



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Ιστότοπου Ερευνητικών Προγραμμάτων & Πτυχιακών
Εργασιών σε WordPress**

Στυλιανός Αδαμίδης

Εισηγητής: Ιωάννης Έλληνας, Καθηγητής

**ΑΘΗΝΑ
ΔΕΚΕΜΒΡΙΟΣ 2015**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Ιστότοπου Ερευνητικών Προγραμμάτων & Πτυχιακών Εργασιών σε WordPress

**Στυλιανός Αδαμίδης
Α.Μ. 37875**

Εισηγητής:

Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

**Ιωάννης Έλληνας
Αναστασία Βελώνη
Διλιντάς Γεώργιος**

Ημερομηνία εξέτασης 02 / 02 / 2016

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Στυλιανός Αδαμίδης του Κωνσταντίνου, με αριθμό μητρώου 37875 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά θα ήθελα να ευχαριστήσω των εισηγητή καθηγητή της πτυχιακής Ιωάννη Έλληνα, που μου έδωσε την δυνατότητα να ασχοληθώ με το συγκεκριμένο θέμα, αλλά και για τις γρήγορες και άμεσες ανταποκρίσεις του σε προβληματισμούς και απορίες μου κατά την διάρκειά της. Επίσης ένα μεγάλο ευχαριστώ στην οικογένειά μου και ειδικά στους γονείς μου για την μεγάλη υπομονή που έδειξαν μέχρι να ολοκληρωθεί αυτή η εργασία και για την υποστήριξή τους καθ' όλη την διάρκεια των σπουδών μου στο τμήμα Μηχανικών Η/Υ Συστημάτων.

ΣΥΝΟΨΗ

Η παρούσα εργασία έχει ως αντικείμενό της την δημιουργία ιστότοπου για το τμήμα Μηχανικών Η/Υ Συστημάτων, στον οποίο θα αναρτώνται και θα παρουσιάζονται τα ερευνητικά προγράμματα του τμήματος και οι πτυχιακές εργασίες των σπουδαστών του.

Η ανάπτυξη του έγινε πάνω στο Σύστημα Διαχείρισης Περιεχομένου WordPress και χρειάστηκε η εις βάθος μελέτη και εκμάθηση του, προκειμένου να κατανοήσουμε και να εκμεταλλευτούμε τις δυνατότητες που μας δίνει στην χρήση αλλά και διαχείριση μιας ιστοσελίδας. Η σχεδίαση του έγινε με κύριο γνώμονα την λειτουργικότητα για τον απλό χρήστη αλλά και την ευκολία διαχείρισης του περιεχομένου του από τον διαχειριστή.

Η ιστοσελίδα παρέχει στους χρήστες αναλυτικές πληροφορίες σχετικά με την ερευνητική δραστηριότητα του τμήματος, τις ολοκληρωμένες πτυχιακές εργασίες που έχουν αναρτηθεί, καθώς και τα διαθέσιμα προς εκπόνηση θέματα. Στον διαχειριστή παρέχει εύκολο και γρήγορο τρόπο προσθήκης και επεξεργασίας υλικού και ανάρτησης σχετικών με αυτό ανακοινώσεις.

ABSTRACT

The present dissertation deals with the creation of a website for the dept. of Computer Systems Engineering, where the department's research programs and the students' dissertation papers will be presented.

The site was developed with the WordPress Content Management System and a deep study of its inner workings and functions was required in order to understand and use the wide range of tools and abilities it provides.

The website presents its visitors with information regarding the department's research activity, dissertation papers written by its students, as well as any available dissertation topics. To the administrator it provides an easy and fast way to add and edit content and related announcements.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Υπηρεσίες Διαδικτύου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: διαδίκτυο, internet, ιστοσελίδα, website, cms, wordpress

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	13
1.1 ΔΟΜΗ ΕΡΓΑΣΙΑΣ	13
1.2 ΠΕΡΙΛΗΨΗ ΕΡΓΑΣΙΑΣ	13
1.3 ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ	14
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ.....	15
2.1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	15
2.1.1 ΤΟ ΔΙΑΔΙΚΤΥΟ.....	15
2.1.2 Ο ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ	16
2.1.3 ΙΣΤΟΣΕΛΙΔΕΣ	17
2.2 ΤΕΧΝΟΛΟΓΙΕΣ ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ.....	19
2.2.1 Η ΓΛΩΣΣΑ HTML	19
2.2.2 Η ΓΛΩΣΣΑ CSS.....	20
2.2.3 Η ΓΛΩΣΣΑ JAVASCRIPT	21
2.2.4 Η ΓΛΩΣΣΑ PHP	22
2.2.5 ΣΥΣΤΗΜΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ MYSQL.....	23
2.2.6 ΕΞΥΠΗΡΕΤΗΤΗΣ ΙΣΤΟΥ APACHE HTTP.....	24
2.3 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ – CMS	25
2.3.1 ΤΙ ΕΙΝΑΙ ΕΝΑ CMS;.....	25
2.3.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΧΡΗΣΗΣ ΕΝΟΣ CMS.....	26
2.3.3 ΕΙΔΗ CMS	27
2.4 WORDPRESS	29
2.4.1 ΤΙ ΕΙΝΑΙ ΤΟ WORDPRESS.....	29
2.4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ WORDPRESS	30
2.4.3 ΘΕΜΑΤΑ (THEMES)	31
2.4.4 ΠΡΟΣΘΕΤΑ (PLUGINS).....	32
2.4.5 WIDGETS.....	33
2.4.6 ΔΟΜΗ ΜΙΑΣ ΣΕΛΙΔΑΣ WORDPRESS	34
2.4.7 ΒΑΣΙΚΟΙ ΟΡΟΙ ΤΟΥ WORDPRESS.....	35
ΚΕΦΑΛΑΙΟ 3: ΥΛΟΠΟΙΗΣΗ ΙΣΤΟΤΟΠΟΥ.....	36
3.1 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ.....	36
3.1.1 ΣΚΟΠΟΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ.....	36
3.1.2 ΕΡΓΑΛΕΙΑ	36

