. Όταν μια νέα εφαρμογή έχει μεγάλη επισκεψιμότητα, χρησιμοποιεί τα στοιχεία από τα google analytics της, ώστε να το αποδείξει σε διαφημιστές. Αυτοί με την σειρά τους μπορούν, χρησιμοποιώντας το ίδιο μέσο, να δουν πόσοι επισκέπτες έρχονται στην σελίδα τους μέσω της εφαρμογής που τους διαφημίζει. Για μια νέα εφαρμογή είναι ότι πιο κοντινό σε επαγγελματικό εργαλείο καταγραφής κίνησης και παρέχεται δωρεάν. Για την ομαλή εγκατάσταση και λειτουργία του στην Joomla εφαρμογή μας, υπάρχει plug-in που μας βοηθά με την ρύθμιση του. Η παρακολούθηση του όμως γίνεται από σελίδα που ανήκει στην google. Περισσότερα για τα google analytics ακολουθούν στο επόμενο κεφάλαιο που αφορά την διαχείριση της εφαρμογής.

➤ AdminExile

Οι τρύπες ασφαλείας που υπάρχουν στα CMS έχουν γίνει πολλές φορές θέμα συζητήσεως για τους web developers. Αυτό έκανε προγραμματιστές να ασχοληθούν με κομμάτια κώδικα που θα προστατεύουν ή θα αποκρύπτουν ευαίσθητα αρχεία και βάσεις δεδομένων. Μια δοκιμασμένη λύση και το πιο δημοφιλές plug-in για δωρεάν θωράκιση της εφαρμογής μας είναι το AdminExile. Όπως κάθε plug-in ενεργοποιείται αρχικά από τον Plugin Manager. Μερικές από τις δυνατότητες που παρέχει είναι αλλαγή του URL για την είσοδο στο σύστημα, blacklist για επικίνδυνες IP και όριο στις προσπάθειες για login. Απαραίτητη εφαρμογή, αλλά χρειάζεται προσοχή γιατί στο διαδίκτυο υπάρχουν πολλές συζητήσεις σε forums με administrators που λόγω απροσεξίας κλειδώνονται έξω από το σύστημα και ψάχνουν λύσεις.

➤ EFSEO

Το EFSEO ή αλλιώς Easy Frontend SEO είναι ένα plug-in το οποίο απλοποιεί τις διαδικασίες του SEO, όπως για παράδειγμα τον ορισμό των metadata, καθώς τα μεταφέρει στο frontend της εφαρμογής. Ανοίγοντας τις σελίδες ο administrator βλέπει στο πάνω δεξιά κομμάτι της οθόνης του σύμβολα που του δηλώνουν αν η σελίδα έχει ολοκληρωμένα στοιχεία. Πατώντας τα σύμβολα ανοίγει ένα παράθυρο που του επιτρέπει να συμπληρώσει τα στοιχεία κάθε σελίδας. Το μόνο αρνητικό είναι ότι αν προκύψει κάποιο πρόβλημα τα documentation είναι περίπλοκα και το support υπάρχει μόνο για

τους επί πληρωμή χρήστες. Αυτό το plug-in βοηθάει μόνο στην εξοικονόμηση χρόνου καθώς το σύστημα του Joomla είναι φτιαγμένο για να μας βοηθάει στο μέγιστο όσον αφορά το SEO. Εάν δεν επιθυμούμε να εγκαταστήσουμε πολλά plug-ins ή απλώς εμπιστευόμαστε περισσότερο τις δυνατότητες του συστήματος αποφεύγουμε αυτό το extension.

➤ RedactorJS Editor

Αν και δεν χρησιμοποιήθηκε τελικά το συγκεκριμένο extension για την κατασκευή του caferpatras.gr, θα ήταν παράληψη να μην αναφερθεί. Κατά την διαδικασία εξοικείωσης μας στο σύστημα του Joomla συνήθως ψάχνουμε για κάποιον editor ο οποίος θα μας βοηθήσει να εξοικονομήσουμε χρόνο με τις παροχές και τα shortcodes του. Μετά από μια μικρή περιήγηση στην σελίδα του Joomla βρήκαμε τον RedactorJS Editor, ο οποίος παρότι ήταν χρήσιμος, δεν ήταν αυτό που χρειαζόμασταν και διαγράφηκε από το σύστημα. Είναι απαραίτητο όμως να γνωρίζει κάποιος ότι είναι σε θέση να αναβαθμίσει τις δυνατότητες του editor με κάποιο extension.

➤ Smart Search

Για το τέλος κρατήσαμε ένα component. Σε αυτό το σημείο, και λόγω των προηγούμενων παραδειγμάτων, είναι πιο εύκολο να καταλάβουμε την διαφορά που έχουν με τα extensions. Τα documentations του Joomla περιέχουν την ερώτηση: “είναι τελικά τα components extensions;”. Η απάντηση που δίνουν περιορίζεται σε δύο λέξεις: Kind of. Για να καταλάβουμε την λειτουργία των components θα προχωρήσουμε στο παράδειγμα με το Smart Search το οποίο χρησιμοποιούμε και στην τελική εφαρμογή. Όπως φαίνεται και από το όνομα του, αυτό το component δημιουργεί μια μπάρα αναζήτησης, ή καλύτερα, ένα search field στο σημείο όπου θα το ορίσουμε. Είναι κατανοητό λοιπόν, πως για να έχουμε ολοκληρωμένη εικόνα πρέπει η εφαρμογή μας να περιέχει κάποιες σελίδες ως προς αναζήτηση. Όταν υπάρχουν σελίδες στην εφαρμογή βρίσκουμε στο οριζόντιο μενού του συστήματος την επιλογή Components και πατάμε στο Smart Search. Εκεί παρατηρούμε πως οι ρυθμίσεις ενός component είναι πιο περίπλοκες. Υπάρχει ακόμα και μενού στο πάνω μέρος με επιλογές για τα σημεία που θα του επιτρέπουμε να επιδράσει στην εφαρμογή. Από το μενού πατάμε στην επιλογή index, ώστε το component να βρει όλες τις σελίδες, τα tags και τις κατηγορίες. Μπορούμε εάν το επιθυμούμε να αποκρύψουμε στοιχεία και σελίδες από την αναζήτηση. Πιέζουμε το κουμπί publish, όμως για να ολοκληρωθεί η τοποθέτηση ενός component στην εφαρμογή, πρέπει υποχρεωτικά να το δηλώσουμε ως module. Οπότε αποθηκεύουμε και πηγαίνουμε στον Module Manager όπου βρίσκουμε και πατάμε το κουμπί +NEW το οποίο θα μας βοηθήσει να κατασκευάσουμε ένα καινούριο module στην λίστα. Όταν το σύστημα μας ρωτάει τι είδους module επιθυμούμε, επιλέγουμε το Smart Search και μεταφερόμαστε στο γνώριμο περιβάλλον επεξεργασίας των module. Εκεί υπάρχουν κάποιες ρυθμίσεις που αφορούν το styling και όχι την λειτουργία του component. Τελειώνοντας, επιλέγουμε την θέση που θέλουμε να βρεθεί στην εφαρμογή και κάνουμε την ενεργοποίηση. Επιτέλους το component είναι έτοιμο και λειτουργικό στο frontend της εφαρμογής. Κάθε component που μας παρέχει το Joomla, επιδρά με τις σελίδες μας ή τα

στοιχεία μας ως administrator και χρειάζεται μια αρχική παραμετροποίηση πριν το κάνουμε publish. Από εκεί και έπειτα, μπορούμε να το χειριστούμε σαν ένα module που βρήκαμε στο διαδίκτυο.

4.2 Δημιουργία σελίδων της εφαρμογής

Στην εποχή που ζούμε, υπάρχει τεράστιος ανταγωνισμός για το που θα διαθέσει ο κόσμος τον ελεύθερο χρόνο του, κάτι που εν τέλει, εκπαίδευσε τους ανθρώπους να μην συμβιβάζονται με το μέτριο. Σημασία για την επιτυχία μιας εφαρμογής δεν έχει μόνο η χρηστική της αξία, αλλά και η σχεδιαστική, εάν δηλαδή χρησιμοποιήσαμε ευδιάκριτο λογότυπο, εάν επιλέξαμε αξιόπιστα χρώματα, εάν η γραμματοσειρά πηγάζει επαγγελματισμό. Το Joomla είναι σε θέση να απλοποιήσει την διαδικασία παραγωγής τόσο, ώστε μέσα σε λίγα λεπτά να μπορούμε να παράγουμε νέες σελίδες, ενώ μέσα σε ώρες μια ολοκληρωμένη εφαρμογή.

4.2.1 Διαμόρφωση του αρχείου index.php

Αναφέραμε και νωρίτερα πως το αρχείο index.php είναι η βάση όλων των αρχείων που παράγονται για την εφαρμογή μας. Αυτό σημαίνει πως αρχικά πρέπει να το διαμορφώσουμε κατάλληλα ώστε το αποτέλεσμα που παράγει να μας καλύπτει. Υπάρχουν κάποια πράγματα που παραμένουν σταθερά σε όλες τις σελίδες μιας διαδικτυακής εφαρμογής. Για παράδειγμα, οι κλήσεις των αρχείων CSS πρέπει να γίνουν εδώ ώστε να περιληφθούν σε όλη την εφαρμογή, όπως και το κάλεσμα των βιβλιοθηκών του jquery. Από δημιουργική άποψη, πράγματα που χρειαζόμαστε σε κάθε σελίδα είναι το header, το μενού και το footer. Είναι στοιχεία τα οποία μπορούμε να τα κατασκευάσουμε μία φορά και να μην χρειαστεί να ασχοληθούμε ξανά.

Από την στιγμή που εδώ θα φτιάξουμε το μοναδικό <head> field που χρειάζεται η εφαρμογή μας, πρέπει να είμαστε προσεχτικοί να περιλάβουμε όλα εκείνα τα στοιχεία που κάνουν άρατη δουλειά στο πίσω μέρος του κώδικα. Ξέρουμε για παράδειγμα πως η κωδικοποίηση χαρακτήρων, ώστε να περιλαμβάνονται και τα Ελληνικά πρέπει να δηλωθεί εδώ με την εντολή `<meta charset="utf-8">`. Για κάποιον που θέλει να περιλάβει λειτουργίες των jquery βιβλιοθηκών χωρίς να τις εγκαταστήσει στην εφαρμογή του, η λύση είναι να συμπεριλάβει στο head του index.php την εντολή `<script src="https://code.jquery.com/jquery-1.10.2.js"></script>`.

Στην περίπτωση του μενού που αναφέραμε νωρίτερα, υπάρχει η πιθανότητα να μην θέλουμε να το κατασκευάσουμε οι ίδιοι, αλλά να προτιμούμε την αυτόματη παραγωγή από το σύστημα. Σε αυτή την περίπτωση δεν προσθέτουμε καθόλου κώδικα και θα δουλέψουμε μέσω του γραφικού περιβάλλοντος όπως θα δούμε λίγο αργότερα. Γενικότερα οτιδήποτε θέλουμε να υπάρχει σταθερά σε όλες τις σελίδες μας πρέπει να το κατασκευάσουμε εδώ. Με αυτόν τον τρόπο προκύπτει ένα ζήτημα. Όταν κάτι το χρειαζόμαστε σε όλες τις σελίδες εκτός από μία πως πράττουμε;

Για παράδειγμα, η εφαρμογή μου μπορεί να έχει 100 σελίδες και να θέλω κάθε σελίδα να περιλαμβάνει ένα div με τα στοιχεία επικοινωνίας μου, εκτός από δύο, την αρχική σελίδα και στην σελίδα με την φόρμα επικοινωνίας. Σε αυτή τη περίπτωση η βέλτιστη επιλογή δεν είναι να προσθέσω το div σε κάθε μία από τις 98 σελίδες, αλλά να το προσθέσω στο index.php και έπειτα να το αφαιρέσω από την αρχική και την σελίδα επικοινωνίας με χρήση Javascript. Πολύ απλά ορίζω το div μου με ένα id ή σε μια κλάση και το αποκρύπτω με τον ακόλουθο κώδικα:

```
1 <script>
2   $(document).ready(function() {
3       $(".contact-div").hide()
4   });
5 </script>
```

Με αυτό τον τρόπο αποκρύπτω από την σελίδα οτιδήποτε περιέχεται στην κλάση contact-div. Προφανώς μπορώ να το χρησιμοποιήσω και για ολόκληρα fields τα οποία επιθυμώ να αποκρύψω από συγκεκριμένες σελίδες. Για παράδειγμα, το header στην HTML5 το δηλώνουμε με το tag <header> στο index αρχείο, άρα το path της εντολής hide θα αλλάξει ως εξής:

```
1 <script>
2   $(document).ready(function() {
3       $("header").hide()
4   });
5 </script>
```

4.2.2 Στάδια δημιουργίας νέας σελίδας

Για κάθε νέα σελίδα που θα προσθέσουμε στην εφαρμογή μας θα πρέπει να ακολουθήσουμε το μονοπάτι Contact / Article Manager / Add New Article ξεκινώντας από το οριζόντιο μενού στο πάνω μέρος. Έτσι μεταφερόμαστε στο μέρος όπου θα περάσουμε τις περισσότερες ώρες της δημιουργίας. Εκεί υπάρχει μία οριζόντια στήλη με έξι διαφορετικά tabs, όπου πρέπει να εξερευνηθούν πριν ξεκινήσουμε καθώς κάποια περιέχουν ρυθμίσεις που προσαρμόζουν το αποτέλεσμα που μας εμφανίζεται. Κάτω από την στήλη και πιάνοντας το μεγαλύτερο μέρος της σελίδας, συναντούμε τον editor, ενώ στα δεξιά μας βλέπουμε ένα μενού με κάποιες βασικές επιλογές. Από την στιγμή που βρισκόμαστε σε αυτό το σημείο, σημαίνει πως έχουμε ήδη δημιουργήσει ένα καινούριο αντίγραφο του index.php στον Server.

➤ Στάδιο 1: Προγραμματισμός

Χωρίς κάποιο plug-in για τον editor, αναγκαστικά κατασκευάζουμε κώδικα σε γλώσσα HTML, προσέχοντας σύνταξη και ορθογραφία εντολών καθώς τα λάθη δεν τονίζονται όπως στους επαγγελματικούς editors. Με αυτήν την λογική είναι προφανές πως εάν ο προγραμματιστής εργάζεται συστηματικά σε κάποιον editor, δημιουργεί εκεί καθε σελίδα και στο τέλος μεταφέρει τον κώδικα.

Ότι γράφουμε εδώ μεταφέρεται μέσα στο body της σελίδας άρα υποκύπτει στους σχεδιαστικούς κανόνες που του ορίζουμε στο CSS. Αν θέλουμε να συμπεριλάβουμε τα πάντα σε ένα wrapper div μπορούμε να το προσθέσουμε στο index πάνω από το jdoc, αλλά εγώ προσωπικά προτιμώ να τον ορίζω σε κάθε νέα σελίδα ώστε ο κώδικας να παρουσιάζεται οπτικά οργανωμένος.

➤ Στάδιο 2: Κοινοποίηση

Η κοινοποίηση της σελίδας, γνωστή και ως publishing περιλαμβάνει λίγα αλλά βασικά βήματα για την ολοκλήρωση της. Αρχικά η σελίδα μας είναι σε status unpublished και αν προσπαθήσουμε να της αλλάξουμε κατάσταση θα μας βγάλει μήνυμα λάθους. Αυτό συμβαίνει γιατί το πρώτο βήμα της διαδικασίας, δηλαδή η ονομασία της σελίδας, είναι υποχρεωτικό. Δίπλα από την ονομασία βρίσκεται το Alias που είτε το συμπληρώνουμε εμείς, είτε παράγεται αυτόματα από το όνομα που δώσαμε. Εάν στα global configuration του Joomla έχουμε δηλώσει τα URL μας να έχουν το όνομα του άρθρου και όχι το id, τότε ότι δηλώσουμε στο Alias πηγαίνει εκεί.

Τα υπόλοιπα βήματα δεν είναι υποχρεωτικά αλλά είναι σημαντικά γιατί μας βοηθούν με την διαμόρφωση των σελίδων και την κατάταξη μας στις μηχανές αναζήτησης. Πατώντας λοιπόν, στο δεύτερο tab με τίτλο publishing βρίσκουμε ένα μενού γεμάτο επιλογές. Οι σημαντικότερες είναι το Created by, γιατί σε εφαρμογές με διαφορετικούς αρθρογράφους είναι σημαντικό να υπάρχει το όνομα του δημιουργού, τα dates που δηλώνουμε τις μέρες δημιουργίας και επεξεργασίας των σελίδων και τα metadata.

Τα metadata είναι απαραίτητες πληροφορίες που δίνουμε για την σελίδα μας οι οποίες απευθύνονται στις μηχανές αναζήτησης. Συγκεκριμένα τα Meta Keywords είναι λέξεις κλειδιά που πρέπει να περιγράψουν συνοπτικά τα περιεχόμενα της συγκεκριμένης σελίδας, ώστε η μηχανές αναζήτησης να παρουσιάζουν την σελίδα σε χρήστες που αναζητούν αυτές τις λέξεις. Το Meta Description είναι το μικρό κειμενάκι που παρουσιάζεται κάτω από το URL των σελίδων στις μηχανές αναζήτησης και περιγράφει το περιεχόμενο τους. Πολλές φορές παρατηρούμε πως το κειμενάκι είναι απόσπασμα του περιεχομένου της εφαρμογής κάτι που είναι λάθος και δείχνει προχειρότητα στην κατασκευή.

Όταν τελειώσουμε με την εξερεύνηση όλων των επιλογών του μενού γυρνάμε στο πρώτο tab και αλλάζουμε το status σε published. Η δουλειά μας σε αυτή τη σελίδα τελειώσε πλέον. Επιστρέφουμε εδώ μόνο για διορθώσεις και επεξεργασία των κειμένων της σελίδας. Κάνοντας όμως, αναζήτηση στον Browser για το URL της σελίδας που δηλώσαμε δεν παρουσιάζει κανένα αποτέλεσμα. Αυτό συμβαίνει γιατί το Joomla απαιτεί όλες οι σελίδες της εφαρμογής να ανήκουν σε κάποιο μενού, έστω και αν αυτό το μενού δεν το παρουσιάζουμε στην σελίδα μας.