3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ	38
3.2.1 ΕΓΚΑΤΑΣΤΑΣΗ NOTEPAD++	38
3.2.2 ΕΓΚΑΤΑΣΤΑΣΗ ΧΑΜΡΡ	39
3.2.3 ΕΓΚΑΤΑΣΤΑΣΗ WORDPRESS	42
3.2.4 ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΣΘΕΤΩΝ	47
3.3 ΥΛΟΠΟΙΗΣΗ ΖΗΤΟΥΜΕΝΩΝ	49
3.3.1 ΕΠΙΛΟΓΗ ΘΕΜΑΤΟΣ	49
3.3.2 ΔΗΜΙΟΥΡΓΙΑ CHILD THEME	50
3.3.3 ΔΙΑΜΟΡΦΩΣΗ ΘΕΜΑΤΟΣ.....	51
3.3.4 ΔΗΜΙΟΥΡΓΙΑ ΣΕΛΙΔΩΝ	52
3.3.5 ΑΡΧΙΚΗ ΣΕΛΙΔΑ (ΕΡΕΥΝΗΤΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ)	54
3.3.6 ΣΕΛΙΔΑ ΑΡΘΡΩΝ (ΑΝΑΚΟΙΝΩΣΕΙΣ).....	54
3.3.7 ΣΕΛΙΔΑ ΠΡΟΓΡΑΜΜΑΤΩΝ	56
3.3.7α ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE	56
3.3.7β ΔΗΜΙΟΥΡΓΙΑ CUSTOM FIELDS GROUP.....	57
3.3.7γ ΚΩΔΙΚΑΣ: ΕΡΕΥΝΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ	58
3.3.8 ΣΕΛΙΔΑ ΕΡΓΑΣΙΩΝ	62
3.3.8α ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE ΓΙΑ ΔΙΑΘΕΣΙΜΑ ΘΕΜΑΤΑ: 62	
3.3.8β ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE ΓΙΑ ΟΛΟΚΛΗΡΩΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	63
3.3.8γ ΔΗΜΙΟΥΡΓΙΑ CUSTOM TAXONOMY ΕΙΣΗΓΗΤΩΝ	63
3.3.8δ ΔΗΜΙΟΥΡΓΙΑ CUSTOM FIELDS GROUP.....	64
3.3.8ε ΚΩΔΙΚΑΣ: ΔΙΑΘΕΣΙΜΑ ΘΕΜΑΤΑ ΠΤΥΧΙΑΚΩΝ.....	66
3.3.8στ ΚΩΔΙΚΑΣ: ΑΝΑΖΗΤΗΣΗ ΠΤΥΧΙΑΚΩΝ ΜΕ AJAX	72
3.3.9 ΣΕΛΙΔΑ ΕΠΙΚΟΙΝΩΝΙΑΣ	84
3.3.10 ΜΕΝΟΥ ΠΛΟΗΓΗΣΗΣ	85
3.3.11 WIDGETS	86
3.3.12 ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΣΘΕΤΟΥ ATTACHMENT TAXONOMIES.....	88
ΚΕΦΑΛΑΙΟ 4: ΕΠΙΛΟΓΟΣ	89
4.1 ΤΕΛΙΚΗ ΕΜΦΑΝΙΣΗ ΙΣΤΟΤΟΠΟΥ	89
4.2 ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΓΑΣΙΑΣ	93

4.3 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	93
ΚΕΦΑΛΑΙΟ 5: ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ.....	94
5.1 ΣΥΝΑΡΤΗΣΕΙΣ	94
5.1.1 functions.php	94
5.2 CSS	98
5.2.1 style.css.....	98
5.3 JAVASCRIPT	102
5.3.1 dissertations_search.js	102
5.4 ΠΡΟΣΘΕΤΟ ΣΥΝΗΜΜΕΝΩΝ	103
5.4.1 attachment_taxonomies.php.....	103
5.5 ΣΕΛΙΔΕΣ & ΠΡΟΤΥΠΑ	104
5.5.1 404.php	104
5.5.2 archive.php	105
5.5.3 category.php.....	107
5.5.4 content.php.....	108
5.5.5 content-aside.php	109
5.5.6 content-d_topic.php	110
5.5.7 content-dissertation.php	112
5.5.8 content-noresults.php	114
5.5.9 content-programs.php.....	114
5.5.10 dissertations-template.php.....	115
5.5.11 index.php.....	117
5.5.12 programs-template.php.....	117
5.5.13 search.php.....	118
5.5.14 single-d_topic.php	119
5.5.15 single-dissertation.php.....	120
5.5.16 single-programs.php	122
ΚΕΦΑΛΑΙΟ 6: ΒΙΒΛΙΟΓΡΑΦΙΑ.....	124
6.1 ΣΥΓΓΡΑΜΜΑΤΑ	124
6.2 ΔΙΑΔΥΚΤΙΑΚΕΣ ΠΗΓΕΣ.....	124

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 ΔΟΜΗ ΕΡΓΑΣΙΑΣ

Σύντομη περιγραφή των περιεχομένων των κεφαλαίων της παρούσας πτυχιακής εργασίας:

Κεφάλαιο 1: Εισαγωγή

Περίληψη και εισαγωγικά στοιχεία για το περιεχόμενο τον σκοπό και τους στόχους της.

Κεφάλαιο 2: Θεωρητικό Μέρος

Βασικό θεωρητικό υπόβαθρο για την λειτουργία του διαδικτύου και τις τεχνολογίες και τεχνικές που θα χρησιμοποιηθούν στην υλοποίηση.

Κεφάλαιο 3: Υλοποίηση

Αναλυτική περιγραφή των βημάτων για την υλοποίηση του ιστότοπου.

Κεφάλαιο 4: Επίλογος

Το τελικό αποτέλεσμα της εργασίας και τα συμπεράσματα που βγήκαν.

Κεφάλαιο 5: Παράρτημα Κώδικα

Ο πλήρης κώδικας όλων των αρχείων που δημιουργήθηκαν ή τροποποιήθηκαν από υπάρχοντα αρχεία του θέματος που χρησιμοποιήθηκε.

Κεφάλαιο 6: Βιβλιογραφία

Συγγράμματα και διαδικτυακές/ηλεκτρονικές πηγές που χρησιμοποιήθηκαν για την συγγραφή της εργασίας.