➤ Στάδιο 3: Προσθήκη σε Μενού

Αναγκαστικά και για να δούμε ποιες είναι οι αρχικές επιλογές μας, ακολουθούμε στο οριζόντιο μενού το μονοπάτι Menus / Menu Manager. Εκεί παρότι βλέπουμε δύο διαθέσιμα μενού, το δεύτερο με όνομα User Menu, δεν είναι για τοποθέτηση νέων σελίδων, αλλά για σελίδες του συστήματος. Το πρώτο έχει το όνομα Main Menu και συνήθως περιέχει τις βασικές σελίδες της εφαρμογής.

Στο Menu Manager μας δίνεται η δυνατότητα να ταξινομήσουμε όλες τις σελίδες της εφαρμογής μας, δημιουργώντας καινούρια μενού και υπομενού. Αναφέραμε νωρίτερα πως έχουμε την δυνατότητα αν θέλουμε να απενεργοποιήσουμε όλα τα μενού και να κατασκευάσουμε το δικό μας στο index.php. Όποια και αν είναι η απόφαση μας θα χρειαστεί να επεξεργαστούμε τα positions των μενού.

Αν δημιουργήσουμε καινούριο μενού πρέπει να του ορίσουμε το position που θα πάρει στην σελίδα μας, και πρέπει να απενεργοποιήσουμε το position στο ήδη υπάρχον, ώστε να μην εμφανίζεται πουθενά. Παρακάτω υπάρχει αναλυτικό παράδειγμα για τα μενού του caferatras.gr, ώστε να δούμε και πρακτικά και να ξεκαθαρίσουμε την διαδικασία. Όταν καταλήξουμε σε ποιο μενού θα τοποθετηθεί η σελίδα μας, μπαίνουμε μέσα του και πατούμε στην επιλογή Add New Item.

Εκεί αρχικά δίνουμε το όνομα και το Alias που δώσαμε και στην σελίδα. Έπειτα στην στήλη Menu Item Type επιλέγουμε το select και στο παράθυρο που ανοίγει ακολουθούμε το μονοπάτι Articles / Single Article. Σε αυτό το σημείο καλό είναι να κάνουμε και μια εξερεύνηση των επιλογών μας ώστε να ανακαλύψουμε άλλες δυνατότητες που μας προσφέρει το Joomla για μελλοντική χρήση. Για το τέλος στην στήλη Link επιλέγουμε το άρθρο που δημιουργήσαμε, το οποίο θα είναι στην πρώτη θέση της λίστας. Ελέγχουμε τις άλλες επιλογές που μας προσφέρονται, στα υπόλοιπα τέσσερα tabs για πιθανές αλλαγές στις ρυθμίσεις και αποχωρούμε κάνοντας Save & Close.

Η πρώτη σελίδα της εφαρμογής μας είναι γεγονός. Πλέον μπορούμε να την δούμε από τον browser και να διορθώσουμε τα λάθη που θα υπάρχουν. Τα τρία στάδια που αναφέραμε δημιουργούν έναν κύκλο, ο οποίος θα επαναλαμβάνεται για κάθε καινούρια σελίδα, μέχρι την ολοκλήρωση της διαδικτυακής εφαρμογής.

4.2.3 Κατηγορίες σελίδων

Κατά την διάρκεια της κοινοποίησης μιας σελίδας παρατηρούμε και μια επιλογή Category, η οποία αρχικώς έχει την τιμή Uncategorized. Οι κατηγορίες είναι ένας τρόπος που μας παρέχει το Joomla ώστε να κρατάμε τις σελίδες μας οργανωμένες και ομαδοποιημένες. Κάθε κατηγορία μπορεί να περιέχει υποκατηγορίες, οι οποίες με την σειρά τους μπορούν να περιέχουν αμέτρητες σελίδες. Εκτός από την οργάνωση όμως, οι κατηγορίες παρέχουν και άλλα πλεονεκτήματα.

Πριν τα δούμε όμως, πρέπει πρώτα να μάθουμε να δημιουργούμε τις κατηγορίες που χρειαζόμαστε. Η διαδικασία είναι όπως πάντα απλή. Το μονοπάτι που ακολουθούμε είναι Content / Category Manager / Add New Category, όπου ουσιαστικά το μόνο που χρειάζεται να κάνουμε είναι να δώσουμε το όνομα και να αποθηκεύσουμε. Έπειτα κατά

το publishing κάθε καινούριας σελίδας επιλέγουμε την κατηγορία μας από την στήλη Category.

Το σημαντικότερο λοιπόν, πλεονέκτημα που μας προσφέρουν οι κατηγορίες είναι το ότι είμαστε σε θέση με το πάτημα ενός κουμπιού να παράγουμε μια σελίδα η οποία θα εμφανίζει λίστα με όλα τα άρθρα που περιέχονται σε μια κατηγορία. Αυτό βοηθάει την εμπειρία πλοήγησης του χρήστη στην σελίδα, καθώς μπορούμε να του παρέχουμε τις βασικές κατηγορίες, να επιλέγει όποια επιθυμεί και οι σελίδες που την αφορούν να εμφανίζονται με χρονολογική σειρά. Όλες τις ρυθμίσεις για την σελίδα κατηγορίας τις πραγματοποιούμε γραφικά.

Η διαδικασία για κατασκευής αυτής της σελίδας ξεκινάει με την επιλογή ή την δημιουργία ενός μενού το οποίο θα περιέχει την σελίδα. Μέσα από το μενού επιλέγουμε το Add New Item. Αυτή τη φορά δεν θέλουμε να δημοσιεύσουμε Article, οπότε στην στήλη Menu Item Type θα επιλέξουμε ανάμεσα στα Article / Category List ή Article / Category Blog. Η διαφορά τους είναι πως στην πρώτη περίπτωση παρουσιάζεται λίστα με τα ονόματα των σελίδων και ο χρήστης τα ανοίγει πατώντας τα, ενώ στην δεύτερη περίπτωση τα άρθρα ολόκληρα παρουσιάζονται το ένα μετά το άλλο. Δίνουμε όνομα και Alias και προχωρούμε στα επόμενα tabs τα οποία περιέχουν τις ρυθμίσεις για την παραγόμενη λίστα. Αποθηκεύουμε και ελέγχουμε το αποτέλεσμα από τον browser.

Άλλο βασικό πλεονέκτημα των κατηγοριών είναι η αλληλεπίδρασή τους με τα plug-ins του συστήματος. Για παράδειγμα το Disqus που είδαμε νωρίτερα, μας δίνει την επιλογή να το συμπεριλάβουμε σε συγκεκριμένες κατηγορίες. Γενικά γίνεται εύκολα κατανοητό πως δεν είναι όλες οι σελίδες ίδιες και το να είμαστε σε θέση να τις κατηγοριοποιούμε και να τις επεξεργαζόμαστε ομαδικά είναι μεγάλη υπόθεση.

Τέλος πρέπει να αναφέρουμε πως οι κατηγορίες μπορούν να μας βοηθήσουν και στον Article Manager. Όταν χρειάζεται να επεξεργαστούμε συγκεκριμένες σελίδες, εάν τις έχουμε κατηγοριοποιήσει η διαδικασία αναζήτησης γίνεται παιχνιδάκι. Απλώς στην στήλη Category επιλέγουμε αυτήν που επιθυμούμε και εμφανίζονται μόνο οι σελίδες που ανήκουν σε αυτήν. Για μεγαλύτερη κατανόηση των κατηγοριών του Joomla, παρακάτω ακολουθεί παράδειγμα με τις σελίδες του caferpatras.gr κατηγοριοποιημένες.

4.2.4 Δημιουργία αρχικής σελίδας

Η αρχική σελίδα είναι πάντα η πρώτη εντύπωση του χρήστη για την εφαρμογή μας. Μια άσχημη και ασύμμετρη σελίδα, δύσκολα θα τραβήξει και θα κυρίως θα κρατήσει χρήστες. Σε κάθε εφαρμογή, εκτός από τα blogs, η αρχική σελίδα είναι μοναδική. Άρα είναι σημαντικό κατά την δημιουργία της αρχικής σελίδας, όπως και άλλων main pages της εφαρμογής, να μην έχουμε περιορισμούς και να αφήσουμε την φαντασία μας ελεύθερη.

Από προεπιλογή η αρχική σελίδα του συστήματος λέγεται Home και βρίσκεται στο Main Menu. Μπορούμε λοιπόν να πάμε και να τροποποιήσουμε τα περιεχόμενα της ώστε να παρουσιάσει το τελικό αποτέλεσμα που θέλουμε. Σε άλλη περίπτωση δημιουργούμε μια καινούρια σελίδα, την προσθέτουμε σε κάποιο μενού και την ορίζουμε εκεί ως αρχική σελίδα της εφαρμογής. Για να το κάνουμε αυτό, πολύ απλά, ανοίγουμε το μενού που την

ορίσαμε, και εμφανίζονται σε λίστα οι σελίδες που περιέχονται. Εκεί υπάρχει μία στήλη που λέγεται Home και έχει γκρι αστέρι σε κάθε καταχώρηση, εκτός της προεπιλεγμένης αρχικής σελίδας. Πιέζουμε το αστέρι της δικής μας αρχικής, αυτό γίνεται αυτόματα κίτρινο και η διαδικασία ολοκληρώθηκε. Βγαίνουμε από την λίστα και ελέγχουμε στον Browser.

Όπως για κάθε νέα εφαρμογή, έτσι και για το caferpatras.gr κατά την διάρκεια κατασκευής του δεν θέλαμε το URL της σελίδας να είναι ανενεργό, ούτε να προσθέσουμε κάποιο coming soon extension το οποίο απλώς θα εμφάνιζε μια εικόνα ως αρχική σελίδα. Οπότε δημιουργούμε ένα όμορφο coming soon page, το θέτουμε ως αρχική σελίδα και δουλεύουμε την κατασκευή της εφαρμογής μας στο παρασκήνιο.

4.3 Παρουσίαση του caferpatras.gr

Ο λόγος κατασκευής μιας διαδικτυακής εφαρμογής για τη συγκεκριμένη εργασία, ήταν ώστε να υπάρχουν παραδείγματα και να γίνεται πιο εύκολα κατανοητή η θεωρία της αξιοποίησης CMS. Ωστόσο δεν θα ήταν σωστό να φτιαχτεί κάτι πρόχειρο απλά για να υπάρχει. Από την στιγμή που μπήκα στην δημιουργική διαδικασία ήταν υποχρέωση μου να δημιουργήσω κάτι όμορφο και προσεγμένο στην λεπτομέρεια, που να είναι συνάμα απλό και να έχει λόγο ύπαρξης. Τελικά το caferpatras.gr μετά από μεγάλη έρευνα και προσπάθεια ολοκληρώθηκε και είναι μια εφαρμογή η οποία έχει στόχο να παρουσιάσει τις καφετέριες της Πάτρας με περιγραφές και φωτογραφίες. Απευθύνεται κυρίως σε νεαρούς Πατρινοί και σε φοιτητές ή επισκέπτες της πόλης, άρα το ύφος των κειμένων, τα χρώματα και ο γενικότερος σχεδιασμός επιλέχθηκαν με σκοπό να απευθύνονται στο συγκεκριμένο κοινό.

4.3.1 Μενού και σελίδες

Στην εφαρμογή, δεν ήθελα το μενού που κατασκευάζει αυτόματα το Joomla καθώς δεν χρειαζόμουν κάτι επαγγελματικό και σύνθετο, αλλά περισσότερο τρεις λέξεις που να περιέχουν συνδέσμους. Οπότε στην αρχική σελίδα θα βρει κάποιος τις λέξεις που ήθελα (Περιοχές, Ασχολίες, Απορίες) σχεδιασμένες με CSS ώστε να υπάρχει ανταπόκριση όταν ο χρήστης περάσει το ποντίκι ή πατήσει τον σύνδεσμο. Το μενού όμως, όπως αναφέραμε νωρίτερα, είναι απαραίτητο για την δημοσίευση των σελίδων, άρα όλες μου οι σελίδες πρέπει να ανήκουν κάπου. Για καλύτερη οργάνωση, πέρα από το Main Menu κατασκεύασα και ένα ακόμα που το ονόμασα Shops με σκοπό να αποθηκεύσω όλες τις σελίδες που αφορούν τις καφετέριες εκεί.

Τελειώνοντας την εφαρμογή ήθελα οι σελίδες όπως η Αρχική, οι σελίδες από το μενού και η φόρμα δήλωσης καφετεριών να ανήκουν στο Main Menu. Το μενού shops περιέχει την κάθε μοναδική καφετέρια της Πάτρας, και μέχρι στιγμής υπάρχουν πάνω από εκατό σελίδες, αριθμός που συνεχώς αυξάνεται. Η μεγάλη ποσότητα σελίδων στο σύστημα και κυρίως αυτών που βρίσκονται στο μενού Shops, καθιστά την ύπαρξη κατηγοριών σχεδόν απαραίτητη. Το μόνο ερώτημα ήταν σε τι κατηγορίες μπορώ να χωρίσω τις καφετέριες;

Η πρώτη μου σκέψη ήταν να τις χωρίσω ανάλογα τις παροχές τους, δηλαδή ξεχωριστή κατηγορία για αυτές που έχουν τηλεόραση, ξεχωριστή για αυτές που έχουν επιτραπέζια, ξεχωριστή για αυτές που έχουν φαγητό κτλ. Το πρόβλημα ήταν πως κάποιες καφετέριες διέθεταν παροχές που κάλυπταν περισσότερες από μια κατηγορίες, και ως γνωστόν κάθε καφετέρια μπορεί να ανήκει μόνο σε μία κατηγορία. Αποφάσισα λοιπόν, πως το μοναδικό που έχουν οι καφετέριες και με εξυπηρετεί στα σχέδια μου είναι η περιοχή στην οποία βρίσκονται. Τελικά οι σελίδες στην εφαρμογή διαχωρίζονται σε κατηγορίες όπως, Ψηλά Αλώνια, Μαρίνα Πατρών, Πλατεία Όλγας και πολλές άλλες παρόμοιες. Άρα όπως είδα νωρίτερα, μπορώ εύκολα πλέον να δημιουργήσω σελίδες οι οποίες θα έχουν το όνομα τις περιοχής και περιέχουν όλες τις καφετέριες που ανήκουν σε αυτήν. Χρήσιμο και για εμένα που θέλω οργάνωση των σελίδων ως administrator, και για τους χρήστες όπου μπορούν να αναζητήσουν καφετέριες βάση της περιοχής που θέλουν.

Η σκέψη όμως με τις παροχές για κάθε καφετέρια μου άρεσε και δεν ήθελα να την αποχωριστώ. Αποφάσισα λοιπόν να δημιουργήσω δέκα μικρά κομμάτια κώδικα, να τους δώσω ονομασίες με παροχές και να προσθέτω σε αυτά τις καφετέριες που ανταποκρίνονται. Τα ονόματα είναι: Γλυκά, Αναγνωστήρια, Τηλεόραση, Κατοικίδια, Καφές στο χέρι, Επιτραπέζια, Σνάκ, Ποτό, Εκδηλώσεις και Φαγητό. Η παρουσίαση των αποτελεσμάτων του κώδικα ήθελα να γίνεται στην αρχική, χωρίς να φορτώνουν καινούριες σελίδες, οπότε εκμεταλλευτικά τον φάκελο Html που κατασκεύασα στο template μου, έβαλα εκεί τον κώδικα και καλούσα τα κομμάτια AJAX εντολές όπως θα παρουσιαστεί λίγο αργότερα, στο κομμάτι του προγραμματισμού.

Τελευταίο σημαντικό κομμάτι είναι τα tags. Λογικά όλοι έχουμε συναντήσει άρθρα τα οποία στο τέλος παραθέτουν τα tags που τους τοποθέτησε ο δημιουργός. Μπορώ να χρησιμοποιήσω τις δυνατότητες του Joomla, ώστε ο χρήστης να κάνει αναζήτηση επιλέγοντας λέξεις κλειδιά που έχω δηλώσει στις σελίδες. Με αυτήν την λογική αποφάσισα κατά την δημιουργία κάθε σελίδας, να δώσω κάποιες συγκεκριμένες λέξεις που εξυπηρετούν στην καλύτερη περιγραφή των παροχών. Για παράδειγμα όταν η παροχή είναι η Τηλεόραση για αθλητικά γεγονότα, οι λέξεις κλειδιά που μπορεί να περιέχει κάποια καφετέρια είναι CosmoteTV ή Nova. Αντίστοιχα παραδείγματα υπάρχουν και για άλλες παροχές όπως τα επιτραπέζια, και οι εκδηλώσεις.

Οι σελίδες σε μια διαδικτυακή εφαρμογή πρέπει συνεχώς να εξελίσσονται ανάλογα με τις συνήθειες του χρήστη. Ειδικότερα σε εφαρμογές όπως το caferatras.gr ή άλλες ενημερωτικές, που συνεχώς πρέπει να προσθέτουν σελίδες ανάλογα με τα γεγονότα, δεν μπορείς να πεις ποτέ πως η δημιουργία σελίδων τελείωσε. Για να έχει επιτυχία η εφαρμογή πρέπει να ανανεώνεται πάντα λαμβάνοντας υπ' όψιν τις κινήσεις των χρηστών.

4.3.2 Εικόνες και λογότυπα

Οι εικόνες είναι βασικό στοιχείο στον προσδιορισμό του ύφους της εφαρμογής. Από τα λογότυπα των καφετεριών, μέχρι τα εικονίδια για την πλοήγηση χρειάστηκαν ώρες επεξεργασίας εικόνων στο Photoshop. Στην συγκεκριμένη εφαρμογή παίζουν μεγάλο ρόλο τα λογότυπα. Πέρα από το λογότυπο της ίδιας της εφαρμογής, ήθελα το λογότυπο της κάθε καφετερίας που θα παρουσιάζεται να ταιριάζει με το ύφος. Άρα για κάθε

καφετέρια έπρεπε είτε να επεξεργαστώ αυτό που έχουν, είτε να τους δημιουργήσω κάποιο δικό μου, σκεπτόμενος πάντα την μοναδικότητα της καφετέριας. Για παράδειγμα σε μια καφετέρια εμπνεύστηκα από τις ιδιαίτερες λάμπες που κοσμούν το εσωτερικό και δημιούργησα το σήμα της. Σε άλλες έμπνευση ήταν ένας ξεχωριστός τοίχος, κάποιο ιδιαίτερο στοιχείο διακόσμησης ή ακόμα και το όνομα της καφετέριας. Σημασία είχε να παρουσιαστεί κάτι το οποίο θα αντιπροσωπεύει μοναδικά την καφετέρια και να συνάδει με τις επιλογές μου στον σχεδιασμό της εφαρμογής.

Για το λογότυπο της εφαρμογής ήθελα κάτι ξεχωριστό που να παίζει έξυπνα με το όνομα της εφαρμογής. Γενικότερα δεν μου αρέσουν τα λογότυπα που περιέχουν ολόκληρες λέξεις, εκτός και αν αυτές καταφέρνουν με το σχεδιασμό τους να δημιουργούν μια γενικότερη εικόνα. Μετά από πολλά σχέδια κατέληξα σε ένα που παρουσιάζει μία κούπα του καφέ, προσαρμόζοντας πάνω της τα αρχικά της εφαρμογής, με το C να αποτελεί ταυτόχρονα και το χερούλι της κούπας. Αρχικά υπήρχε και το πιατάκι τις κούπας, το οποίο όμως θυσιάστηκε για λόγους απλότητας. Κάνοντας πολλές δοκιμές και πειράματα στο αρχικό σχέδιο, κατέληξα τελικώς σε μια ασημί κούπα, με σκιές που δίνουν βάθος και την ψευδαίσθηση του τρισδιάστατου, ενώ το P παρουσιάζεται ως τρύπα της κούπας. Δίνοντας ξανά βαρύτητα στο νεανικό ύφος της εφαρμογής, το λογότυπο στο μενού παρουσιάζεται πάνω σε δίσκο σερβιρίσματος, από αυτούς που χρησιμοποιούν συχνά οι καφετερίες.



Εικόνα 4.3-1 Αρχικό και τελικό λογότυπο της εφαρμογής

Εικόνες χρησιμοποιώ ακόμα και σε σελίδες όπως οι Περιοχές της Πάτρας ή Ασχολίες στην Πάτρα. Πιστεύω πως οι εικόνες λένε περισσότερα από τις λέξεις, και τις προτιμά ο περαστικός χρήστης που τα μεγάλα κείμενα πάντα τον κουράζουν. Με τις εικόνες στην εφαρμογή όμως πρέπει πάντα να είμαστε προσεχτικοί, καθώς υπάρχει ο κίνδυνος υπερφόρτωσης του Server κάτι που θα κάνει την απόκριση των σελίδων πολύ αργή. Έτσι προσπάθησα να με προγράμματα να ελαττώσω το μέγεθος των μεγάλων εικόνων χωρίς να ρίξω την ανάλυση, ενώ έθεσα όριο οι σελίδες για κάθε καφετέρια να περιέχουν έως πέντε φωτογραφίες. Παρακάτω παραθέτω κάποιες εικόνες που δημιουργήθηκαν ώστε να εξυπηρετούν το ύφος του cafepatras.gr.



Εικόνα 4.3-2 Εισαγωγική εικόνα στην σελίδα Περιοχές



Εικόνα 4.3-3 Εικονίδια που συμβολίζουν τις παροχές



Εικόνα 4.3-4 Παραδείγματα λογότυπων που δημιουργήθηκαν για την εφαρμογή

Όσον αφορά την επεξεργασία εικόνων που υπάρχουν μέσα στο σύστημα, το Joomla ανήκει στο CMS με μηδενικές δυνατότητες. Δεν υπάρχει καν επιλογή στο Media Manager που να μας οδηγεί σε κάποιον επεξεργαστή κατά την προβολή της εικόνας. Σε άλλα συστήματα μπορούμε να περικόψουμε ή να προσθέσουμε κείμενο πάνω σε εικόνες, αλλά έως εκεί.

4.3.3 Προχωρημένος Προγραμματισμός

Στο τελευταίο κομμάτι της παρουσίασης δεν γίνεται παρά να μιλήσουμε για προγραμματισμό. Όμως δεν θα παρουσιαστεί ολόκληρος ο κώδικας της εφαρμογής, γιατί το `caferatras.gr` υπάρχει ως παράδειγμα και όχι για να μας δείξει πως φτιάχνουμε μια εφαρμογή για παρουσίαση καφετεριών. Έτσι και αλλιώς υπάρχουν δύο βασικά προβλήματα. Πρώτον ο κώδικας όλων των σελίδων με όλες τις λεπτομέρειες είναι τεράστιος, και δεν είναι δυνατόν να συμπεριλάβουμε όλες τις γραμμές του. Δεύτερον ο κώδικας που περιέχουν τα `plug-in` συνήθως σπάει σε κομμάτια και χωρίζεται σε πολλές διαφορετικές σελίδες δυσκολεύοντας την παρουσίαση τους, η οποία αχρηστεύεται από το γεγονός ότι έτσι και αλλιώς είναι `open source`.

Γενικότερα δεν υπάρχει λόγος να παραθέσω τον όγκο που περιέχει πολύ απλά κομμάτια HTML και CSS. Υπάρχουν όμως κάποια μικρά πιο σύνθετα κομμάτια στον κώδικα που κάνουν την εφαρμογή να ξεχωρίζει. Παρακάτω παραθέτω πέντε σημεία στην σελίδα όπου για να πάρω το αποτέλεσμα που ήθελα, δεν χρησιμοποίησα το Joomla, αλλά τις γνώσεις μου στον προγραμματισμό.

➤ Rotate Εικόνων

Όπως είδαμε και νωρίτερα στην σελίδα υπάρχουν κάποια εικονίδια για να διαχωρίζονται οι καφετέριες ανάλογα τις παροχές τους. Ήθελα αυτά τα εικονίδια, όπως και το λογότυπο που έχει στρογγυλό `background`, με το πέρασμα του ποντικιού να αντιδρούν ώστε να βοηθούν τον χρήστη. Μετά από πολλά πειράματα κατέληξα πως το πιο όμορφο είναι να γίνεται μια μικρή περιστροφή προς τα δεξιά για όσο διαρκεί το `hovering` του δείκτη από το ποντίκι.

Ο κώδικας για αυτό το εφέ πραγματοποιήθηκε με την χρήση της γλώσσας CSS όπως παρουσιάζεται παρακάτω:

```
1 .rotate{
2   padding-right:0px;
3   padding-left:0px;
4   cursor: pointer; cursor: hand;
5   display: inline-block;
6   vertical-align: middle;
7   -webkit-transform: translateZ(0);
9   transform: translateZ(0);
10  box-shadow: 0 0 1px rgba(0, 0, 0, 0);
```

```

11  -webkit-backface-visibility: hidden;
12  backface-visibility: hidden;
13  -moz-osx-font-smoothing: grayscale;
14  -webkit-transition-duration: 0.3s;
15  transition-duration: 0.3s;
16  -webkit-transition-property: transform;
17  transition-property: transform;
18  }
19  .rotate:hover, .rotate:focus, .rotate:active {
20  -webkit-transform: rotate(9deg);
21  transform: rotate(9deg);
22  }

```

Αρχικά φτιάχνουμε μια κλάση που θα περιέχει τον κώδικα. Εδώ την ονομάζουμε **rotate**. Μέσα ξεκινάμε με εντολές **padding** που αφορούν τις αποστάσεις του αντικειμένου στον χώρο που θα δηλωθεί. Βάζουμε 0 pixels καθώς μπορεί να υπάρξει πρόβλημα με την περιστροφή όταν υπάρξει μετακίνηση. Μετά βλέπουμε την εντολή **cursor** όπου παίρνει τιμές pointer και hand. Με αυτές τις εντολές ο δείκτης του ποντικιού μετατρέπεται σε χέρι, ανάλογα τον browser. Το **display** το πρόσθεσα καθώς τα εικονίδια που θα ανήκουν στην κλάση θα είναι σε σταθερά και σε σειρά (inline block). Η εντολή **vertical-align** ορίζει την θέση που θα πάρει το στοιχείο στον κάθετο άξονα κάτι πολύ σημαντικό για σωστή περιστροφή. Από εκεί και κάτω βλέπουμε συχνά το **-webkit-** το οποίο είναι απαραίτητο ώστε κάποιες εντολές να διαβαστούν σίγουρα από τους browser Safari, Opera και Chrome. Το **transform** χρησιμοποιείται πάντα όταν υπάρχει animation ώστε να ειδοποιεί τους browser για αλλαγή στα pixel, το **box-shadow** το χρησιμοποιούμε για να προσθέσουμε σκιά στα αντικείμενα και το **backface-visibility** αποκρύπτει την εικόνα μετά την περιστροφή. Ακολουθεί το **moz-osx-font-smoothing** όπου με την τιμή grayscale βοηθά τα κείμενα που πιθανώς θα περιστραφούν να μην χάνουν τα όριά τους. Με το **transition-duration** εύκολα καταλαβαίνουμε πως ορίζουμε τον χρόνο μέχρι να ολοκληρωθεί η περιστροφή και τελευταίο έρχεται το **transition-property** στο οποίο δηλώνουμε το είδος του animation και το χρησιμοποιούμε ως σύνδεσμο με την επόμενη αγκύλη. Άρα πρέπει η τιμή του τελευταίου να δοθεί ως εντολή στο επόμενο κομμάτι.

Κλείνουμε την πρώτη αγκύλη και δηλώνουμε ξανά το όνομα της κλάσης αλλά με άνω κάτω τελείες ώστε να δηλώσουμε και γεγονός. Με τα hover, focus και active ενημερώνουμε τον browser πως τα παρακάτω θα γίνουν μόνο όταν ο δείκτης του ποντικιού βρίσκεται πάνω στο αντικείμενο. Επιτέλους έφτασε η στιγμή που δηλώνουμε το animation, με την εντολή **transform** την οποία δηλώσαμε ως τιμή λίγο νωρίτερα. Για την περιστροφή του αντικειμένου χρησιμοποιούμε την εντολή rotate δηλώνοντας και τις μοίρες που επιθυμούμε στην παρένθεση. Για την εφαρμογή έβαλα 9 μοίρες αλλά προφανώς μπορούμε να ορίσουμε την κλάση να περιστρέφει τα αντικείμενα ακόμα και 360 μοίρες.

Μία μικρή παρένθεση στο τέλος για κάτι που αφορά τα εικονίδια, αλλά όχι την περιστροφή. Τα εικονίδια που αφορούν τις παροχές βρίσκονται στην αρχική σελίδα, αλλά και στην σελίδα κάθε καφετέριας. Κάθε εικόνα αντιπροσωπεύει διαφορετική παροχή, άρα κατά την τοποθέτησή τους στην σελίδα τα δήλωνα και σε μια κλάση, η

οποία έχει το όνομα της παροχής. Για παράδειγμα όταν δήλωνα σε μια καφετέρια το εικονίδιο για τα επιτραπέζια, το τοποθετούσα σε μια κλάση games. Αυτές τις κλάσεις δεν τις έχω δηλώσει στο CSS αρχείο μου γιατί προς το παρόν δεν θέλω η κάθε παροχή να κάνει κάτι ξεχωριστό. Παρόλα αυτά είναι σωστή η τακτική μου, γιατί αν στο μέλλον αλλάξω γνώμη μπορώ σε μία αγκύλη στο CSS να πραγματοποιήσω ομαδική επεξεργασία με πρόσβαση σε όλες τις σελίδες.

➤ Φόρτωση HTML σελίδων με χρήση AJAX

Είδαμε λοιπόν τι γίνεται όταν ο δείκτης περνάει πάνω από τα εικονίδια. Τι γίνεται όμως όταν πατάμε ένα από αυτά; Στην αρχική σελίδα της εφαρμογής ο χρήστης έχει την δυνατότητα να κάνει αναζήτηση καφετεριών ανάλογα με τις επιθυμίες του. Αρχικά του παρουσιάζονται σε λίστα όλες οι καφετέριες της Πάτρας, αλλά αν για παράδειγμα έχει περάσει η ώρα και αυτός ψάχνει μόνο αυτές που να λειτουργούν ως cafe-bar δεν έχει παρά να πατήσει στο εικονίδιο του ποτού. Αυτόματα η λίστα θα αλλάξει χωρίς να ανανεωθεί ο browser. Αυτό υπερβαίνει τις δυνατότητες των γλωσσών HTML και CSS.

Για να πραγματοποιηθεί το εφέ χρειαζόμαστε μία μέθοδο που ονομάζεται AJAX. Τα αρχικά αυτής της μεθόδου προέρχονται από τις λέξεις Asynchronous Javascript And Xml. Θυμίζω πως η Javascript είναι μία δυναμική γλώσσα προγραμματισμού ενώ η Xml είναι γλώσσα η οποία σχεδιάστηκε για να αποθηκεύει και να μεταφέρει δεδομένα. Η λειτουργία του AJAX είναι να συνδέσει αυτές τις δύο γλώσσες προγραμματισμού ώστε να δημιουργήσει δυναμικά κομμάτια κώδικα, τα οποία θα ανταλλάσουν δεδομένα με τον Server, χωρίς όμως να γίνεται ανανέωση ολόκληρης της σελίδας. Αυτό βοηθάει πολύ τον χρήστη που βρίσκει άμεσα αυτό που ζητάει, αλλά κυρίως τον Server του ελαττώνονται τα αιτήματα από τον browser.

Το πρώτο βήμα είναι να κατασκευάσω διαφορετικές λίστες σε HTML αρχεία τα οποία θα έχουν το όνομα της παροχής που αντιπροσωπεύουν. Μπορώ με τον τρόπο που έδειξα και νωρίτερα να δημιουργήσω αυτά τα αρχεία από τον Article Manager του Joomla, αλλά προτίμησα να αξιοποιήσω τον άδαιο μέχρι τώρα HTML φάκελο που υπάρχει στο template μου. Τα αρχεία που θα φτιάξω θα περιέχουν μόνο ένα μοναδικό div το οποίο θα αντικαθιστά το div που εμφανίζεται αρχικά στην σελίδα. Άρα δεν υπάρχει λόγος να μπω στην διαδικασία δημιουργίας νέας σελίδας, τα φτιάχνω στον τοπικό editor και τα κάνω upload.

Έπειτα πρέπει να δώσω ένα id ή μια κλάση στον div της αρχικής σελίδας που θέλω να αντικαταστήσω. Επέλεξα να το ορίσω μέσα σε μια κλάση που λέγεται main. Κάνω το ίδιο και στα εικονίδια με τις παροχές τα οποία παίρνουν id ανάλογα την ιδιότητα τους. Έτσι ο κώδικας για να εμφανιστεί η λίστα των cafe-bar είναι ο ακόλουθος:

```
1 <script src="https://code.jquery.com/jquery-1.10.2.js">
2   $(document).ready(function(){
3     $("#bar").click(function(){
4       $(".main").load("templates/taza/html/bar.html");
5     });
```



```
6     });
7 </script>
```

Το παράδειγμα είναι απλό αλλά δείχνει την δύναμη της μεθόδου AJAX. Αρχικά συμπεριέλαβα στο απόσπασμα του κώδικα και το source για την έκδοση της βιβλιοθήκης jquery που χρησιμοποιούν τα function του παραδείγματος. Έπειτα βλέπουμε την συνάρτηση `$(document).ready(function){...}` την οποία προσθέτουμε πάντοτε όταν θέλουμε να εκτελέσουμε εντολές jquery. Ουσιαστικά ελέγχει προτού εκτελεστούν τα περιεχόμενα του εάν έχει φορτώσει κανονικά το έγγραφο. Μέσα του βρίσκεται η δεύτερη συνάρτηση που θα χρειαστούμε η οποία έχει το όνομα `$(...).click(function(){...}`. Η λειτουργία της είναι να εκτελέσει κάποιο γεγονός όταν πατηθεί από τον χρήστη αυτό που βρίσκεται στην πρώτη παρένθεση. Εδώ το `#bar` σημαίνει πως κατά την τοποθέτηση του εικονιδίου για τα cafe-bars μέσα στο image tag ορίσαμε και το `id="bar"`. Άρα όταν πατηθεί το εικονίδιο εκτελείται το `$(".main").load("templates/taza/html/bar.html");`. Λειτουργία της εντολής `.load` είναι να φορτώνει στην θέση που του ορίζει η πρώτη παρένθεση το αρχείο που θα βρει στο path της δεύτερης. Για τα υπόλοιπα εικονίδια επαναλαμβάνουμε τα δύο functions κάνοντας τις αλλαγές σε path και id.

Αυτός ο δυναμικός κώδικας κάνει τόσο γρήγορα το φόρτωμα του HTML αρχείου που μπορεί ο χρήστης να μην το καταλάβει. Για να δώσουμε χρόνο στον χρήστη να δει την αλλαγή που πραγματοποιείται είναι συνηθισμένη τακτική να βάζουμε κάποια εφέ στο transition. Για παράδειγμα ένα εφέ είναι το σταδιακό σβήσιμο του παλιού div που δημιουργείται με τον παρακάτω κώδικα:

```
1 <script src="https://code.jquery.com/jquery-1.10.2.js">
2 $(document).ready(function(){
3     $("#bar").click(function(){
4         $(".main").fadeOut(1000, function(){
5             $(".main").load("templates/taza/html/bar.html");
6         });
7     });
8 });
9 </script>
```

Προσθέσαμε απλά μια γραμμή η οποία προτού φορτωθεί το καινούριο div κάνει fade out το παλιό. Μέσα στην παρένθεση δίνουμε τον χρόνο που θέλουμε να διαρκέσει το σβήσιμο σε χιλιοστά του δευτερολέπτου. Στο παράδειγμά μας η διαδικασία θα διαρκέσει ένα δευτερόλεπτο.