1.2 ΠΕΡΙΛΗΨΗ ΕΡΓΑΣΙΑΣ

Η πτυχιακή εργασία πραγματεύεται τον σχεδιασμό και την κατασκευή ιστότοπου (website) για το τμήμα Μηχανικών Η/Υ Συστημάτων, στον οποίο θα παρουσιάζονται τα ερευνητικά προγράμματα και πτυχιακές εργασίες του τμήματος. Η ανάπτυξη του έγινε με την χρήση του Συστήματος Διαχείρισης Περιεχομένου (**C**ontent **M**anagement **S**ystem – CMS) WordPress και για την υλοποίησή του ήταν απαραίτητη η γνώση και χρήση των εξής γλωσσών προγραμματισμού και τεχνολογιών διαδικτύου: HTML (*H*yper**T**ext **M**arkup **L**anguage), CSS (**C**ascading **S**tyle **S**heets), PHP (*H*yper**t**ext **P**reprocessor), Javascript και SQL (**S**tructured **Q**uery **L**anguage).

Ο ιστότοπος παρέχει πληροφορίες για την γενικότερη ερευνητική δραστηριότητα του τμήματος αλλά και ειδικότερα πληροφορίες για το κάθε πρόγραμμα στο οποίο έχει συμμετάσχει ή συμβάλλει με κάποιον τρόπο.

Επιπροσθέτως, δίνει την δυνατότητα στο χρήστη να αναζητήσει πτυχιακές εργασίες φοιτητών του τμήματος που έχουν αναρτηθεί και σε περίπτωση που το επιθυμεί να τις κατεβάσει στον υπολογιστή του. Εμφανίζονται επίσης και τα διαθέσιμα προς εκπόνηση θέματα πτυχιακών με όλες τις απαραίτητες σχετικές πληροφορίες και στοιχεία επικοινωνίας των εισηγητών καθηγητών που τα έχουν προτείνει.

Επιπλέον, στον ιστότοπο θα αναρτώνται σχετικά νέα και ανακοινώσεις που θα αφορούν τα ερευνητικά προγράμματα και τις πτυχιακές εργασίες.

Το WordPress παρέχει πολλές δυνατότητες στον διαχειριστή να επεξεργαστεί εύκολα το περιεχόμενο και την μορφή των ιστοσελίδων. Λαμβάνοντας αυτό υπόψη, ένα ακόμα αντικείμενο της παρούσας εργασίας ήταν και η επέκταση αυτών των δυνατοτήτων μέσω της χρήσης, δημιουργίας και επεξεργασίας των επεκτάσεων (plugins) που χρησιμοποιούνται από το σύστημα διαχείρισης.

1.3 ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΕΡΓΑΣΙΑΣ

Ο κύριος σκοπός αυτής της πτυχιακής εργασίας είναι η δημιουργία ενός ιστότοπου φιλικού, τόσο στη χρήση από έναν απλό επισκέπτη που αναζητεί πληροφορίες, όσο και στην καθημερινή διαχείριση του από τον διαχειριστή και τους συγγραφείς του περιεχομένου και των αναρτήσεών του.

Για την επιτυχή εκπλήρωση αυτού του σκοπού έπρεπε πρώτα να επιτευχθούν κάποιοι άλλοι στόχοι. Ο σημαντικότερος ήταν η μελέτη και εκμάθηση της δομής και των βασικών παρασκηνιακών λειτουργιών του συστήματος WordPress, έτσι ώστε να εκμεταλλευτούμε αποτελεσματικά τις δυνατότητες που μας παρέχει ήδη αλλά και να μπορέσουμε να δημιουργήσουμε και να ενσωματώσουμε δικές μας λειτουργίες που θα ανταποκρίνονται καλύτερα σε πιο ειδικές μας ανάγκες.

Απαραίτητο για την επίτευξη αυτού του στόχου ήταν η επαρκή γνώση τεχνολογιών του διαδικτύου και γλωσσών προγραμματισμού που χρησιμοποιούνται για την κατασκευή ιστοσελίδων (HTML, CSS, PHP, Javascript, SQL). Ένα σημαντικό μέρος αυτών των γνώσεων αυτών το αποκομίσαμε κατά την διάρκεια της φοίτησής μας, όμως για την καλύτερη κατανόηση του συστήματος και την υλοποίηση του ιστότοπου, χρειάστηκε να τις επεκτείνουμε αρκετά.

Με αυτόν τον τρόπο επετεύχθη και ένας επιμέρους αλλά εξίσου σημαντικός σκοπός: η εμπάθυνση των γνώσεων μας στο πεδίο των διαδικτυακών εφαρμογών, που είναι μια μόνιμη απαίτηση στον συνεχώς εξελισσόμενο χώρο της τεχνολογίας.

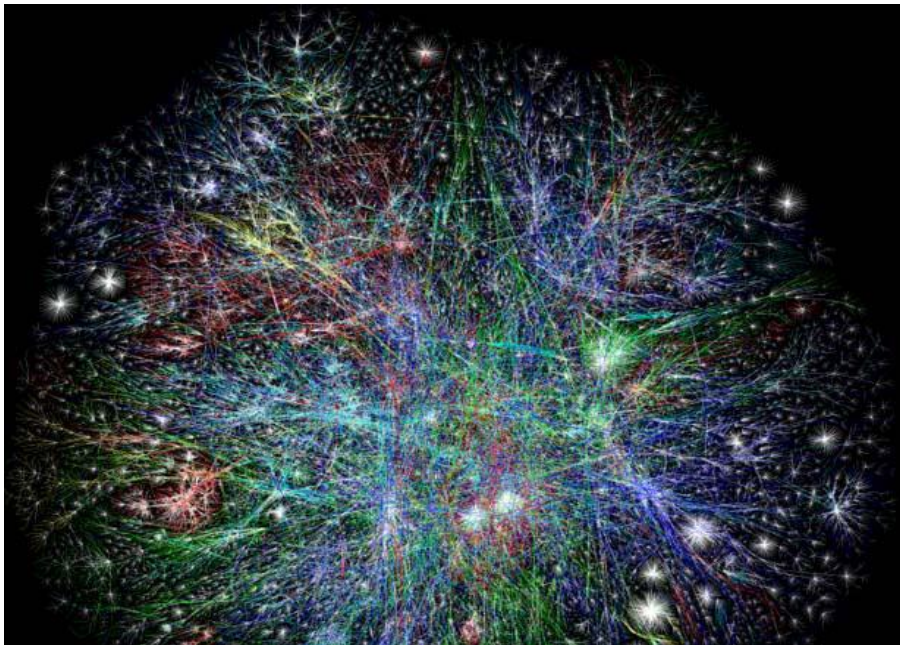
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