➤ Απόκρυψη και εμφάνιση div με χρήση jQuery

Όπως είδαμε είναι πραγματικά απλό να καλούμε αρχεία από τον Server, αυτό όμως δεν σημαίνει πως είναι πάντα σωστό. Υπάρχουν περιπτώσεις που αυτό που χρειάζεται να εμφανιστεί σε μια σελίδα είναι ένα μικρό div ή ακόμα και μια λέξη. Δεν χρειάζεται να φτιάξουμε ολόκληρο αρχείο και να το καλούμε με <script> απλά για μια λέξη. Σε αυτές τις περιπτώσεις το jQuery μας δίνει την δυνατότητα να αποκρύπτουμε και να εμφανίζουμε κάποιο div όπου και όποτε θέλουμε.

Στην εφαρμογή caferpatras.gr ήθελα κάθε μοναδική σελίδα καφετέριας να περιέχει χάρτη με το ακριβές σημείο που βρίσκεται. Ο χάρτης όμως όταν πρόκειται για καφετέριες που βρίσκονται σε μία πόλη είναι για κάποιους χρήστες περιττός και απλά τους προσθέτει επιπλέον scroll. Ήταν αναγκαίο για την σωστή λειτουργία της εφαρμογής ο χάρτης να εμφανίζεται και να εξαφανίζεται ανάλογα τις επιθυμίες του χρήστη. Αποφάσισα λοιπόν, να υπάρχει ένα κείμενο που να λέει “Δες την περιοχή στον χάρτη” και δίπλα να έχει ένα βελάκι που να δείχνει κάτω. Αν ο χρήστης το πατήσει τότε αυτόματα θα εμφανίζεται ο χάρτης και το βελάκι θα δείχνει προς τα πάνω. Πατώντας το, όλα θα παίρνουν την αρχική τους θέση. Αυτό επιτυγχάνεται με τον παρακάτω τρόπο:

```
1 <div class="xartes">
2   <h2 class="text-center">
3     Δες την Περιοχή στον Χάρτη
4     
6     
8   </h2>
9 </div>
10 <div class="xartes-content">
11   <iframe> Σύνδεσμος από Google Maps </iframe>
12   <p class="text-center"> Διεύθυνση & Αριθμός</p>
13 </div>
14 <script>
15 $(document).ready(function() {
16   $(".xartes-content").hide();
17   $(".arrow-up").hide();
18   $(".xartes").click(function(){
19     $(".xartes-content").slideToggle(500);
20     $(".arrow-up, .arrow-down").toggle();
21   });
22 });
23 </script>
```

Το πρώτο div που ανήκει στην κλάση xartes περιέχει το κείμενο που θέλω να εμφανίζεται μαζί με τα δύο βελάκια. Το δεύτερο που ανήκει στην κλάση xartes-content

είναι το `div` που αρχικά θα είναι τελείως κρυμμένο και περιέχει τον χάρτη μαζί με την οδό που βρίσκεται η καφετέρια. Το `iframe` το οποίο θα περιέχει τον κώδικα του χάρτη το βρίσκουμε έτοιμο από τους χάρτες της Google και τον τοποθετούμε στην σελίδα. Η μόνη αλλαγή που μπορούμε να κάνουμε είναι στις διαστάσεις του. Στις δική μας σελίδα ο χάρτης απλώνεται σε όλο το πλάτος του `container` αλλά το ύψος του παραμένει πάντα σταθερό στα 400 pixel.

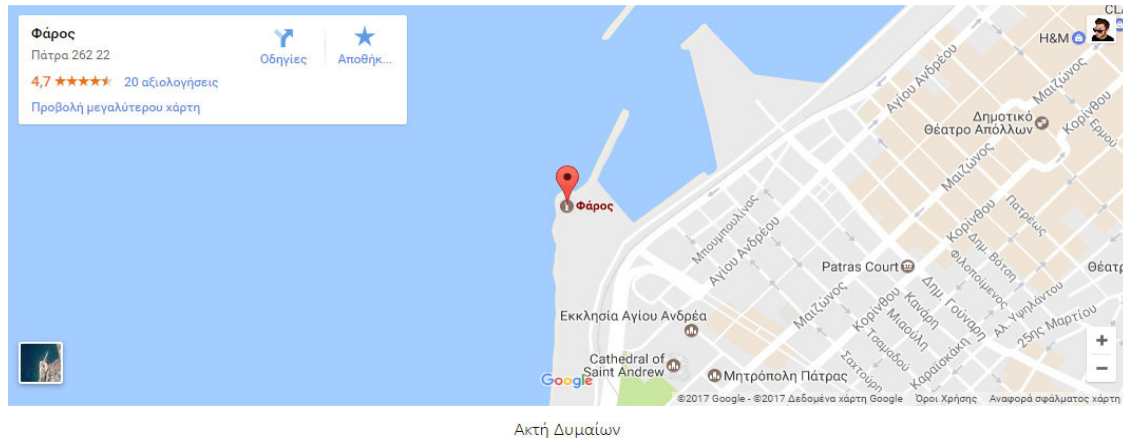
Η δουλειά όμως γίνεται με τις εντολές jQuery στην σειρά 14, οι οποίες ξεκινούν όπως πάντα με την συνάρτηση **ready**. Στις επόμενες δύο σειρές κρύβουμε τα περιεχόμενα των κλάσεων `xartes-content` και `arrow-up` με μία καινούρια εντολή, την **hide**. Τα περιεχόμενα που κρύφτηκαν όμως πρέπει να εμφανίζονται με το πάτημα στο κάτω βελάκι, οπότε χρειαζόμαστε ξανά την συνάρτηση **click** που είδαμε και νωρίτερα. Στην πρώτη παρένθεση της συνάρτησης θέσαμε την κλάση `xartes`, άρα ότι βρίσκεται εκεί λειτουργεί ως διακόπτης που με το πάτημα μας οδηγεί στο περιεχόμενο της συνάρτησης. Μέσα στην συνάρτηση συναντάμε την εντολή **toggle** σε δύο παραλλαγές. Αρχικά βλέπουμε το **slideToggle** το οποίο θα εμφανίσει με `slide` εφέ τον χάρτη στην σελίδα σε 500 χιλιοστά του δευτερολέπτου, και έπειτα υπάρχει το σκέτο **toggle** το οποίο αλλάζει την αντιστρέφει την κατάσταση των αντικειμένων. Αν για παράδειγμα κάτι είναι σε κατάσταση `hide` το κάνει `show` και αντίστροφα. Έτσι πάντα θα εμφανίζεται μόνο το ένα βελάκι στην σελίδα μας.

Αν παρατηρήσει κανείς τον κώδικα θα δει ότι τα βελάκια ανήκουν σε δύο κλάσεις. Τον ρόλο της πρώτης τον είδαμε, στην προηγούμενη συνάρτηση, αλλά δεν μιλήσαμε για τις κλάσεις `float` και `sink`. Αυτές οι κλάσεις έχουν ακριβώς τις ίδιες ιδιότητες με την περιστροφή των εικόνων που είδαμε νωρίτερα, απλά αντί να περιστρέφουν τα αντικείμενα, τα μετατοπίζουν στον `y` άξονα. Το πάνω βελάκι με `hover` του δείκτη μετακινείται -8 pixel στον κάθετο άξονα και πηγαίνει επάνω, ενώ αντίστοιχα το κάτω βελάκι πηγαίνει 8 pixel προς τα κάτω. Στο `animation` του κώδικα υπάρχει και μία μικρή καθυστέρηση για να δίνεται η ψευδαίσθηση πως η κίνηση που κάνουν τα βελάκια είναι φυσική.

Δες την Περιοχή στον Χάρτη ▼

Εικόνα 4.3-5 Ο χάρτης αρχικά είναι κρυμμένος στην σελίδα

Δες την Περιοχή στον Χάρτη ^



Εικόνα 4.3-6 Ο χάρτης μετά την αλλαγή κατάστασης της κλάσης του

➤ Parallax Scrolling

Ένα από τα πιο σημαντικά εφέ της εφαρμογής είναι το Parallax των εικόνων το οποίο το συναντάμε σχεδόν σε όλες τις σελίδες. Γενικότερα το εφέ αυτό ονομάζεται Parallax Scrolling και έχει πολλές παραλλαγές, αλλά εδώ το χρειαζόμαστε περισσότερο ως εισαγωγή στις σελίδες των καφετεριών. Το χρησιμοποιούμε για μεγάλες εικόνες στην σελίδα μας, στις οποίες φαίνεται όλο το πλάτος τους αλλά πάντα μόνο ένα μέρος του ύψους τους. Για παράδειγμα στην εφαρμογή οι περισσότερες εικόνες έχουν Height πάνω από 1000 pixels αλλά το μέγιστο height που επιτρέπω να φανεί είναι 500 pixels. Αυτό σημαίνει ότι ποτέ η εικόνα δεν θα εμφανίζεται ολόκληρη. Όταν ο χρήστης είναι ψηλά θα βλέπει μόνο το πάνω μέρος της, ενώ όταν κατεβαίνει θα βλέπει το κάτω.

Αυτή η τεχνική χρησιμοποιείται για να δώσει τρισδιάστατη ψευδαίσθηση στην σελίδα καθώς ο χρήστης νιώθει ότι η εικόνα είναι ένα επίπεδο κάτω από το κείμενο. Στην εφαρμογή μου όμως δεν ήθελα απλώς πετύχω αυτό το εφέ, αλλά να το εξελίξω τοποθετώντας σταθερό κείμενο πάνω στις εικόνες. Ας δούμε ένα παράδειγμα από την Πατρινή καφετέρια Amelie. Αρχικά παραθέτω τον HTML κώδικα που τοποθέτησα στην σελίδα της καφετέριας.

```
1 <section class="module parallax parallax-amelie">
2   <div class="box-con">
3     <h1>Amelie</h1>
4   </div>
5 </section>
```

Δήλωσα το section σε τρεις κλάσεις. Οι πρώτες δύο περιέχουν τον κώδικα του εφέ και θα τις δηλώνω κάθε φορά που θέλω να δημιουργήσω το Parallax Scrolling. Η τελευταία κλάση περιέχει την εικόνα, άρα κάθε φορά πρέπει να δίνω άλλο όνομα ανάλογα την εικόνα. Μέσα βρίσκεται ένα div με την κλάση του και μέσα σε αυτό μία H1 κεφαλίδα. Ο HTML κώδικας είναι πολύ βασικός άρα όλη η δουλειά γίνεται στο CSS. Στον παρακάτω CSS κώδικα, ο σχολιασμός των εντολών βρίσκεται στο πλάι σε μορφή σχολίου για ευκολότερη ανάγνωση.

```
1  .box-con {                               /*Ορίζω το μήκος την θέση και
2    max-width: 960px;                       τις αποστάσεις του div που
3    margin: 0 auto;                          περιέχει το κείμενο*/
4    padding-top:45px;
5  }
6  section.module:last-child {              /*Θέτω το τελευταίο αντικείμενο που
7    margin-bottom: 0;                        θα περιέχει το section στον πάτο*/
8  }
9  section.module p:last-child {            /*Θέτω το τελευταίο p στον πάτο*/
10   margin-bottom: 0;
11 }
12 section.module.content {                 /*Ορίζω τις αποστάσεις στο section*/
13   padding: 40px 0;
14 }
15 section.module.parallax {                /*Ο κώδικας του εφέ*/
16   height: 500px;                          /*Υψος του parallax*/
17   background-position: 50% 50%;           /*Εικόνα στο κέντρο των αξόνων*/
18   background-repeat: no-repeat;          /*Καμία επανάληψη εικόνας*/
19   background-attachment: fixed;          /*Η εικόνα πάντα σταθερή*/
20   background-size: cover;                /*Μέγεθος εικόνας όλη η οθόνη*/
21 }
22 section.module.parallax h1 {              /*Θέτω προτιμήσεις για την
23   color: #ffffff;                          εμφάνιση των κεφαλίδων H1*/
24   font-size: 48px;
25   line-height: 300px;
26   font-weight: 700;
27   text-align: center;
28   text-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
29 }
40 section.module.parallax-amelie {         /*Εικόνα του Parallax*/
41   background-image: url("amelie.jpg");
42 }
43 @media all and (min-width: 600px) {       /*Θέτω διαστάσεις H1
44   section.module.parallax h1 {           για κινητά*/
45     font-size: 96px;
46     -webkit-text-stroke-width: 1px;
47     -webkit-text-stroke-color: #999999;
```

```

48     }
49   }
50   @media all and (min-width: 960px) {           /*Θέτω διαστάσεις H1
51     section.module.parallax h1 {                για υπόλοιπες οθόνες*/
52       font-size: 140px;
53     }
54   }

```

➤ Responsive Boxes της αρχικής σελίδας

Το κυρίαρχο στοιχείο της αρχικής σελίδας είναι σίγουρα η λίστα με τα λογότυπα των καφετεριών. Τα λογότυπα αυτά όμως δεν είναι απλές εικόνες παρατεταγμένες σε σειρά, αλλά είναι περισσότερο κουτιά τα οποία ενώνονται και σχηματίζουν κάτι σαν μια μεγάλη βιβλιοθήκη. Αυτά τα κουτιά έχουν προφανώς κάποιες λειτουργίες παραπάνω από το να είναι απλοί σύνδεσμοι για τις σελίδες των καφετεριών. Για παράδειγμα αντιδρούν με το πέρασμα του δείκτη από το ποντίκι του χρήστη. Η σημαντικότερη λειτουργία τους όμως είναι πως ανάλογα το μήκος της οθόνης το μέγεθος τους αλλάζει αυτόματα ώστε να είναι ευδιάκριτα χωρίς να παραμορφώνονται.

Καταρχάς ας δούμε πρώτα πως καλούμε ένα τέτοιο κουτί στην αρχική μας σελίδα. Ως παράδειγμα θα πάρουμε το κουτί της καφετέριας Mikel:

```

1   <div class="item">
2     <a class="work" href="mikel">
3       
4       <div class="content">
5         <h2 class="title">Mikel</h2>
6       </div>
7     </a>
8   </div>

```

Το div του κουτιού το δημιουργούμε στην πρώτη σειρά, και το δηλώνουμε στην κλάση item. Μέσα εκεί υπάρχουν δύο αντικείμενα τα οποία λειτουργούν σαν σύνδεσμος της σελίδας του Mikel. Στην σειρά 3 βρίσκεται το λογότυπο της καφετέριας μέσα σε μία κλάση media, και στην σειρά 4 έχουμε ένα div που περιέχει το όνομα της. Οι ενέργειες που θέλουμε από αυτά τα κουτιά είναι τόσες που για κάθε αγκύλη που ανοίξαμε υπάρχει και μία κλάση. Δεν θα ασχοληθούμε με όλες τις κλάσεις γιατί οι περισσότερες απλά περιέχουν βασικές εντολές για τις γραμματοσειρές και το hovering των αντικειμένων. Ο κώδικας όμως που κάνει τα κουτιά να αλλάζουν σχηματισμό ανάλογα το μέγεθος της οθόνης κρύβεται μέσα στην κλάση item. Παρακάτω υπάρχει ο CSS κώδικας μόνο για αυτήν την κλάση.

```

1  .item{
2    width: 20%;
3    height: auto;
4    float: left;
5    position: relative;
6    overflow: hidden;
7    margin: 0;
8    padding: 0;
9  }

10 @media (max-width:1000px){
11   .item{
12     width: 33.33333333333333%;
13   }
14 }

15 @media (max-width:390px){
16   .item{
17     width: 50%;
18   }
19 }

```

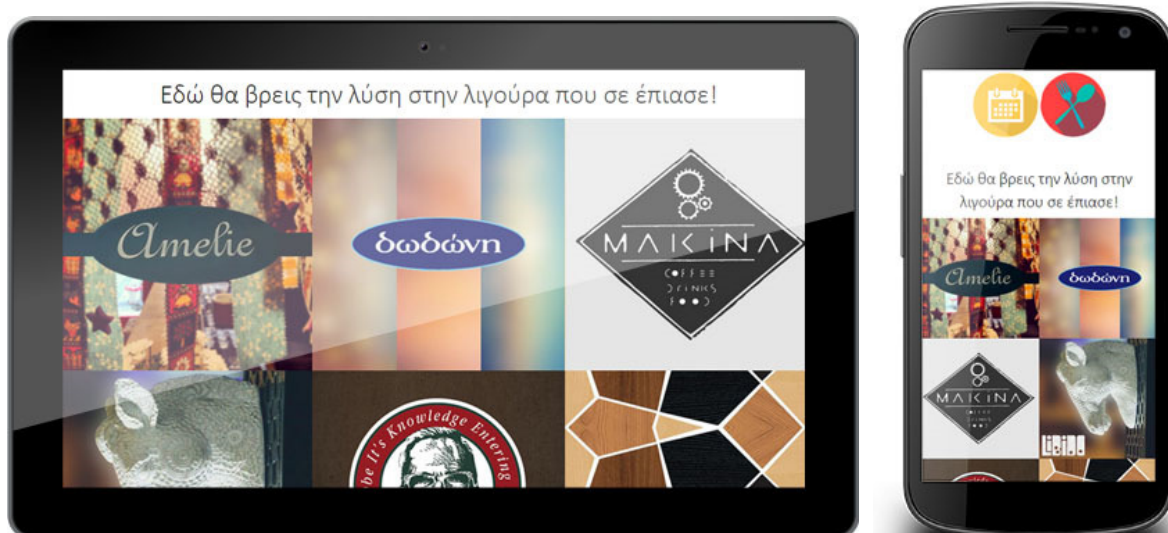
Στην πρώτη γραμμή ανοίγουμε την αγκύλη για να χτίσουμε τα κουτιά. Βλέπουμε πως η πρώτη εντολή αφορά το μήκος των κουτιών το οποίο δίνεται σε ποσοστό επί τις εκατό. Αυτό γιατί θέλουμε τα κουτιά να κάνουν scaling ανάλογα το μέγεθος της οθόνης του χρήστη. Στην δική μας περίπτωση αποφασίσαμε στις μεγάλες οθόνες να τοποθετούνται οριζόντια πέντε κουτιά, στις οθόνες των tablet και σε κινητά γυρισμένα στο πλάι να τοποθετούνται τρία και τέλος σε κινητά που βρίσκονται κάθετα δύο. Τον αριθμό τον επιλέξαμε μετά από δοκιμές και πειράματα από άποψη αισθητικής και ευχρηστίας.