2.1 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

2.1.1 ΤΟ ΔΙΑΔΙΚΤΥΟ

Η λέξη 'Διαδίκτυο' είναι σύνθετη και παράγεται από τις λέξεις Διασύνδεση Δικτύων, ενώ στα αγγλικά ο όρος 'Internet' παράγεται από την συνένωση των **International Network** (Διεθνές Δίκτυο).

Το Διαδίκτυο είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων υπολογιστών που επικοινωνούν μεταξύ τους ανταλλάσσοντας δεδομένα γνωστά και ως 'πακέτα δεδομένων' (data packets). Για την επικοινωνία τους αυτή χρησιμοποιούν την σουίτα πρωτοκόλλων TCP/IP (Transmission Protocol/Internet Protocol). Αποτελείται από εκατομμύρια επιμέρους ιδιωτικά, δημόσια, ακαδημαϊκά και κυβερνητικά δίκτυα σε όλο τον κόσμο που συνδέονται με μια ευρεία γκάμα ενσύρματων και ασύρματων τεχνολογιών.



Εικόνα 2.1: Οπτική αναπαράσταση του Διαδικτύου (Project Opte).

Το Διαδίκτυο είναι η βασική υποδομή πάνω στην οποία στηρίζονται για την λειτουργία τους υπηρεσίες όπως το ηλεκτρονικό ταχυδρομείο (Electronic Mail) αλλά και ο Παγκόσμιος Ιστός (World Wide Web).

2.1.2 Ο ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ

Ο Παγκόσμιος Ιστός (World Wide Web), γνωστός απλά και ως ‘Ο Ιστός’ (The Web), εφευρέθηκε από τον Άγγλο επιστήμονα Τιμ Μπέρνερς Λι (Tim Berners-Lee) το 1989 στο ερευνητικό ίδρυμα CERN και έχει διαδραματίσει κεντρικό ρόλο στην ανάπτυξη της εποχής της πληροφορίας. Είναι η δημοφιλέστερη υπηρεσία του διαδικτύου και το βασικό εργαλείο που επιτρέπει στους χρήστες να έχουν πρόσβαση σε μια τεράστια γκάμα πληροφοριών και περιεχομένου πολυμέσων.

Η αλληλεπίδραση των χρηστών με τον Παγκόσμιο Ιστό γίνεται κυρίως διαμέσου ειδικών διαδικτυακών εγγράφων που ονομάζονται ιστοσελίδες (web pages). Για την προβολή τους χρησιμοποιούνται ειδικά προγράμματα λογισμικού που ονομάζονται φυλλομετρητές ιστοσελίδων (web browsers) ή απλά περιηγητές, οι οποίοι χρησιμοποιούν το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol – HTTP) για την μεταφορά δεδομένων. Ο φυλλομετρητής επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση στις πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες μέσω υπερσυνδέσμων (hyperlinks) και μέσω διευθύνσεων ‘πόρων’ του Παγκόσμιου Ιστού ή URLs (Uniform Resource Locators)

Μερικοί από τους πιο διαδομένους φυλλομετρητές είναι οι εξής:

- Google Chrome
- Mozilla Firefox
- Microsoft Internet Explorer / Microsoft Edge
- Apple Safari
- Opera



Εικόνα 2.2: Δημοφιλείς Φυλλομετρητές.

2.1.3 ΙΣΤΟΣΕΛΙΔΕΣ

Ιστοσελίδα (web page) είναι ένα είδος εγγράφου του Παγκόσμιου Ιστού και μπορεί να περιλαμβάνει πληροφορίες με την μορφή κειμένου, υπερκειμένου (hypertext), ήχου, εικόνας και βίντεο. Μια συλλογή από διασυνδεδεμένες ιστοσελίδες στο ίδιο όνομα χώρου (domain) συνθέτουν έναν ιστότοπο ή ιστοχώρο (website) στον οποίο ο χρήστης μπορεί να περιηγηθεί από την μία σελίδα στην άλλη χρησιμοποιώντας συνδέσμους (links).

Οι ιστοσελίδες είναι συνήθως γραμμένες σε HTML (**H**yper**T**ext **M**arkup **L**anguage) που είναι μια γλώσσα σήμανσης υπερκειμένου με την οποία ορίζεται η συγκεκριμένη δομή που θα έχουν τα δεδομένα μιας ιστοσελίδας και μεταφράζεται από τον φυλλομετρητή. Σε ένα δίκτυο, οι φυλλομετρητές παίζουν τον ρόλο του 'πελάτη' (client) που ανακτούν ιστοσελίδες μέσω HTTP αιτήσεων από κάποιον εξυπηρετητή ή διακομιστή ιστού (web server) ο οποίος αναλαμβάνει να 'εξυπηρετεί' τέτοιες αιτήσεις και να στέλνει τα δεδομένα που του ζητούνται, εφόσον φυσικά υπάρχουν τα εκάστοτε κατάλληλα δικαιώματα πρόσβασης.

Οι ιστοσελίδες χωρίζονται σε 2 κύριες κατηγορίες:

Στατικές:

Στατικές ιστοσελίδες (static webpages) είναι αυτές των οποίων το περιεχόμενο μεταφέρεται και εμφανίζεται σε όλους τους χρήστες ακριβώς με την μορφή που είναι αποθηκευμένο στον εξυπηρετητή. Βρίσκονται συνήθως σε HTML μορφή και έχουν το πλεονέκτημα να μην απαιτούν ιδιαίτερες γνώσεις προγραμματισμού για την δημιουργία τους. Επιπλέον μπορούν να προβάλλονται από τον φυλλομετρητή και χωρίς την μεσολάβηση διακομιστή, που σημαίνει πως μπορούν μεταφέρονται και ως απλά αρχεία σε κάποιο αποθηκευτικό μέσο.