Αρα για μεγάλες οθόνες δηλώνουμε το μέγεθος 20% όπως φαίνεται στην δεύτερη σειρά. Εύκολα καταλαβαίνουμε πως αν θέλαμε για παράδειγμα τέσσερα κουτιά, θα βάζαμε 25%. Στην σειρά 10 και με την εντολή **@media** δηλώνουμε εντολές που ο browser τις εκτελεί μόνο όταν ισχύει το μέγεθος οθόνης που βρίσκεται στην παρένθεση. Αρα για οθόνες έως 1000 pixel το width αλλάζει και παίρνει την τιμή 33.333(...)% . Με αυτόν τον τρόπο πλέον στις μικρές οθόνες θα μπαίνουν οριζόντια τρία κουτιά. Τέλος το ίδιο επαναλαμβάνουμε και για τα κινητά, όπου για τις περισσότερες οθόνες κινητών ο διαχωρισμός κάθετης με οριζόντιας οθόνης γίνεται στα 390 pixels, όσα ορίζουμε στην σειρά 15.

Πλέον τα κουτιά μας είναι responsive ανάλογα τα pixels της οθόνης του χρήστη. Την εντολή **@media** την χρησιμοποιώ σε κάθε σημείο της εφαρμογής όπου αντικείμενα ανάλογα την οθόνη χρειάζονται ιδιαίτερη διαχείριση. Το αποτέλεσμα του παραπάνω κώδικα φαίνεται μέσα από αυτές τις φωτογραφίες:



Εικόνα 4.3-7 Τα κουτιά σε μεγάλες οθόνες



Εικόνα 4.3-8 Τα κουτιά σε οθόνες tablet και κινητού

5. Προώθηση και διαχείριση διαδικτυακής εφαρμογής

Με την ολοκλήρωση και δημοσίευση της εφαρμογής στο διαδίκτυο ξεκινάει η περίοδος προώθησης και διαχείρισης. Είναι μία περίοδος που ουσιαστικά θα υπάρχει για όσο υπάρχει και η εφαρμογή. Με τον όρο διαχείριση εννοούμε τις ενέργειες που πραγματοποιούμε ώστε να διατηρούμε την εφαρμογή σε βέλτιστη λειτουργία. Το Joomla όπως θα δούμε μας δίνει λύσεις και extension ώστε να απλοποιεί κατά πολύ την δουλεία μας. Όσον αφορά την προώθηση, αυτή γίνεται με γνώμονα τις φιλοδοξίες και τις δυνατότητες της εφαρμογής, καθώς κάποιο μικρό project απλώς το προτείνουμε στον κύκλο μας, ενώ σε κάποιο μεγάλο project δίνουμε βαρύτητα και επενδύουμε στην διαφήμιση.

5.1 Μηχανισμοί προώθησης στις μηχανές αναζήτησης

Ο σκοπός της διαφήμισης για όλες τις εταιρίες είναι το κέρδος. Αυτό σημαίνει πως αν δεν περιμένουμε κέρδος από την εφαρμογή μας τότε η προώθησή της επιβάλλεται να είναι ανέξοδη. Επίσης πολλές φορές δεν είμαστε βέβαιοι πως ακόμα και κάποια πολυέξοδη διαφήμιση που πληρώσαμε θα βρει κοινό που ενδιαφέρεται και δεν θα εμφανιστεί σε άτομα που δεν έχουν σχέση με το προϊόν μας. Καταλήγουμε δηλαδή, πως αν και η προώθηση της σελίδας θεωρείται απαραίτητο συστατικό για την τελική επιτυχία της εφαρμογής, αυτό δεν σημαίνει ότι πρέπει να δαπανήσουμε μεγάλο χρηματικό ποσό. Αντίθετα μεγαλύτερη σημασία έχει να αξιοποιήσουμε έξυπνα τους δωρεάν πόρους που διαθέτουμε, όπως τις σελίδες κοινωνικής δικτύωσης και τα SEO και να στοχεύσουμε σε άτομα που ψάχνουν το προϊόν μας.

5.1.1 Search Engine Optimization (SEO)

Με τον όρο SEO δηλώνουμε τις δωρεάν ενέργειες που πραγματοποιούμε ώστε η εφαρμογή μας να ανέβει ψηλότερα από άλλες παρόμοιες εφαρμογές στην λίστα αποτελεσμάτων των μηχανών αναζήτησης. Εξαιρούνται φυσικά οι διαφημίσεις που πολλές φορές κατέχουν την πρώτη θέση καθώς βρίσκονται εκεί επί πληρωμή και όχι λόγω του SEO, άρα δεν μπορούμε να τις προσπεράσουμε.

Η θέση που θα κατέχει η εφαρμογή μας στην λίστα αποτελεσμάτων παίζει τεράστιο ρόλο στην τελική της επιτυχία. Αν ο πελάτης βρει νωρίς αυτό που ψάχνει, πιθανώς δεν θα μπει καν στην δική μας εφαρμογή. Άρα για παράδειγμα, εγώ θέλω το cafepatras.gr να εμφανίζεται στην πρώτη σελίδα αποτελεσμάτων όταν κάποιος θα αναζητήσει καφετέριες στην Πάτρα γιατί αλλιώς δεν υπάρχει περίπτωση ο χρήστης να ψάξει περισσότερο. Για

να καταλάβουμε πως λειτουργεί αυτή η διαδικασία ταξινόμησης πρέπει να αναλύσουμε τρεις όρους: Internet Bots, Crawling και Indexing.

Τα **Internet Bots** είναι προγράμματα τα οποία τρέχουν επαναλαμβανόμενα κομμάτια κώδικα (scripts) στο διαδίκτυο. Ο κώδικας συνήθως είναι πολύ απλώς αλλά η λειτουργία τους βασίζεται στην παραπάνω αντοχή και ταχύτητα που διαθέτουν σε σχέση με κάποιον άνθρωπο που θα πραγματοποιούσε την διαδικασία. Τα Bots που κατασκευάστηκαν ειδικά για την ταξινόμηση των εφαρμογών λέγονται αράχνες ή web crawlers. Η κάθε εταιρία όμως μπορεί να ονομάσει τα δικά της διαφορετικά. Για παράδειγμα η επίσημη ονομασία των Internet Bots της Google είναι Googlebot.

Η δουλειά αυτών των Bots είναι το **Crawling** στις εφαρμογές διαδικτύου και η επιστροφή αποτελεσμάτων. Για να βρεθούν γρήγορα οι καινούριες εφαρμογές χρειάζεται να οδηγούν σε αυτήν σύνδεσμοι από άλλους γνωστούς ιστότοπους όπου ελέγχουν τα Bots. Η διαδικασία του Crawling περιλαμβάνει την συλλογή στοιχείων (όπως λέξεις κλειδιά, metadata, τίτλους, σχόλια) και τον έλεγχο του κώδικα. Τα αποτελέσματα που συλλέγονται επιστρέφονται από τα Bots στην βάση των μηχανών αναζήτησης.

Όταν τα αποτελέσματα φτάσουν στην βάση εκεί ξεκινάει η διαδικασία του **Indexing**. Ένας αλγόριθμος λαμβάνει τα δεδομένα, υπολογίζει την αξία της εφαρμογής και την ταξινομεί ανάλογα στην λίστα της μηχανής αναζήτησης. Η διαδικασία δεν σταματά ποτέ ακόμα και για τις ήδη ταξινομημένες εφαρμογές, καθώς λαμβάνονται υπ' όψιν τα updates και η εξέλιξη μίας εφαρμογής και πραγματοποιούνται αλλαγές στην θέση της λίστας. Ο ακριβής αλγόριθμος που ταξινομεί τα αποτελέσματα για την κάθε μηχανή αναζήτησης είναι μυστικός, παρόλα αυτά υπάρχουν κάποιες ενέργειες που αποδεδειγμένα μας βοηθούν να ανέβουμε θέσεις. Κάποιες από αυτές τις ενέργειες τις πραγματοποιούμε μέσα από το περιβάλλον των CMS τα οποία προσπαθούν να απλοποιήσουν την διαδικασία του SEO.

5.1.2 Ο ρόλος των CMS στην προώθηση

Είδαμε τις διαδικασίες που κάνουν οι μηχανές αναζήτησης ώστε να ταξινομήσουν τις εφαρμογές. Τώρα ας εξετάσουμε κάποιους τρόπους που μπορούμε εμείς, ως διαφημιστές της εφαρμογής, να συμβάλουμε στην προώθηση της, με την βοήθεια των Συστημάτων Διαχείρισης Περιεχομένου.

➤ Υποβολή Sitemap

Σε μεγάλες εφαρμογές υπάρχει η πιθανότητα κάποιες σελίδες να μην είναι σωστά συνδεδεμένες μεταξύ τους, κάτι που δημιουργεί πρόβλημα στα Internet Bots. Ο καλύτερος τρόπος για να εκμεταλλευτούμε την παρουσία τους στην σελίδα είναι να τους δώσουμε ένα αρχείο το οποίο να περιέχει στοιχεία για όλες τις σελίδες της εφαρμογής και τα περιεχόμενα τους. Όταν ανανεώνουμε τις σελίδες και τα περιεχόμενα της εφαρμογής μας, ανανεώνουμε και το αρχείο sitemap έτσι ώστε να ειδοποιηθούν τα Internet Bots που ψάχνουν την σελίδα μας.

Το sitemap γράφεται συνήθως σε XML με εξαίρεση κάποιες περιπτώσεις μικρών εφαρμογών στις οποίες αρκεί ένα TXT αρχείο. Η κατασκευή του είναι απλή, αλλά πρέπει να υπάρχουν γνώσεις κυρίως για την δήλωση και περιγραφή εικόνων και βίντεο. Η δομή του sitemap για την δήλωση μιας σελίδας που περιέχει μία εικόνα και ένα βίντεο γίνεται ως εξής:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
3   xmlns:image="http://www.google.com/schemas/sitemap-image/1.1"
4   xmlns:video="http://www.google.com/schemas/sitemap-video/1.1">
5   <url>
6     <loc>Link Σελίδας</loc>
7     <image:image>
8       <image:loc>Link Εικόνας</image:loc>
9       <image:caption>Περιγραφή Εικόνας</image:caption>
10    </image:image>
11    <video:video>
12      <video:content_loc>
13        Link Βίντεο
14      </video:content_loc>
15      <video:player_loc allow_embed="yes" autoplay="ap=1">
16        Link Βίντεο στον Video Player
17      </video:player_loc>
18      <video:thumbnail_loc>
19        Link Thumbnail του βιντεο
20      </video:thumbnail_loc>
21      <video:title>Τίτλος Βίντεο</video:title>
22      <video:description>
23        Περιγραφή Βίντεο
24      </video:description>
25    </video:video>
26  </url>
27 </urlset>
```

Με την εντολή **xmlns** ορίζουμε τα namespaces προκαθορισμένων συνθηκών στην XML. Εδώ για παράδειγμα θέλουμε να συμπεριλάβουμε στο αρχείο μας εντολές για εικόνες και βίντεο, άρα οι γραμμές 3 και 4 είναι απαραίτητες. Από εκεί και πέρα το παράδειγμα μας για να είναι κατανοητό περιλαμβάνει μόνο ένα url σελίδας, που το δηλώνουμε στην γραμμή 6 και από ένα αρχείο εικόνας και βίντεο στις γραμμές 7 και 11 αντίστοιχα. Όπου τα γράμματα είναι bold (παχιά) θεωρούνται σχόλια και για να λειτουργήσει το αρχείο τα αντικαθιστούμε με τα ανάλογα στοιχεία που περιγράφουν.

Το να δημιουργήσουμε sitemap για μία μεγάλη εφαρμογή η οποία περιέχει πολλά media αντικείμενα είναι πολύπλοκο. Εδώ μας βοηθούν για άλλη μια φορά τα CMS όπου μας παρέχουν plug-ins τα οποία αυτόματα παράγουν και ανανεώνουν το sitemap αρχείο της

εφαρμογής μας. Αρκεί να τα εγκαταστήσουμε και να τους δώσουμε τις αρχικές ρυθμίσεις. Άλλος τρόπος για να παραχθεί εύκολα το αρχείο sitemap είναι μέσω online third-party tools τα οποία βάζοντας το link της εφαρμογής μας παράγουν έτοιμο το αρχείο.

Όταν έχουμε το αρχείο έτοιμο το μόνο που μένει είναι να το καταθέσουμε στους αλγόριθμους ταξινόμησης των μηχανών αναζήτησης. Η κάθε μηχανή αναζήτησης έχει την δική της διαδικασία την οποία μπορούμε να βρούμε εύκολα πληκτρολογώντας Submit Sitemap στην μπάρα αναζήτησής της. Για την υποβολή στην εταιρία Google για παράδειγμα, θα οδηγηθούμε σε ένα Search Console όπου αφού πιστοποιήσουμε ότι είμαστε διαχειριστές της εφαρμογής μπορούμε αρχικά να δοκιμάσουμε το αρχείο, και αν η Google το κρίνει κατάλληλο μας επιτρέπει να το δηλώσουμε ως sitemap.

➤ Metadata & Key Words

Όπως είδαμε και νωρίτερα κατά την διαδικασία δημιουργίας νέων σελίδων, το Joomla μας επιτρέπει να ορίσουμε Key Words και Metadata. Τα Key Words καταλαβαίνουμε από το όνομα, πως είναι κάποιες λέξεις κλειδιά που περιγράφουν το περιεχόμενο των κειμένων κάποιας σελίδας. Τα Metadata είναι δεδομένα των οποίων η λειτουργία τους είναι να περιγράφουν άλλα δεδομένα. Και τα δύο αυτά στοιχεία απευθύνονται στις μηχανές αναζήτησης και τους browser, με την διαφορά πως τα Metadata εμφανίζονται και στον χρήστη.

Η εφαρμογή μας θα λειτουργεί ακόμα και αν τα αφήσουμε κενά, όμως με το να τα δηλώσουμε βοηθάμε τα Internet Bots να καταχωρήσουν πιο σωστά την σελίδα μας. Για να πετύχει όμως αυτό καλό θα ήταν να ακολουθούμε κάποιους άγραφους νόμους που ορίζουν οι μηχανές αναζήτησης.

Τα Key Words βοηθάνε την προώθηση της εφαρμογής μας σε χρήστες, αλλά μόνο όταν αυτές οι λέξεις υπάρχουν και στο περιεχόμενο της σελίδας. Δεν μπορώ δηλαδή να δηλώνω κάποιες λέξεις κλειδιά, αλλά αυτές να μην υπάρχουν πουθενά στα κείμενα της σελίδας. Αυτό είναι κακό SEO και τα Bots το καταλαβαίνουν αγνοώντας πολλές φορές την εφαρμογή μας ή κρατώντας την στα τελευταία αποτελέσματα της αναζήτησης. Για την βέλτιστη αξιοποίηση των Key Words το ιδανικό σενάριο είναι οι λέξεις που θα δώσω να βρίσκονται στο URL της σελίδας, στον τίτλο της και μέσα στο κείμενο της.

Το ίδιο συμβαίνει και με τα Metadata τα οποία είναι το κειμενάκι που θα εμφανίζει η μηχανή αναζήτησης κάτω από τον σύνδεσμο της σελίδας μας. Αυτό το κειμενάκι πρέπει να περιέχει τις λέξεις κλειδιά ώστε να έχουμε καλύτερη ταξινόμηση στην λίστα αποτελεσμάτων. Πρέπει όμως και αυτό να είναι σχετικό με το περιεχόμενο της σελίδας, ώστε να δίνει σωστή περιγραφή και να μην προσπαθεί να ξεγελάσει τον χρήστη. Συνηθίζεται λοιπόν, να αξιοποιούμε τα extension των CMS, όπως είδαμε στο κεφάλαιο 4.1.3 ώστε να εκμεταλλευόμαστε πλήρως της δυνατότητας των SEO. Ένα καλό extension θα εντοπίσει τις σελίδες με κακό SEO και θα μας δώσει οδηγίες για την βελτίωση τους.

➤ Social Link Sharing

Οι αράχνες των μηχανών αναζήτησης χρησιμοποιούν τα μέσα κοινωνικής δικτύωσης ώστε να ανακαλύπτουν καινούριες εφαρμογές. Είναι πλέον διαδεδομένο κάθε νέα εφαρμογή να έχει δικό της profile σε κάθε μέσο. Με αυτόν τον τρόπο οι αράχνες ανακαλύπτουν καινούριες εφαρμογές στο διαδίκτυο και ανανεώνουν τα στοιχεία από αυτές που ήδη υπάρχουν στην βάση δεδομένων τους.

Πέρα από τα μέσα κοινωνικής δικτύωσης, ισχύει γενικότερα πως ο αλγόριθμος των μηχανών αναζήτησης επηρεάζεται από τον αριθμό συνδέσμων που οδηγούν στην σελίδα μας. Και όσο καλύτερη θέση στην κατάταξη έχει η σελίδα που περιέχει τον σύνδεσμο, τόσο περισσότερο βοηθάει την εφαρμογή μας. Με αυτό το σκεπτικό είναι χρήσιμο να προωθούμε την εφαρμογή μας ακόμα και σε διαδικτυακούς τηλεφωνικούς καταλόγους, καθώς η κάθε αναφορά προς την εφαρμογή μας μετράει.

Ο ρόλος των CMS σε αυτή την κατηγορία έρχεται για άλλη μια φορά μέσω των extension. Υπάρχουν πολλά extensions όπου επιτρέπουν στον χρήστη να κάνει share σελίδες της εφαρμογής στα μέσα κοινωνικής δικτύωσης. Άρα αυτόματα δημιουργούνται σύνδεσμοι που οδηγούν στην εφαρμογή μας μέσα από αξιόπιστους ιστότοπους. Αυτά τα extension τοποθετούν ένα div σε όποιο μέρος της σελίδας μας επιλέξουμε και περιέχουν τα πιο διαδεδομένα μέσα κοινωνικής δικτύωσης.

Αξίζει να αναφέρουμε εδώ πως και τα μέσα κοινωνικής δικτύωσης από την μεριά τους επιθυμούν την προώθηση εφαρμογών στον χώρο τους γιατί μια εφαρμογή με πολλά share λειτουργεί για αυτά σαν μια δωρεάν διαφήμιση. Η εταιρεία facebook στην προσπάθεια της να γίνει η επικρατέστερη πλατφόρμα για την προώθηση εφαρμογών, προτρέπει τους δημιουργούς να συμπεριλαμβάνουν στο head των σελίδων της εφαρμογής τον παρακάτω κώδικα:

```
1 <meta property="og:url" content="Link Σελίδας" />
2 <meta property="og:type" content="Τύπος (π.χ. article)" />
3 <meta property="og:image" content="Εικόνα Σελίδας" />
4 <meta property="og:title" content="Τίτλος Σελίδας" />
5 <meta property="og:description" content="Περιγραφή
   Περιεχομένου" />
```

Στην τιμή property βλέπουμε τύπους που από το όνομα καταλαβαίνουμε τι να περιμένουμε (π.χ. url, image), αλλά πάντα μπροστά υπάρχει το **og:**. Αυτό προέρχεται από τις λέξεις Open Graph οι οποίες βοηθούν το facebook να καταλαβαίνει πως θέλουμε να του περιγράψουμε κάποιο αντικείμενο. Αντικαθιστώντας τα bold κείμενα με τις αντίστοιχες τιμές για την εφαρμογή μας, πλέον όταν κοινοποιήσουμε την συγκεκριμένη σελίδα στο facebook (είτε στην αρχική σελίδα είτε σε παράθυρο διαλόγου) θα εμφανιστεί η **εικόνα**, ο **τίτλος** και η **περιγραφή** που εμείς δηλώσαμε. Και όλα αυτά ανάλογα το **type**, καθώς το να δηλώσουμε την σελίδα ως article θα παράγει διαφορετικό αποτέλεσμα από το να την δηλώσουμε ως book.

Ανάλογος κώδικας υπάρχει και για να εμφανιστούν στην σελίδα μας στοιχεία από το facebook, αλλά περιέχουν πολύπλοκο κώδικα. Για παράδειγμα την κοινοποίηση που πραγματοποιούμε στο profile μας, μπορούμε να την συμπεριλάβουμε αυτούσια σε σελίδα της εφαρμογής με την προϋπόθεση αρχικά να συμπεριλάβουμε κάποια Javascript functions.

5.1.3 Χειραγώγηση των μηχανών αναζήτησης

Οι μέθοδοι που αναφέραμε παραπάνω βοηθούν πρώτα από όλα τους χρήστες, ώστε να βρουν ακριβώς αυτό που ψάχνουν, έπειτα βοηθούν εμάς με την προώθηση της εφαρμογής μας, και τέλος βοηθούν τις μηχανές αναζήτησης να παράγουν καλύτερα αποτελέσματα. Υπάρχουν όμως στο διαδίκτυο εφαρμογές οι οποίες δεν νοιάζονται για τον χρήστη και προσπαθούν να τον εξαπατήσουν εκμεταλλευόμενοι τις μηχανές αναζήτησης. Αυτή η χειραγώγηση ονομάζεται Spamdexing (από τις λέξεις Spam και Indexing), όρος που χρησιμοποιήθηκε για πρώτη φορά το 1996 σε ένα άρθρο που αφορούσε την παράνομη εισαγωγή πορνογραφικού υλικού στο διαδίκτυο. Αλλιώς το Spamdexing αναφέρεται ως search engine spam, search engine poisoning, black-hat SEO, search spam ή απλώς web spam.

Η συχνότερη μέθοδος Spamdexing είναι χρησιμοποιώντας διαδεδομένα Key Words και κρυμμένα κείμενα στην σελίδα που δεν έχουν καμία σχέση με το πραγματικό περιεχόμενο. Δηλώνουν δηλαδή στα Metadata πως η σελίδα τους περιέχει κάτι δημοφιλές και γράφουν κείμενα με ανάλογο περιεχόμενο τα οποία όμως τα κρύβουν (π.χ. σε div με μηδενικές διαστάσεις, ορίζοντας position εκτός οθόνης, χρησιμοποιώντας χρώμα γραμμάτων ίδιο με το background) με αποτέλεσμα η μηχανή αναζήτησης να προτείνει μία κακόβουλη σελίδα σε χρήστες. Αυτό έχει αποτέλεσμα στην καλύτερη περίπτωση ο χρήστης να μην βρει αυτό που ψάχνει και στην συχνότερη περίπτωση να μολυνθεί ο υπολογιστής του με κακόβουλα προγράμματα.

Άλλος ένας τρόπος για να πραγματοποιηθεί Spamdexing είναι χρησιμοποιώντας ένα κομμάτι κώδικα το οποίο ελέγχει τα αιτήματα στον Server. Αυτός ο κώδικας έχει την ιδιότητα να καταλαβαίνει εάν το αίτημα πραγματοποιείται από άνθρωπο ή από κάποιο πρόγραμμα όπως τα Internet Bots. Ανάλογα με το είδος του αιτήματος εμφανίζεται άλλη σελίδα, ώστε οι μηχανές αναζήτησης να μην καταλαβαίνουν πως ο σύνδεσμος περιέχει κακόβουλο λογισμικό.

Ακόμα και αν το Spamdexing δεν χρησιμοποιείται κακόβουλα για τον χρήστη αλλά ως τρόπος διαφήμισης της εφαρμογής οι μηχανές αναζήτησης όταν το εντοπίσουν αποκλείουν μόνιμα ή προσωρινά τους συνδέσμους από τις λίστες τους. Αυτό γίνεται γιατί στα τέλη της δεκαετίας του 1990 η αξιοπιστία των μηχανών αμφισβητήθηκε έντονα από τους χρήστες λόγω τεράστιων κρουσμάτων Spamdexing. Αυτός είναι και ο σημαντικότερος λόγος που ο αλγόριθμος του Indexing παραμένει κρυφός.

Κάθε μηχανή αναζήτησης έχει ομάδα η οποία ασχολείται αποκλειστικά με την καταπολέμηση αυτών των ενεργειών, καθώς και με την επιβολή των ανάλογων τιμωριών. Αξίζει να αναφέρουμε πως το 2010, σύμφωνα με τον τότε CEO της εταιρίας Google, Eric Schmidt, η εταιρία στην προσπάθεια της να καταπολεμήσει την

χειραγώγηση των μηχανών αναζήτησης, πραγματοποίησε περισσότερες από 500 βασικές αλλαγές στον αλγόριθμό της, δηλαδή περίπου 1,5 αλλαγή την ημέρα.

5.2 Διαχείριση εφαρμογής μέσω CMS

Το κουραστικό κομμάτι τελειώνει κάπου εδώ, καθώς μετά την προώθηση μέσω των CMS, η δουλειά του προγραμματιστή περιορίζεται στην διαχείριση της εφαρμογής και στην συντήρηση της. Για τον τομέα της συντήρησης αναφέραμε και νωρίτερα πως πολλά θέματα μπορεί να προκύψουν, πρώτα απ' όλα ασφαλείας και έπειτα λόγω των αναβαθμίσεων, οπότε θα εξερευνήσουμε τις δυνατότητες του Joomla στον συγκεκριμένο τομέα. Θα δούμε επίσης πως γίνεται να εξελίξουμε την εφαρμογή μας και κυρίως τα έσοδα μας με την βοήθεια των extensions.

5.2.1 Συντήρηση εφαρμογής

Συντήρηση χρειάζονται όλες οι εφαρμογές, απλώς η καθεμία στο δικό της ποσοστό, ανάλογα το μέγεθος της και την επισκεψιμότητα της. Μεγάλο μέρος της συντήρησης σε μία εφαρμογή που κατασκευάστηκε με την χρήση CMS αποτελούν τα updates. Τα updates έχουν τον δικό τους χώρο στην τελευταία θέση του αριστερού κάθετου μενού στο σύστημα Joomla. Η διαδικασία τους είναι σχεδόν αυτοματοποιημένη, αρκεί απλώς να τους δώσουμε την έγκριση. Ανάλογα με την λειτουργία τους επεξεργάζονται ή αντικαθιστούν αρχεία που χρειάζονταν αναβάθμιση χωρίς να μας δίνουν δυνατότητα για περαιτέρω ρυθμίσεις. Στο τρίτο κεφάλαιο όπου γίνεται εκτενή αναφορά στα templates αναφέραμε και την ανάγκη για τις αναβαθμίσεις. Υπενθυμίζουμε πως με κάθε αναβάθμιση στο σύστημα, στα templates ή στα extensions οι αλλαγές που έχουμε κάνει στον πυρήνα τους θα χαθούν. Για περισσότερες πληροφορίες σχετικά με τους κινδύνους ανατρέξτε στο υποκεφάλαιο 3.3.3 με όνομα Child Themes.

Πέρα από τις αναβαθμίσεις σε αυτό το σημείο θα αναφέρουμε άλλα θέματα που μπορεί να προκύψουν σε μία εφαρμογή μετά την ολοκλήρωση της και θα εξετάσουμε τους τρόπους επίλυσης που μας προσφέρουν τα CMS.

➤ Επίθεση στο σύστημα

Εάν ποτέ παρατηρήσουμε ξαφνικές αλλαγές στα αρχεία της εφαρμογής, ή κάποια ασυνήθιστη δραστηριότητα στον Server μπορεί να σημαίνει πως κάποιος παραβίασε ή προσπαθεί να παραβιάσει το σύστημα. Είναι απαιτητό να έχουμε εγκατεστημένο κάποιο extension για επιπλέον ασφάλεια, το οποίο θα μας ενημερώσει με e-mail σε κάθε προσπάθεια παραβίασης ή ασυνήθιστων αιτημάτων. Υπάρχουν πολλοί διαφορετικοί τύποι επίθεσης στον Server οι οποίοι έχουν διαφορετικές συνέπειες στην εφαρμογή. Είναι διαφορετικό να γίνει επίθεση με ψεύτικη κίνηση στην οποία ο Server δεν μπορεί να ανταποκριθεί (όπως λέμε να “πέσει” η εφαρμογή) σε σχέση με το να αποκτήσει κάποιος πρόσβαση στην βάση δεδομένων της εφαρμογής.

Όταν υποπτευόμαστε πως είμαστε θύματα της δεύτερης περίπτωσης το Joomla, όπως και τα υπόλοιπα CMS, μας παρέχει ένα security forum με υποστήριξη προγραμματιστών του συστήματος ώστε να μας βοηθήσουν. Υποχρεωτικά όμως, πριν απευθυνθούμε σε αυτούς, υπάρχουν κάποιες κινήσεις που πρέπει να πραγματοποιήσουμε:

- Γυρνάμε την εφαρμογή σε offline Mode από το Control Panel.
- Ελέγχουμε όλες τις συσκευές που περιέχουν αρχεία FTP (File Transfer Protocol) και Joomla Admin Access με κάποιο πρόγραμμα που εντοπίζει κακόβουλα λογισμικά.
- Εξασφαλίζουμε ότι διαθέτουμε την τελευταία έκδοση του συστήματος καθώς παρέχει μεγαλύτερη προστασία σε θέματα ασφαλείας.
- Εάν ο υπολογιστής που λειτουργεί ως Server της εφαρμογής δεν μας ανήκει, ειδοποιούμε τον διαχειριστή να πραγματοποιήσει και ο ίδιος έναν καθαρισμό, καθώς και να ελέγξει για πιθανές παραβιάσεις.
- Αλλάζουμε παντού τους κωδικούς και (εάν είναι δυνατόν) τα ονόματα σε κάθε σημείο πρόσβασης σε ευαίσθητα δεδομένα. (Server, Domain Name, Βάση Δεδομένων κτλ).
- Εάν ήμασταν ο μόνος administrator της εφαρμογής δημιουργούμε νέο λογαριασμό και διαγράφουμε τον παλιό.
- Τα αρχεία πολυμέσων είναι ευάλωτα και μπορεί να αντικατασταθούν εύκολα με νέα στα οποία κρύβεται κακόβουλο λογισμικό. Τα ελέγχουμε ή αν είναι δυνατόν τα αντικαθιστούμε.
- Ελέγχουμε τα logs του Server για αιτήματα σε φακέλους του συστήματος ή για POST Commands.
- Ελέγχουμε τα δικαιώματα σε φακέλους και αρχεία για Linux Server. Δεν πρέπει να επιτρέπεται ποτέ η επεξεργασία στην τελευταία κατηγορία (others), ειδικά σε media φακέλους όπως είπαμε νωρίτερα.

Για να κατανοήσουμε πλήρως την τελευταία κίνηση πρέπει να γνωρίζουμε τις βασικές ρυθμίσεις προσπέλασης αρχείων η οποία πλέον για μεγαλύτερη ευελιξία δηλώνεται με αριθμούς. Υπάρχουν τρεις κατηγορίες που ορίζουμε δικαιώματα: **Admin**, **Users** και **Others**. Τα δικαιώματα που μπορεί να έχουν είναι πάλι τρία: **Read**, **Write** και **Execute**. Τα νούμερα που δίνουμε καθορίζουν ποια από τα δικαιώματα θα έχουν οι κατηγορίες με την σειρά που τις αναφέραμε. Τα νούμερα αντικατοπτρίζουν τα δικαιώματα όπως παρουσιάζεται στον παρακάτω λίστα:

- 0 – Κανένα δικαίωμα
- 1 – Execute
- 2 – Write
- 3 – Write & Execute
- 4 – Read
- 5 – Read & Execute
- 6 – Read & Write
- 7 – Read, Write, & Execute

Αρα θέτοντας δικαιώματα 000 δεν επιτρέπεται καμία ενέργεια και σε κανέναν, ενώ με δικαιώματα 777 ακόμα και οι επισκέπτες της εφαρμογής μπορούν να επεξεργαστούν και να εκτελέσουν αρχεία. Το Joomla προτείνει στον διαχειριστή να θέσει τα δικαιώματα για τα αρχεία στο 644 και για τους φακέλους στο 755.

➤ Διαχείριση χρηστών

Μέχρι την έκδοση 3.4 του συστήματος Joomla, σε κάθε νέα εγκατάσταση από προεπιλογή υπήρχε στην αρχική σελίδα μια φόρμα εγγραφής χρηστών. Εάν δεν την χρειαζόμασταν έπρεπε να βρούμε το module και να το απενεργοποιήσουμε. Πλέον, ο μινιμαλισμός και η απλότητα που κυριαρχούν στον σχεδιασμό των εφαρμογών, ανάγκασαν τους δημιουργούς του συστήματος να κρύψουν την φόρμα, αλλά όχι και να την αφαιρέσουν εντελώς. Σε κάθε νέα εγκατάσταση υπάρχει ένας διακριτικός σύνδεσμος με το όνομα login ή Sign up όπου σε οδηγεί στην φόρμα εγγραφής.

Γενικότερα ο έλεγχος των χρηστών θεωρείται από τα δυνατά σημεία του Joomla συγκριτικά με τα υπόλοιπα open source CMS. Μας δίνει την δυνατότητα να αποκρύπτουμε σελίδες και components από απλούς επισκέπτες της εφαρμογής, αλλά ακόμα και από χρήστες περιορισμένων δικαιωμάτων. Δεν είναι εύκολη δουλειά το να διαχειρίζεσαι μεγάλο όγκο χρηστών, αλλά το μενού του Joomla απλοποιεί κατά πολύ την διαδικασία ομαδοποιώντας τους χρήστες ανάλογα το είδος τους.

Με την εφαρμογή μας ενεργή στο διαδίκτυο είναι σίγουρο πως κάποια στιγμή θα προκύψει θέμα με κάποιον χρήστη. Η διαχείριση των χρηστών πραγματοποιείται στην διαδρομή Users / User Manager από το οριζόντιο μενού στο πάνω μέρος του συστήματος. Εκεί παρουσιάζεται λίστα με όλους τους χρήστες που είναι εγγεγραμμένοι και φυσικά υπάρχουν διάφορα search tools για την εύκολη εύρεση τους. Επίσης στα αριστερά βλέπουμε ένα νέο μενού το οποίο περιέχει κατηγορίες με ονόματα όπως **Users Group** και **Access Levels**.

Το Joomla χρησιμοποιεί αρχικά κάποια επίπεδα για να ομαδοποιεί τους εγγεγραμμένους χρήστες, τα οποία χαρακτηρίζονται με διαφορετικά δικαιώματα πρόσβασης. Παράλληλα με αυτά υπάρχουν κάποιες κατηγορίες που έχουν τα δικαιώματα κάποιου επιπέδου, μαζί με δικά τους χαρακτηριστικά. Οι κατηγορίες είναι πιθανό να σχετίζονται με παραπάνω από ένα επίπεδα, ειδικότερα οι κατηγορίες με τα περισσότερα δικαιώματα. Τα επίπεδα είναι 5, όπως παρουσιάζονται παρακάτω.

- Public
- Guest
- Registered
- Special
- Super Users

Είναι προφανές και από τα ονόματα πως όσο κατεβαίνουμε στην λίστα, τόσα περισσότερα δικαιώματα έχει ο εγγεγραμμένος χρήστης. Το πρώτο επίπεδο, φυσικά, δεν σχετίζεται με κάποια κατηγορία, αφού αναφέρεται στους επισκέπτες της σελίδας. Αυτό σημαίνει πως μπορούμε έναν χρήστη να τον υποβαθμίσουμε εντελώς απλά τοποθετώντας

τον στο επίπεδο Public. Οι κατηγορίες που προσφέρει το Joomla λέγονται Groups και χωρίζονται σε δύο κομμάτια. Τα **FronD End Groups**, που περιέχουν τους χρήστες που έχουν δικαιώματα στα περιεχόμενα που βλέπει και αλληλεπιδρά ο επισκέπτης και τα **Back End Groups**, στα οποία οι χρήστες ασχολούνται με πράγματα που δεν αφορούν τους επισκέπτες, όπως ο κώδικας και το σύστημα της εφαρμογής.

FronD End Groups	Δικαιώματα
Guest	Βλέπει τις Public σελίδες και άρθρα Ελάχιστα επιπλέον δικαιώματα όπως σχολιασμοί
Registered	Όλα τα Δικαιώματα της κατηγορίας Guest Σελίδες μόνο για εγγεγραμμένα μέλη
Author	Όλα τα δικαιώματα της κατηγορίας Registered Δημιουργία Σελίδων Επεξεργασία Σελίδων Προβολή κρυφών περιεχομένων
Editor	Όλα τα δικαιώματα της κατηγορίας Author Επεξεργασία όλων των σελίδων, ακόμα και των unpublished
Publisher	Όλα τα δικαιώματα της κατηγορίας Editor Δημοσίευση σελίδων

Πίνακας 5.2-1 FronD End Groups

Back End Groups	Δικαιώματα
Manager	Διαμοιρασμός δικαιωμάτων σε χρήστες Πρόσβαση στο περιβάλλον του συστήματος
Administrator	Όλα τα Δικαιώματα της κατηγορίας Manager Δημιουργία νέων χρηστών Εγκατάσταση extension
Super User	Όλα τα Δικαιώματα της κατηγορίας Administrator Αλλαγές στα templates Αλλαγές στα Global Configuration

Πίνακας 5.2-2 Back End Groups

Παρατηρούμε πως συνολικά οι κατηγορίες είναι οκτώ, με τις πέντε από αυτές να ανήκουν στο Front End Group. Οι αλλαγές στα δικαιώματα, τις οποίες όπως φαίνεται μπορούν να τις πραγματοποιήσουν μόνο τα Back End Groups, πραγματοποιούνται πολύ απλά πατώντας στο όνομα του χρήστη από την λίστα των Users και έπειτα πηγαίνοντας στο tab Assigned User Groups. Έχουμε και την δυνατότητα απενεργοποίησης ενός χρήστη, πολύ απλά από την στήλη Activated στην αρχική λίστα ονομάτων.

➤ Αναφορά προβλήματος / bug

Είναι πιθανό για προγραμματιστές που χρησιμοποιούν τακτικά τα CMS, να συναντήσουν κάποια ανωμαλία ή δυσλειτουργία στο σύστημα. Όπως είδαμε και στο 2^ο κεφάλαιο η διαδικασία της εγκατάστασης είναι περίπλοκη και κρύβει κινδύνους οι οποίοι επηρεάζουν το τελικό αποτέλεσμα. Εδώ θα αναλύσουμε τις ενέργειες που πρέπει να κάνει ο προγραμματιστής όταν, χωρίς δική του παρέμβαση στον πυρήνα του συστήματος, το περιβάλλον του Joomla δεν ανταποκρίνεται. Το πρόβλημα μπορεί να είναι από σοβαρό, όπως το να μην παράγονται νέες σελίδες, έως και κάτι πιο απλό, όπως να μην εμφανίζεται η σωστή ημερομηνία στα άρθρα.

Η πρώτη κίνηση είναι αμέσως να επισκεφτούμε το Joomla Issue Tracker, στο issues.joomla.org. Εκεί βρίσκεται ένα forum με συζητήσεις χρηστών του συστήματος, πάνω σε bugs της εφαρμογής, με απαντήσεις από τεχνικούς του συστήματος και από έμπειρους χρήστες. Κάνουμε μία αναζήτηση στους τίτλους των συζητήσεων ψάχνοντας με τα search tools κάποια συζήτηση που να σχετίζεται με το πρόβλημά μας. Προσέχουμε να ψάχνουμε στις συζητήσεις της κατηγορίας closed, καθώς σε αυτές έχει βρεθεί λύση. Μόνο όταν βεβαιωθούμε πως το πρόβλημα που έχουμε δεν υπάρχει στις συζητήσεις δημιουργούμε νέο αντικείμενο συζήτησης.

Για την καλύτερη εξυπηρέτηση των προγραμματιστών, στο forum η δημιουργία αντικειμένου συζήτησης γίνεται υπό συγκεκριμένες προϋποθέσεις. Καταρχάς πρέπει ο δημιουργός της συζήτησης καθώς και όσοι συμμετέχουν να διαθέτουν λογαριασμό GitHub ώστε να επιβεβαιώνονται από εκεί τα στοιχεία τους. Το GitHub είναι μια παγκόσμια πλατφόρμα που λειτουργεί ως χώρος δημιουργίας και διαμοιρασμού open source κατασκευών και επιτρέπει την ομαδική συνεργασία απομακρυσμένων προγραμματιστών σε ένα project.

Η δημιουργία συζήτησης ξεκινά πατώντας το κουμπί New Item, με το οποίο εμφανίζεται ένα νέο παράθυρο. Εκεί συμπληρώνουμε όσο καλύτερα μπορούμε τις κατηγορίες, όπως παρουσιάζονται παρακάτω:

- **Priority:** Η προεπιλογή και η συνήθης επιλογή στην προτεραιότητα είναι η επιλογή Medium (Μεσαία)
- **Build:** Λεπτομέρειες του συστήματος όπως έκδοση
- **Categories:** Δίνουμε τις κατηγορίες που επηρεάζει το πρόβλημα που έχουμε, όπως της είδαμε νωρίτερα
- **Title:** Ο τίτλος χρησιμεύει σαν μια σύντομη περιγραφή του προβλήματος
- **Description:** Δίνουμε λεπτομέρειες του προβλήματος. Ακολουθεί αναλυτική περιγραφή για την σωστή διατύπωση αυτής της κατηγορίας

- **Uploads:** Επιτρέπονται εικόνες για την καλύτερη περιγραφή του προβλήματος

Η κατηγορία που αφορά περισσότερο τους τεχνικούς του Joomla είναι η περιγραφή του προβλήματος από τον χρήστη. Το μεγαλύτερο ποσοστό των προβλημάτων στο forum προκλήθηκε από απροσεξία των χρηστών, και έτσι ο μόνος τρόπος να εντοπιστεί το πρόβλημα από τον τεχνικό μέσω των συζητήσεων είναι μια αναλυτική περιγραφή. Οι τεχνικοί έχουν καταλήξει πως η καλύτερη διατύπωση των προβλημάτων χωρίζεται σε τέσσερα μέρη ως εξής:

1. Περιγραφή των κινήσεων για το πως ο χρήστης ανακάλυψε το πρόβλημα
2. Περιγραφή των προβληματικών αποτελεσμάτων
3. Περιγραφή των αποτελεσμάτων που θα περίμενε ο χρήστης
4. Λοιπές πληροφορίες, πιθανολογίες του χρήστη για την αιτιολογία του προβλήματος και προτάσεις του ίδιου για την επίλυση

Το Joomla Issue Tracker forum είναι διαφορετικό από το Joomla Security forum. Προβλήματα που αφορούν την ασφάλεια του συστήματος τα δηλώνουμε στο δεύτερο forum. Γενικά όμως στο Joomla Issue Tracker πρέπει να είμαστε ιδιαίτερα προσεχτικοί με τα προσωπικά μας στοιχεία καθώς όλες οι συζητήσεις είναι ανοιχτές ακόμα και σε απλούς επισκέπτες του forum.

5.2.2 Παρακολούθηση εφαρμογής

Τεράστιο ρόλο στην σωστή διαχείριση της εφαρμογής παίζει η συνεχής παρακολούθηση των συνηθειών που έχουν οι επισκέπτες της σελίδας μας. Πιο συγκεκριμένα, ελέγχοντας τις κινήσεις τους στην σελίδα, την τοποθεσία τους, το λειτουργικό τους, τον browser που μας επισκέπτονται, μπορούμε να διαμορφώσουμε την εφαρμογή μας ώστε να καλύπτει συγκεκριμένες ανάγκες αποκλειστικά για τις συσκευές των σταθερών επισκεπτών.

Για παράδειγμα σύμφωνα με τις μετρήσεις για την εφαρμογή cafepatras.gr, την πρώτη εβδομάδα λειτουργίας της σελίδας, οι φορητές συσκευές που την επισκέφθηκαν περισσότερες φορές φαίνονται στην παρακάτω εικόνα.

Παρατηρούμε πως οι φορητές συσκευές της Apple ξεπερνάν το 15% των συσκευών που επισκέφθηκαν την εφαρμογή μας. Έπειτα η λίστα περιέχει μόνο συσκευές που διαθέτουν λειτουργικό σύστημα Android. Άρα όταν στην εφαρμογή μου προσθέσω animation σε CSS κώδικα ή κομμάτια Javascript που έχω την υποψία πως μπορεί να δυσλειτουργούν σε κάποιους browser, κάνω αναλυτικό έλεγχο της λειτουργίας τους σε Safari, Chrome, και Android WebView ξέροντας πως ελάχιστοι θα μπουν από λειτουργικό Windows Phone ή Blackberry.

1.	Apple iPhone	13,92%
2.	Huawei ALE-L21 P8 Lite	12,66%
3.	Samsung SM-J500FN Galaxy J5	10,13%
4.	Asus ME173X Memo Pad HD7	5,06%
5.	Samsung SM-G935F Galaxy S7 Edge	5,06%
6.	Sony D2303 Xperia M2	5,06%
7.	eSTAR Beauty HD Quad Core	3,80%
8.	Apple iPhone 6s	2,53%
9.	Asus ME172V MeMO Pad ME172V	2,53%
10.	Samsung SM-G110H Galaxy Pocket 2 Duos	2,53%

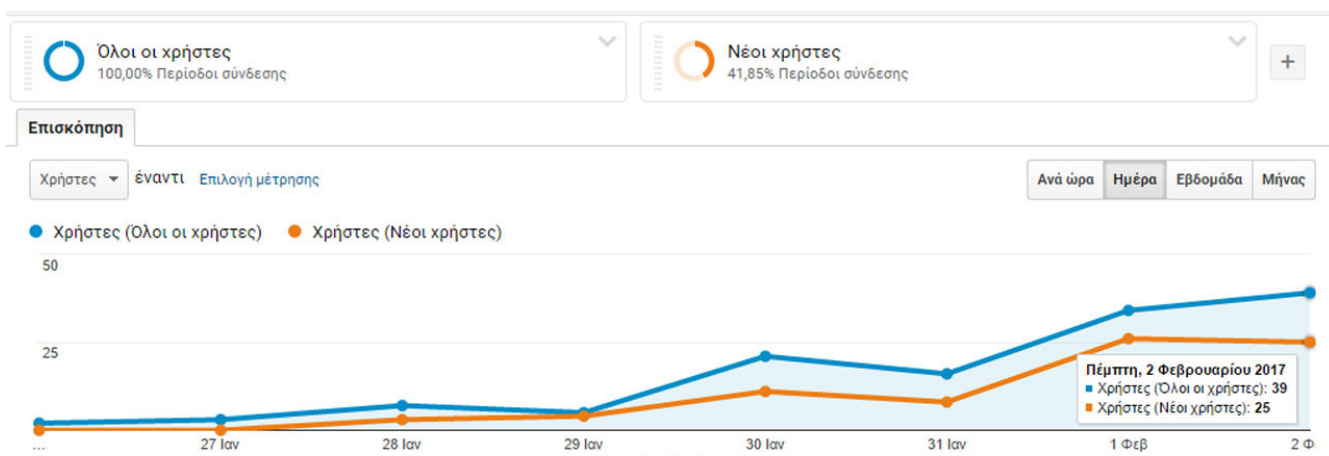
Εικόνα 5.2-1 Ποσοστά φορητών συσκευών που επισκέπτονται την εφαρμογή

➤ Google Analytics

Την παραπάνω λίστα όπως και άλλα ποσοστά που αφορούν την εφαρμογή μας τα βλέπουμε μέσω των Google Analytics όπως αναφέραμε και στο κεφάλαιο με τα extensions της εφαρμογής. Υπάρχουν πληροφορίες για την επισκεψιμότητα της εφαρμογής και στο Control Panel του Joomla, αλλά δεν τα παραθέτουμε εδώ καθώς περιέχουν λίγες πραγματικά χρήσιμες πληροφορίες για την εξέλιξη της εφαρμογής μας.

Η εφαρμογή caferpatras.gr μπήκε σε online mode και έγινε διαθέσιμη στους χρήστες του διαδικτύου στις 27 Ιανουαρίου του 2017. Από τότε έγινε μια σχετική προώθηση μέσω των μηχανών αναζήτησης και των μέσων κοινωνικής δικτύωσης ώστε να την βρουν άτομα που ενδιαφέρονται για το περιεχόμενο. Για να μπορούμε να εξελίξουμε την εφαρμογή πρέπει να παρατηρήσουμε τις συνήθειες των χρηστών. Όταν για παράδειγμα μία σελίδα είναι εξαιρετικά δημοφιλής αυτό σημαίνει πως, με γνώμονα εκείνη, πρέπει να τροποποιήσουμε τις υπόλοιπες σελίδες. Αφήσαμε ενεργή την εφαρμογή και ελέγχουμε τα δεδομένα της μια εβδομάδα μετά από την ενεργοποίηση όπως παρουσιάζονται μέσα στα διαγράμματα του Google Analytics.

Αρχικά ας δούμε τις μετρήσεις χρηστών της εφαρμογής την πρώτη εβδομάδα ώστε να υπολογίσουμε τον αριθμό επισκεπτών που αναμένεται να έχει η σελίδα μας. Στο παρακάτω διάγραμμα ο κάθετος άξονας ορίζει το νούμερο των επισκεπτών, ο οριζόντιος την πάροδο των ημερών, η μπλε γραμμή δηλώνει τους μοναδικούς επισκέπτες της σελίδας κάθε μέρα, ενώ η πορτοκαλή γραμμή δηλώνει τους χρήστες που επισκέφθηκαν για πρώτη φορά την εφαρμογή.



Εικόνα 5.2-2 Επισκέψεις μοναδικών χρηστών στην εφαρμογή ανά ημέρα

Γενικότερα το συγκεκριμένο διάγραμμα μας βοηθάει όχι τόσο στο να εξελίξουμε την εφαρμογή μας, αλλά περισσότερο στον να διαπιστώσουμε την αξία της στην αγορά. Μία εφαρμογή με μεγάλο μέσο όρο καθημερινών επισκέψεων και κυρίως από χρήστες που επιστρέφουν και παραμένουν πολλά λεπτά προσφέρεται για διαφήμιση προϊόντων. Μια εφαρμογή με μεγάλα ποσοστά επισκεψιμότητας προσελκύει και το ενδιαφέρον της ίδιας της Google η οποία με χρηματικά ανταλλάγματα που ορίζονται από το Google Analytics θα προβάλλει στοχευόμενες διαφημίσεις στην εφαρμογή μας. Υπάρχουν πολλά τέτοια διαγράμματα ώστε να υπολογίσουμε την διαφημιστική της αξία όπως παρουσιάζονται παρακάτω:



Εικόνα 5.2-3 Διαγράμματα επίσκεψης χρηστών στην εφαρμογή

Μεγάλο ρόλο για την διαφημιστική αξία και κυρίως για την εξέλιξη της εφαρμογής παίζει και η τοποθεσία των επισκεπτών.

Χώρα ?	Απόκτηση			Συμπεριφορά			Μετατροπές		
	Περίοδοι σύνδεσης ? ↓	% Νέες περιόδοι σύνδεσης ?	Νέοι χρήστες ?	Ποσοστό εγκατάλειψης ?	Σελίδες / περίοδο σύνδεσης ?	Μέση διάρκεια περιόδου σύνδεσης ?	Ποσοστό μετατροπής στόχου ?	Ολοκληρώσεις στόχου ?	Τιμή στόχου ?
	184 % του συνόλου: 100,00% (184)	41,85% Μέσος όρος για προβολή: 41,85% (0,00%)	77 % του συνόλου: 100,00% (77)	30,98% Μέσος όρος για προβολή: 30,98% (0,00%)	7,23 Μέσος όρος για προβολή: 7,23 (0,00%)	00:06:35 Μέσος όρος για προβολή: 00:06:35 (0,00%)	0,00% Μέσος όρος για προβολή: 0,00% (0,00%)	0 % του συνόλου: 0,00% (0)	0,00 \$ % του συνόλου: 0,00% (0,00 \$)