Από τα βασικά τους μειονεκτήματα είναι η χαμηλή διαδραστικότητα με τον χρήστη και η δυσκολία στην τροποποίηση και εισαγωγή νέου περιεχομένου, ιδιαίτερα αν ο αριθμός τους είναι μεγάλος. Πρακτικά αυτό σημαίνει πως δεν έχουν πολλές δυνατότητες εξέλιξης μετά την δημιουργία τους και μένουν, όπως το όνομά τους υποδεικνύει, στατικές.



Εικόνα 2.3: Αποστολή στατικής σελίδας από τον διακομιστή.

Δυναμικές:

Δυναμικές ιστοσελίδες (dynamic webpages) είναι αυτές οι οποίες δημιουργούνται από ένα σενάριο εντολών την στιγμή της πρόσβασης ή την στιγμή που ο χρήστης αλληλεπιδρά με κάποιο στοιχείο της σελίδας.

Τα σενάρια εντολών μπορούν να εκτελούνται είτε τοπικά στον φυλλομετρητή είτε στον απομακρυσμένο διακομιστή. Το που θα εκτελούνται εξαρτάται από το ποια γλώσσα προγραμματισμού σεναρίων (scripting language) έχει χρησιμοποιηθεί και για ποιο σκοπό.

Λειτουργίες που απαιτούν επικοινωνία με την βάση δεδομένων είναι γραμμένες σε γλώσσα που τρέχει στο λογισμικό του διακομιστή (server-side scripting), όπου πρώτα διερμηνεύονται και στη συνέχεια αποστέλλονται τα ζητούμενα δεδομένα στον πελάτη. Λειτουργίες που δεν χρειάζονται άμεσα δεδομένα από την βάση (π.χ. έλεγχος συμπλήρωσης πεδίου σε φόρμα) γράφονται σε γλώσσες που μπορεί να εκτελέσει ο φυλλομετρητής χωρίς την συμβολή του διακομιστή (client-side scripting).



Εικόνα 2.4: Σελίδα που περιέχει εντολές PHP οι οποίες εκτελούνται από το λογισμικό του διακομιστή και αποστέλλεται σελίδα HTML με δεδομένα από την βάση δεδομένων MySQL.

Μεγάλο πλεονέκτημα των δυναμικών σελίδων είναι η ευκολία και η ταχύτητα με την οποία μπορεί να γίνουν αλλαγές στο περιεχόμενο τους χρησιμοποιώντας έναν πίνακα διαχείρισης. Αν και η ανάπτυξη τους απαιτεί αρκετές προγραμματιστικές γνώσεις, η ύπαρξη Συστημάτων Διαχείρισης Περιεχομένου (**C**ontent **M**anagement **S**ystems – **CMS**) τα οποία χρησιμοποιούνται για την ανάπτυξη δυναμικών σελίδων, έχει απλουστεύσει σημαντικά την διαδικασία.

2.2 ΤΕΧΝΟΛΟΓΙΕΣ ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

2.2.1 Η ΓΛΩΣΣΑ HTML

<HTML>

Η HTML (HyperText Markup Language) είναι μια γλώσσα σήμανσης υπερκειμένου. Διαφέρει από μια γλώσσα προγραμματισμού καθώς δεν έχει δυνατότητες εκτέλεσης εντολών και πράξεων. Είναι βασισμένη σε χαρακτηρισες απλού κειμένου και χρησιμοποιείται για την περιγραφή της δομής και του περιεχομένου μιας ιστοσελίδας.

Τα δομικά στοιχεία της HTML είναι οι ετικέτες (tags). Συνήθως συναντώνται ανά ζεύγη με την πρώτη να ονομάζεται ετικέτα εκκίνησης (opening tag) και την δεύτερη ετικέτα τερματισμού (closing tag), ανάμεσα στις οποίες υπάρχει το περιεχόμενο του οποίου την δομή οποίο περιγράφουν. Οι ετικέτες έχουν την εξής γενική μορφή:

`< ΕΤΙΚΕΤΑ ιδιότητα1="τιμή1" > Περιεχόμενο </ ΕΤΙΚΕΤΑ >`

Κάποια στοιχεία περιγράφονται ως 'άδεια' διότι δεν έχουν περιεχόμενο και δεν ακολουθούνται από ετικέτα τερματισμού.

Αυτός ο τρόπος γραφής αναγνωρίζεται από τον φυλλομετρητή ο οποίος τον χρησιμοποιεί ως 'οδηγό' για την σύνθεση μιας ιστοσελίδας και για την ερμηνεία του περιεχομένου τους χωρίς να εμφανίζονται στον επισκέπτη της σελίδας.

Στην παρακάτω εικόνα μπορούμε να δούμε ένα παράδειγμα κώδικα HTML στα αριστερά και την έξοδο (output) που θα εμφανίσει ο φυλλομετρητής στην οθόνη στα δεξιά:

```
<!DOCTYPE html>
<html>
<head>
<title>Τίτλος Σελίδας</title>
</head>
<body>

<h1>Αυτή είναι μια κεφαλίδα</h1>
<p>Αυτή είναι μία παράγραφος.</p>

</body>
</html>
```

Αυτή είναι μια κεφαλίδα

Αυτή είναι μία παράγραφος.

Εικόνα 2.5: Δείγμα κώδικα HTML και εξόδου.

2.2.2 Η ΓΛΩΣΣΑ CSS

CSS

Η γλώσσα επικαλυπτόμενων στυλ μορφοποίησης ή CSS (**C**ascading **S**tyle **S**heets) χρησιμοποιείται για τον έλεγχο της εμφάνισης εγγράφων που είναι γραμμένα σε κάποια γλώσσα σήμανσης, όπως η HTML.