1. Greece	173 (94,02%)	38,73%	67 (87,01%)	29,48%	7,53	00:06:45	0,00%	0 (0,00%)	0,00 \$ (0,00%)
2. France	4 (2,17%)	75,00%	3 (3,90%)	75,00%	1,25	00:03:49	0,00%	0 (0,00%)	0,00 \$ (0,00%)
3. United States	4 (2,17%)	100,00%	4 (5,19%)	25,00%	3,75	00:06:22	0,00%	0 (0,00%)	0,00 \$ (0,00%)
4. Netherlands	2 (1,09%)	100,00%	2 (2,60%)	100,00%	1,00	00:00:00	0,00%	0 (0,00%)	0,00 \$ (0,00%)
5. Cyprus	1 (0,54%)	100,00%	1 (1,30%)	0,00%	7,00	00:04:28	0,00%	0 (0,00%)	0,00 \$ (0,00%)

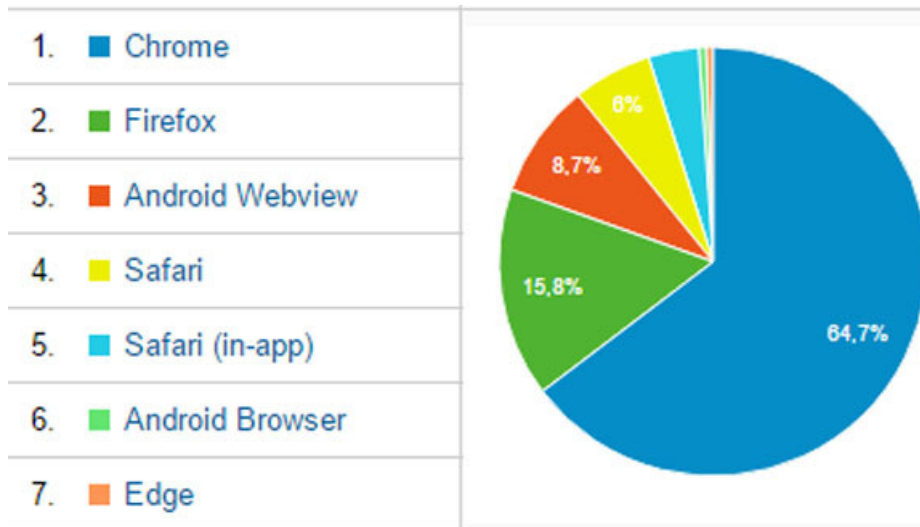
Εικόνα 5.2-4 Πίνακας προέλευσης επισκεπτών στην εφαρμογή

Ο πάνω πίνακας μας εμφανίζει τις χώρες από τις οποίες έχουν πραγματοποιηθεί αιτήματα στον Server. Για μεγαλύτερη ακρίβεια και έλεγχο στην εφαρμογή, υπάρχουν διαθέσιμες ακόμα και οι πόλεις όπου έγιναν τα αιτήματα.

Μέσα από τους πίνακες και τα διαγράμματα του Google Analytics μπορούμε να μάθουμε λεπτομέρειες όπως την IP διεύθυνση των επισκεπτών, την ανάλυση οθόνης που χρησιμοποιούν και τον τρόπο προέλευσης τους στην εφαρμογή. Το τελευταίο στοιχείο είναι μεγάλης σημασίας ειδικά όταν έχουμε διαφήμιση επί πληρωμή σε κάποια άλλη διαδικτυακή εφαρμογή και θέλουμε να καταγράψουμε το ποσοστό κίνησης που μας προσφέρει.

Επίσης εάν διαχειριζόμαστε e-shop το Google Analytics είναι το κατάλληλο εργαλείο ώστε να διαχειριζόμαστε τους πελάτες μας. Προσφέρει διαγράμματα με τα ποσοστά πωλήσεων, τους κορυφαίους χρήστες με τις περισσότερες αγορές και τα συνολικά κέρδη της εφαρμογής σε αντιδιαστολή με τα έξοδα. Επίσης παρέχει την δυνατότητα να ορίζουμε στόχους με οποιαδήποτε κριτήρια επιθυμούμε και για οποιοδήποτε χρονικό διάστημα, ώστε να ελέγχουμε αποκλειστικά τον τομέα που μας απασχολεί περισσότερο στην πορεία της εφαρμογής.

Για να παραμείνουμε όμως στον προγραμματιστικό τομέα, ένα διάγραμμα που αφορά πάντα τους δημιουργούς εφαρμογών είναι ο browser που χρησιμοποιούν οι επισκέπτες. Εδώ παρουσιάζονται οι προτιμήσεις των χρηστών της εφαρμογής safepatras.gr:



Εικόνα 5.2-5 Προτιμήσεις browser των επισκεπτών της εφαρμογής

Τελευταίο παράδειγμα της χρησιμότητας των Google Analytics στην εφαρμογή μας είναι τα διαγράμματα επισκεψιμότητας της εφαρμογής ανά ώρα και η παρακολούθηση της σε πραγματικό χρόνο. Αυτά τα δύο είναι πολύτιμα όταν επιθυμούμε να κάνουμε κάποια μεγάλη αλλαγή στην εφαρμογή. Ελέγχουμε πρώτα ποιες ώρες δεν υπάρχει μεγάλη κίνηση στην εφαρμογή, και έπειτα εκείνη την ώρα παρακολουθούμε την ζωντανή ροή επισκεπτών στην σελίδα. Όταν δούμε ότι δεν υπάρχει επισκέπτης ή έστω ότι υπάρχει ο μικρότερος δυνατός αριθμός, πραγματοποιούμε την αλλαγή.

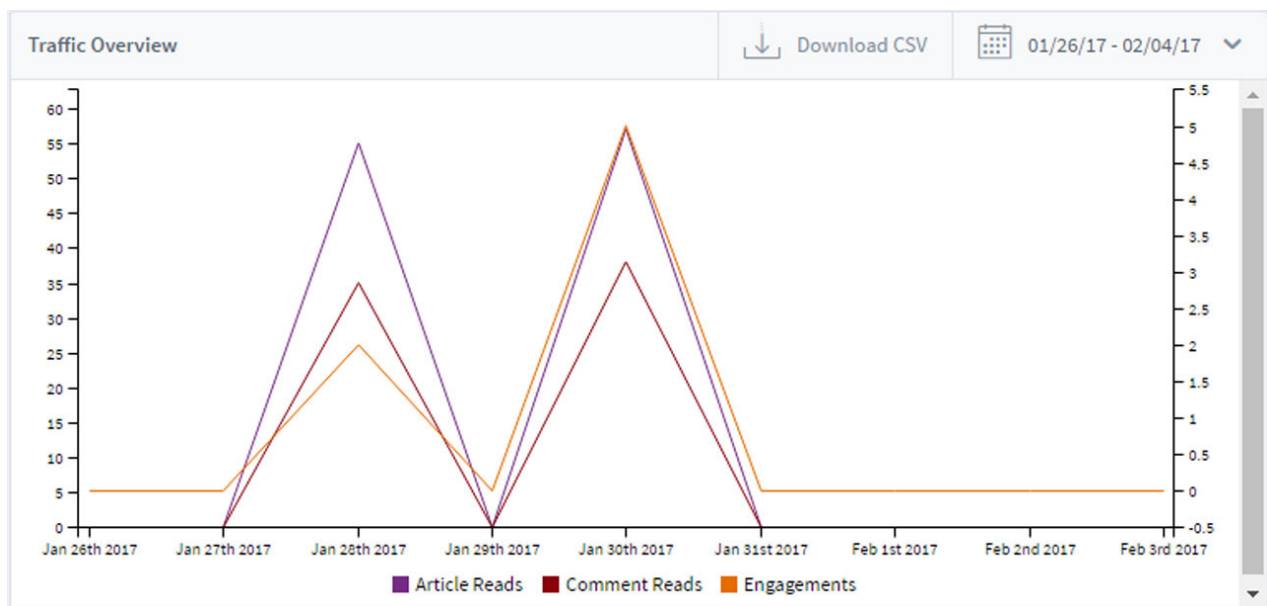
➤ Disqus

Παρόμοιο έλεγχο χρειαζόμαστε και για τα σχόλια, καθώς δεν εκφέρουν τις δικές μας απόψεις, αλλά εν' μέρη είμαστε υπεύθυνοι για το περιεχόμενο τους μέσα στην εφαρμογή μας. Το extension Disqus μας παρέχει την δυνατότητα να διαχειριζόμαστε απόλυτα το τι θα εμφανίζεται. Αρχικά μπορούμε να ορίσουμε όλα τα σχόλια να χρειάζονται έγκριση από τον διαχειριστή ώστε να δημοσιευτούν. Αν είναι μεγάλος ο όγκος μπορούμε επίσης να απαγορεύσουμε τον σχολιασμό σε απλούς επισκέπτες ή έστω να ορίσουμε την έγκριση μόνο για τα δικά τους σχόλια. Αλλά και από την στιγμή που θα ανέβει ένα σχόλιο στην εφαρμογή μας έχουμε την δυνατότητα να το διαγράψουμε, να το μαρκάρουμε ως spam ή ακόμα για ιδιαίτερες περιπτώσεις μπορούμε να το επεξεργαστούμε ώστε να διαγράψουμε συγκεκριμένες φράσεις. Προφανώς στην τελευταία περίπτωση θα εμφανίζεται σημείωση ότι το σχόλιο δέχτηκε επεξεργασία.

Τα σχόλια της εφαρμογής παρουσιάζονται σε λίστα και χωρίζονται σε τέσσερις κατηγορίες. Εκείνα που **αναμένουν έγκριση**, τα **δημοσιευμένα**, τα **spam** και τα **διαγραμμένα**. Κάθε σχόλιο περιέχει τον σύνδεσμο της σελίδας όπου βρίσκεται μαζί με το όνομα, το e-mail και την IP του χρήστη, ακόμα και αν σχολίασε ως guest. Έτσι είναι πολύ εύκολο να ξεχωρίσουμε προβληματικές συμπεριφορές σε χρήστες, όταν παρουσιάζεται πολλές φορές το όνομα τους στις δύο τελευταίες κατηγορίες. Οι ενέργειες

που μπορούμε να πραγματοποιήσουμε σε τέτοιες περιπτώσεις είναι αρχικά η προειδοποίηση και έπειτα ο αποκλεισμός του χρήστη.

Επιπλέον το Disqus μας παρέχει ανάλογους πίνακες και διαγράμματα οι οποίοι αφορούν αποκλειστικά τον σχολιασμό στην εφαρμογή μας. Πιο συγκεκριμένα μπορούμε να δούμε πόσα σχόλια υπάρχουν ανά ημέρα στην εφαρμογή, πόσα up-votes και down-votes έλαβαν, ενώ ακόμα μπορούμε να μάθουμε τον αριθμό επισκεπτών που φτάνει μέχρι το κάτω σημείο της σελίδας όπου υπάρχουν οι σχολιασμοί. Όλα τα στοιχεία με επιπλέον λεπτομέρειες είναι διαθέσιμα και σε αρχείο κειμένου το οποίο μπορούμε να κατεβάσουμε από το περιβάλλον του extension.



Εικόνα 5.2-6 Διαγράμματα προβολής σχολίων συγκριτικά με τις επισκέψεις

Στο παραπάνω διάγραμμα, με το εύρος ημερών όπως και νωρίτερα είναι η εβδομάδα 27/1/2017 με 2/2/2017, παρατηρούμε τρεις γραμμές. Την πορτοκαλί που είναι η μοναδική που αντιστοιχεί στον δεξί κάθετο άξονα και αφορά τον αριθμό σχολίων που πραγματοποιήθηκαν στις 28/1 και 30/1 (2 και 5 σχόλια αντίστοιχα), την μπλε γραμμή που είναι οι προβολές σελίδων για εκείνες τις ημερομηνίες και την κόκκινη που παρουσιάζει τον αριθμό των επισκεπτών που κατέβηκε μέχρι το κάτω μέρος της σελίδας όπου υπάρχουν τα σχόλια. Παρατηρούμε πως ο μέσος όρος επισκεπτών στα άρθρα είναι 56, αλλά αυτοί που ενδιαφέρονται για τα σχόλια κατά μέσο όρο μόλις 36.

Η σταδιακή αύξηση του δεύτερου αριθμού είναι κάτι που θα μας απασχολήσει στο μέλλον. Τα δεδομένα στον χώρο του web development, λόγω του τεράστιου ανταγωνισμού και της προόδου της τεχνολογίας, αλλάζουν συνεχώς και παράλληλα οι απαιτήσεις των πελατών αυξάνονται. Πλέον ο χρήστης αν δεν βρει εύκολα και γρήγορα την πληροφορία που ψάχνει απογοητεύεται και εγκαταλείπει την σελίδα. Άλλωστε αυτός είναι και ο σκοπός των διαγραμμάτων που μελετήσαμε. Να μας βοηθήσουν να εξελίξουμε την εφαρμογή μας σε όλους τους τομείς προσφέροντας σε κάθε επισκέπτη μια ολοκληρωμένη εμπειρία πλοήγησης.

Μελλοντικοί στόχοι

Για το πρακτικό μέρος της πτυχιακής μου άσκησης, επιλέξαμε μαζί με τον επιβλέποντα καθηγητή, ένα ενδιαφέρον θέμα το οποίο έχει χρηστική αξία για επισκέπτες και κατοίκους της Πάτρας, άρα μπορεί να συνεχίσει να δέχεται χρήστες. Πρώτος και κυριότερος στόχος μου είναι να συντηρήσω την εφαρμογή και να διορθώσω τα όποια προβλήματα προκύψουν, καθώς λειτουργεί και ως καλή διαφήμιση της δουλείας μου στους επισκέπτες. Έπειτα υπάρχουν κάποια σημεία τα οποία αν εξελιχθούν θα κάνουν μεγάλη διαφορά στην εμπειρία επίσκεψης της εφαρμογής. Προφανώς όμως, οι μελλοντικοί στόχοι καθώς και η γενικότερη εξέλιξη της εφαρμογής εξαρτώνται αποκλειστικά από την κίνηση που θα έχει η εφαρμογή.

Ο κυριότερος μελλοντικός στόχος είναι να βγαίνει ένας μέσος όρος από τις βαθμολογίες των χρηστών για κάθε καφετέρια, ο οποίος θα παρουσιάζεται στο λογότυπο της στην αρχική σελίδα. Έτσι οι επισκέπτες που δεν γνωρίζουν της καφετέριες της Πάτρας θα μπορούν εύκολα να δουν τις δημοφιλέστερες μέσα στην εφαρμογή. Το μόνο ζήτημα εδώ είναι να ταιριάζει αυτή η αλλαγή στην δομή της αρχικής σελίδας και να μην ταραξεί την αισθητική της ισορροπία. Άλλο ένα σημείο που επιδέχεται βελτίωσης είναι οι παροχές που επιτρέπουν την εύκολη και στοχευμένη αναζήτηση. Μελετώντας τις αναζητήσεις των χρηστών και τις κινήσεις της αγοράς θα μπορούσαμε να προσθέσουμε επιπλέον παροχές ώστε ο επισκέπτης να βρίσκει πάντα αυτό που ψάχνει. Τέλος ξανά πάνω στον τομέα της αναζήτησης, θα πρέπει κάποτε να διαθέσω χρόνο στην προσθήκη επιπλέον tags στις περιγραφές των σελίδων, ώστε να δώσω ακόμη περισσότερους τρόπους στον χρήστη για την εύρεση της ιδανικής καφετέριας.

Επίλογος

Βάση λογικής, οι αναγνώστες της πτυχιακής άσκησης που μελέτησαν όλα τα κεφάλαια μέχρι και τον επίλογο χωρίζονται σε δύο κατηγορίες. Αυτούς που έχουν κάποιες εμπειρίες προγραμματισμού και κατανόησαν απόλυτα την λογική στις οδηγίες κατασκευής και σε αυτούς που προσπάθησαν να ακολουθήσουν αλλά χάθηκαν σε κάποια βήματα ή κάποια σημεία αφήγησης τους φάνηκαν θολά. Η μοναδική διαφορά ανάμεσα σε αυτές τις δύο κατηγορίες είναι η χαώδης απόσταση που χωρίζει την θεωρία από την πράξη στον τομέα της τεχνολογίας. Προσωπικά σαν φοιτητής του τμήματος Μηχανικών Πληροφορικής Τ.Ε. είχα την τύχη να διδαχθώ τις αρχές του προγραμματισμού από έμπειρους και ικανούς καθηγητές. Όλοι τους όταν συγκεντρωνόμασταν στα εργαστήρια του τμήματος, υποστήριζαν πως η θεωρία στον προγραμματισμό είναι πολύ χρήσιμη, αλλά η πραγματική μάθηση κρύβεται στο να πειραματιστείς και να τρέξεις ο ίδιος τον κώδικα του προγράμματος στον compiler.

Όταν κάποιος αποφασίσει να ασχοληθεί με τον προγραμματισμό, σε οποιαδήποτε γλώσσα και οποιαδήποτε πλατφόρμα, να θυμάται πως όσα βιβλία και να διαβάσει, όσες τεχνικές και να μελετήσει, τελικά τίποτα δεν μπορεί να συγκριθεί με την ικανοποίηση να βλέπεις ένα hello world στην οθόνη του υπολογιστή σου.

Βιβλιογραφία & Σύνδεσμοι

1. **Εισαγωγή στις PHP 6 & MySQL 5 / PHP 6 and MySQL 5 for Dynamic Web Sites** Larry Ullman || Κλειδάριθμος
2. **Συστήματα Βάσεων Δεδομένων / Database System Concepts (6th Edition)** Abraham Silberschatz, Henry F. Korth, S. Sudarshan || Γκιούρδας
3. **Java™ Προγραμματισμός / Java™ How to Program (8th Edition)** Paul Deitel, Harvey Deitel || Γκιούρδας
4. **The Beginner's Guide to Linux Command Line** Himanshu Arora || MakeTechEasier
5. **Joomla! CMS**
<https://downloads.joomla.org/>
6. **Joomla! Documentations**
https://docs.joomla.org/Main_Page
7. **Πληροφορίες για χαρακτηριστικά των CMS:**
Wordpress: <https://wordpress.com/>
Drupal: <https://www.drupal.org/>
Magento: <https://magento.com/>
Blogger: <https://www.blogger.com/>
8. **Στατιστικά Προλόγου**
<http://www.internetlivestats.com/internet-users/>
9. **Διαγράμματα σελίδων 14 & 15 / Content management systems market position report** https://w3techs.com/technologies/market/content_management
10. **Διάγραμμα σελίδας 16 / CMS usage in Greece**
<https://trends.builtwith.com/cms/country/Greece>
11. **Extensions σελίδας 55:**
ExtraVote: <https://extensions.joomla.org/extension/top-extravote/>
SigPlus: <https://extensions.joomla.org/extension/sigplus/>
Disqus: <https://extensions.joomla.org/extension/disqus-comments/>
POWr Survey: <https://extensions.joomla.org/extension/contacts-and-feedback/surveys/powr-survey/>
Google Analytics: <https://extensions.joomla.org/extension/asynchronous-google-analytics/>
AdminExile: <https://extensions.joomla.org/extension/adminexile/>
EFSEO: <https://extensions.joomla.org/extension/easy-frontend-seo/>
RedactorJS Editor:
<https://extensions.joomla.org/extensions/extension/edition/editors/redactorjs-editor/>
12. **Πίνακας σελίδας 88 / Access Control List (ACL) for Joomla**
https://docs.joomla.org/Changing_user_groups