Αν και η HTML παρέχει δυνατότητες οπτικής μορφοποίησης των εγγράφων υπερκειμένου, με την CSS αυτές επεκτείνονται σημαντικά. Επιπροσθέτως, η χρήση της μας δίνει την δυνατότητα διαχωρισμού του περιεχομένου μιας ιστοσελίδας (HTML) από την εμφάνισή της. Μερικά από τα σημαντικότερα πλεονεκτήματα αυτού του διαχωρισμού είναι τα εξής:

- Αποφεύγεται η επανάληψη των ίδιων κομματιών κώδικα για την περιγραφή της εμφάνισης ίδιων στοιχείων που μπορεί να βρίσκονται σε πολλές σελίδες ενός ιστότοπου.
- Ευκολία στην επεξεργασία της εμφάνισης πολλών στοιχείων ενός ιστότοπου που μπορεί να χρησιμοποιούνται σε μεγάλο αριθμό σελίδων, κάνοντας τις απαραίτητες τροποποιήσεις σε ένα μόνο αρχείο CSS.
- Μείωση του εύρους ζώνης (bandwidth) που είναι απαραίτητο για την φόρτωση της ιστοσελίδας, άρα γρηγορότερη ταχύτητα πρόσβασης των επισκεπτών της. Αυτό είναι ιδιαίτερα σημαντικό για την προσβασιμότητα του ιστότοπου από συνδέσεις χαμηλής ταχύτητας.

Η γενική μορφή των εντολών της CSS είναι η παρακάτω:

```
css_επιλογέας { ιδιότητα: τιμή; }
```

Με τον css επιλογέα (css selector) 'στοχεύουμε' ένα συγκεκριμένο στοιχείο στην HTML προσδιορίζοντας μια ιδιότητα του ή ένα σύνολο ετικετών των οποίων το περιεχόμενο θέλουμε να μορφοποιήσουμε. Με την ιδιότητα προσδιορίζουμε τι ακριβώς θέλουμε να αλλάξουμε στην εμφάνιση του στοιχείου ή στοιχείων.

```
body {  
    background-color: #d0e4fe;  
}  
  
h1 { color: red; }  
  
p {  
    font-family: "Times New Roman";  
    font-size: 22px;  
}
```

Αυτή είναι μια κεφαλίδα

Αυτή είναι μια παράγραφος.

Εικόνα 2.6: Μορφοποίηση του κώδικα της εικόνας 5 με CSS.

2.2.3 Η ΓΛΩΣΣΑ JAVASCRIPT



Η JavaScript είναι μια γλώσσα σεναριακού προγραμματισμού η οποία συνήθως εκτελείται στον φυλλομετρητή, δηλαδή στην πλευρά του πελάτη (client-side) και χρησιμοποιείται κυρίως για την δημιουργία στοιχείων και περιεχομένου που αλληλεπιδρά με τον χρήστη. Ωστόσο, τα τελευταία χρόνια με την εμφάνιση του Node.js (μοντέλο προγραμματισμού βασισμένο στα γεγονότα) επανέρχεται ξανά η χρήση της και στον διακομιστή, όπως συνέβαινε και τον πρώτο καιρό της δημιουργίας της.

Ένα σημαντικό χαρακτηριστικό της εκτέλεσής της στον browser είναι το ότι δεν χρειάζεται η επαναφόρτωση της σελίδας για την αλληλεπίδραση με τον χρήστη. Αυτό της το χαρακτηριστικό έχει δώσει την δυνατότητα στους προγραμματιστές να φορτώνουν ασύγχρονα ορισμένα στοιχεία μιας σελίδας, ακόμα και για την ανάκτηση πληροφοριών από την βάση δεδομένων χρησιμοποιώντας τεχνικές γνωστές ως AJAX (**A**synchronous **J**avascript **A**nd **X**ML) τις οποίες θα εξηγήσουμε πιο αναλυτικά στο κομμάτι της υλοποίησης.

Η σύνταξή της είχε αρχικά βασιστεί στην γλώσσα C αλλά κατά την εξέλιξή της έχει ενσωματώσει στοιχεία και από νεότερες γλώσσες. Μπορεί να συνδυαστεί τόσο με στατικές, όσο και με δυναμικές σελίδες.

Όταν χρησιμοποιείται σε έγγραφο HTML, περικλείεται από την ετικέτα `<script type="text/javascript"></script>`. Αξίζει να σημειωθεί πως στην HTML5, την πιο πρόσφατη έκδοση της γλώσσας σήμανσης HTML, η ιδιότητα `script type="text/javascript"` δεν είναι απαραίτητη καθώς όλοι οι μοντέρνοι browsers θεωρούν την Javascript ως την προεπιλεγμένη γλώσσα εκτέλεσης σεναρίων.

```
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == null || x == "") {
        alert("Name must be filled out");
        return false;
    }
}
```

Εικόνα 2.7: Παράδειγμα κώδικα ελέγχου συμπλήρωσης πεδίου σε Javascript.

2.2.4 Η ΓΛΩΣΣΑ PHP



Η PHP είναι μια γλώσσα προγραμματισμού κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών και ιστοσελίδων με δυναμικό περιεχόμενο. Εκτελείται στην πλευρά του διακομιστή (server-side) και αποστέλλει το τελικό αποτέλεσμα στον φυλλομετρητή-πελάτη.

Η ονομασία της προέρχεται από τα αρχικά **P**ersonal **H**ome **P**age καθώς είχε δημιουργηθεί από τον τότε φοιτητή, τον Rasmus Lerdorf, για προσωπική χρήση. Με την διάδοση και εξέλιξη της η σημασία της ονομασίας της άλλαξε στο αναδρομικό ακρώνυμο **H**ypertext **P**re**P**rocessor (Προεπεξεργαστής Υπερκειμένου) με το οποίο είναι σήμερα γνωστή.

Για να εκτελεστεί ένα αρχείο με κώδικα PHP, θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. .php, .php4, .phtml). Σε αντίθεση με την Javascript, η PHP δεν μπορεί να ενσωματωθεί σε αρχείο με HTML αν δεν έχουν γίνει οι κατάλληλες ρυθμίσεις στον διακομιστή/εξυπηρετητή ιστού (web server) για την επεξεργασία και μετάφραση του κώδικα PHP σε μορφή που μπορεί να αναγνωρίσει ο φυλλομετρητής (browser). Ως εκ τούτου, η εκτέλεση του κώδικα απαιτεί την ύπαρξη συμβατού λογισμικού διακομιστή. Λόγω της ευρείας χρήσης της PHP στις διαδικτυακές εφαρμογές, τα περισσότερα προγράμματα διακομιστή είναι συμβατά με αυτήν. Το πιο διαδομένο από αυτά είναι το λογισμικό ανοιχτού κώδικα (Open Source Software) Apache HTTP ή απλά Apache.

Ο κώδικας PHP μιας ιστοσελίδας πρέπει να περικλείεται ανάμεσα από τις ετικέτες `<?php` και `?>` για να μπορεί να ξεχωρίζει ο μεταφραστής τα κομμάτια προς επεξεργασία. Οτιδήποτε εκτός των ετικετών αυτών το αγνοεί.

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

Εικόνα 2.8: Δείγμα κώδικα PHP ενσωματωμένου σε HTML.

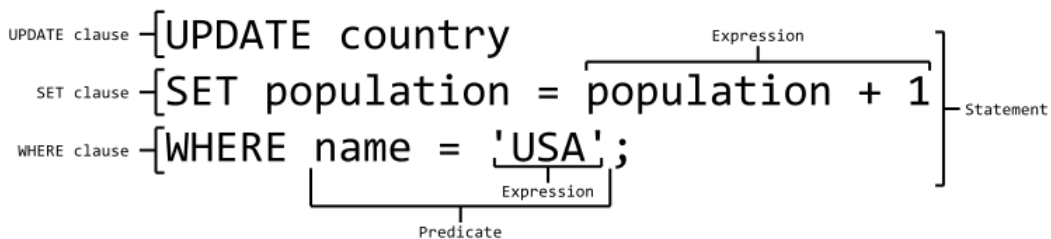
2.2.5 ΣΥΣΤΗΜΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ MYSQL



Η MySQL είναι μια γρήγορης απόδοσης και μεγάλων δυνατοτήτων υλοποίηση ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων (**R**elational **D**atabase **M**anagement **S**ystem, RDBMS). Χρησιμοποιεί την δομημένη γλώσσα επερωτήσεων SQL (**S**tructured **Q**uery **L**anguage), την πιο διαδεδομένη γλώσσα διαχείρισης βάσεων δεδομένων.

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία:

- *Clauses*, που σε μερικές περιπτώσεις είναι προαιρετικά, αλλά αναγκαία συστατικά των δηλώσεων και ερωτήσεων.
- *Expressions* που παραγάγουν είτε τις κλιμακωτές τιμές είτε πίνακες που αποτελούνται από στήλες και σειρές στοιχείων.
- *Predicates* που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν ως σωστό ή λάθος.
- *Queries* που ανακτούν τα στοιχεία βασισμένες σε ειδικά κριτήρια.
- *Statements* που μπορούν να έχουν μια επίδραση στα σχήματα και τα στοιχεία ή μπορούν να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις από άλλα προγράμματα.



Εικόνα 2.9: Παράδειγμα της δομής ενός Query.

Η MySQL, εκτός από το πιο δημοφιλές σύστημα διαχείρισης βάσεων, αποτελεί και το προεπιλεγμένο δομικό στοιχείο μιας σειράς ολοκληρωμένων πακέτων λογισμικού που λειτουργούν ως πλατφόρμες εκτέλεσης δυναμικών ιστοσελίδων. Γνωστά ως AMP (**A**rache, **M**ySQL, **P**HP/Perl/Python) τα πακέτα αυτά έχουν απλοποιήσει την διαδικασία εγκατάστασης των απαραίτητων εργαλείων για το στήσιμο ενός ιστότοπου.

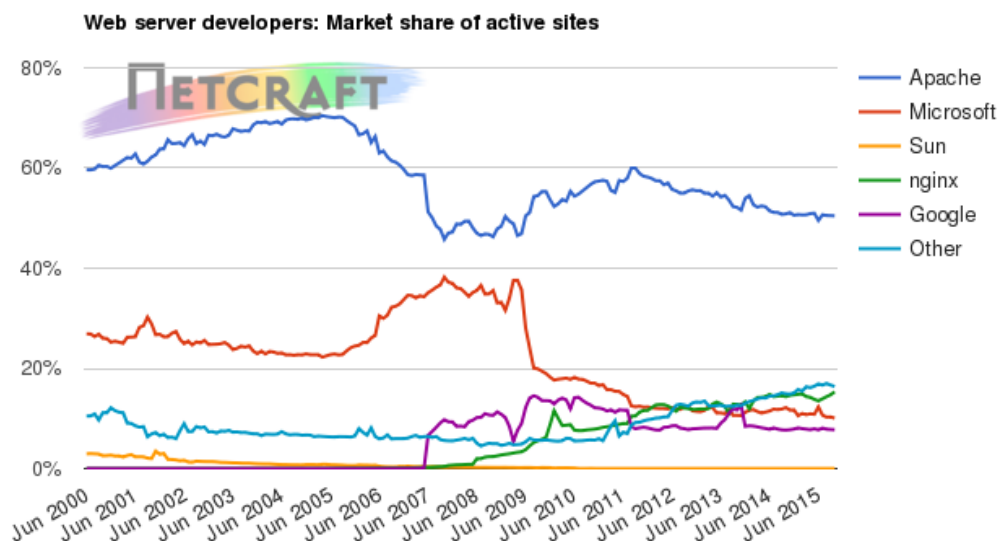
2.2.6 ΕΞΥΠΗΡΕΤΗΤΗΣ ΙΣΤΟΥ APACHE HTTP



Ο Apache HTTP ή απλά Apache είναι ο πιο διαδεδομένος εξυπηρετητής / διακομιστής του παγκόσμιου ιστού (Web server). Είναι λογισμικό ανοιχτού κώδικα (open source) το οποίο αναπτύσσεται και συντηρείται από μια ανοιχτή κοινότητα προγραμματιστών υπό την επιτήρηση του Apache Software Foundation. Η συμβατότητά του με μια μεγάλη ποικιλία λειτουργικών συστημάτων (Windows, Unix, Linux, Mac OS κ.α.) και το μεγάλο εύρος δυνατοτήτων και χαρακτηριστικών του, το καθιστούν ως το πλέον δημοφιλές λογισμικό διακομιστή παγκοσμίως.

Διανέμεται συχνά ως μέρος των προαναφερθέντων πακέτων λογισμικού AMP που συνδυάζουν τον Apache, την MySQL και την PHP. Αναλόγως το λειτουργικό στο οποίο γίνεται η εγκατάστασή τους, υπάρχουν διαφορετικές διανομές πακέτων: LAMP (για Linux), WAMP (για Windows), MAMP (για Macintosh) SAMP (για Solaris και FAMP (για FreeBSD). Υπάρχουν όμως και διανομές συμβατές με περισσότερα από ένα λειτουργικά, όπως είναι το XAMPP που έχουμε χρησιμοποιήσει για την τοπική υλοποίηση του παρόντος ιστότοπου της πτυχιακής εργασίας.

Αντίπαλο δέος του Apache όσο αφορά το συνολικό ποσοστό χρήσης τους σε ιστότοπους, είναι ο IIS της Microsoft, ακολουθούμενος από τον Nginx, ο οποίος μάλιστα σύμφωνα με πρόσφατη έρευνα [[October 2015 Web Server Survey](#)] προηγείται του ISS στους ενεργούς ιστότοπους και τον ξεπερνάει μόνο ο χρόνια πρωτοπόρος του χώρου Apache.



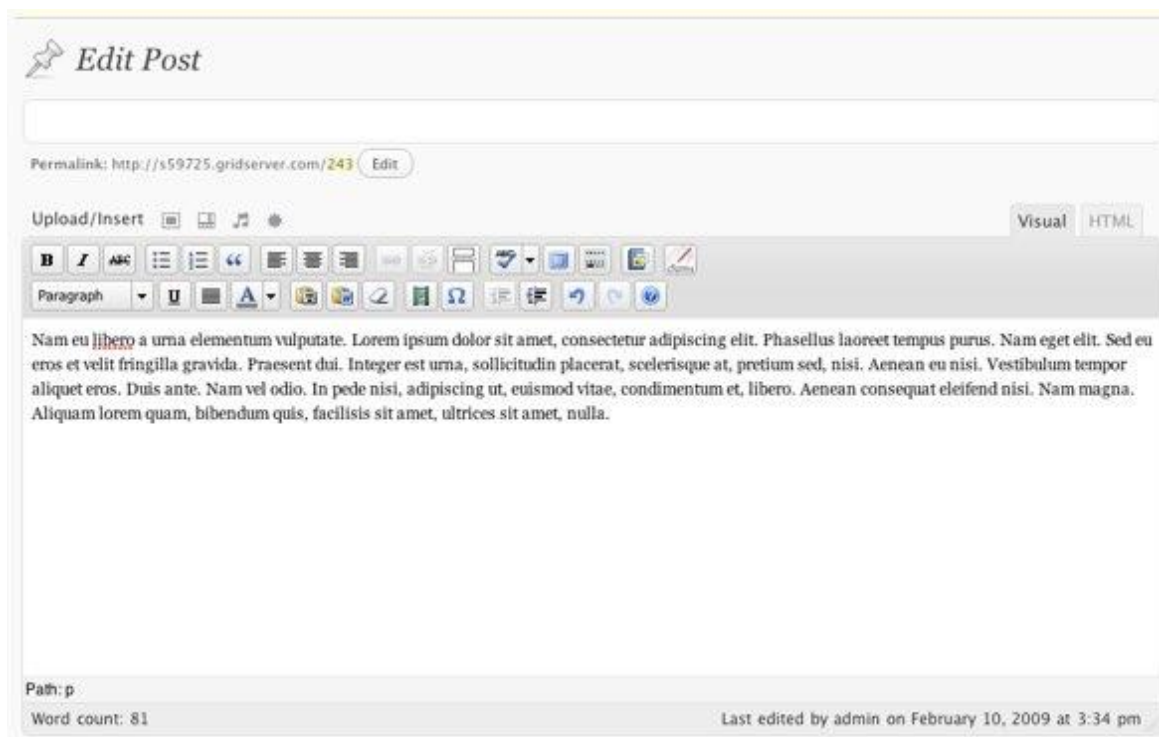
Εικόνα 2.10: Ποσοστά χρήσης των servers στους ενεργούς ιστότοπους.

2.3 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ – CMS

2.3.1 ΤΙ ΕΙΝΑΙ ΕΝΑ CMS;

Ένα σύστημα διαχείρισης περιεχομένου ή αλλιώς CMS (**C**ontent **M**anagement **S**ystem), είναι ένα σύνολο διαδικτυακών εφαρμογών που επιτρέπουν στον χρήστη να διαχειρίζεται το περιεχόμενο (κείμενο, εικόνες, βίντεο, πίνακες κλπ.) ενός δυναμικού ιστότοπου με πολύ εύκολο και γρήγορο τρόπο.

Η χρήση των εφαρμογών αυτών δεν απαιτεί ειδικές προγραμματιστικές ή άλλες σχετικές με την ανάπτυξη ιστοσελίδων γνώσεις. Διαμέσου ενός εύκολου στη χρήση πίνακα ελέγχου, δίνετε η δυνατότητα στον διαχειριστή να επεξεργαστεί το περιεχόμενο μιας σελίδας, ενώ τα κείμενα γράφονται και μορφοποιούνται συνήθως με ενσωματωμένους ή online WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) επεξεργαστές κειμένου, οι οποίοι είναι παρόμοιοι με το MS Word.



Εικόνα 2.11: WYSIWYG επεξεργαστής κειμένου ενός CMS.

Οι αλλαγές στον ιστότοπο μπορούν να γίνουν από οποιονδήποτε υπολογιστή που είναι συνδεδεμένος στο Διαδίκτυο, χωρίς να χρειάζεται εγκατάσταση ειδικού λογισμικού. Μέσω ενός απλού φυλλομετρητή, ο χρήστης μπορεί να έχει πλήρη πρόσβαση στον πίνακα ελέγχου και να τον ενημερώσει άμεσα.

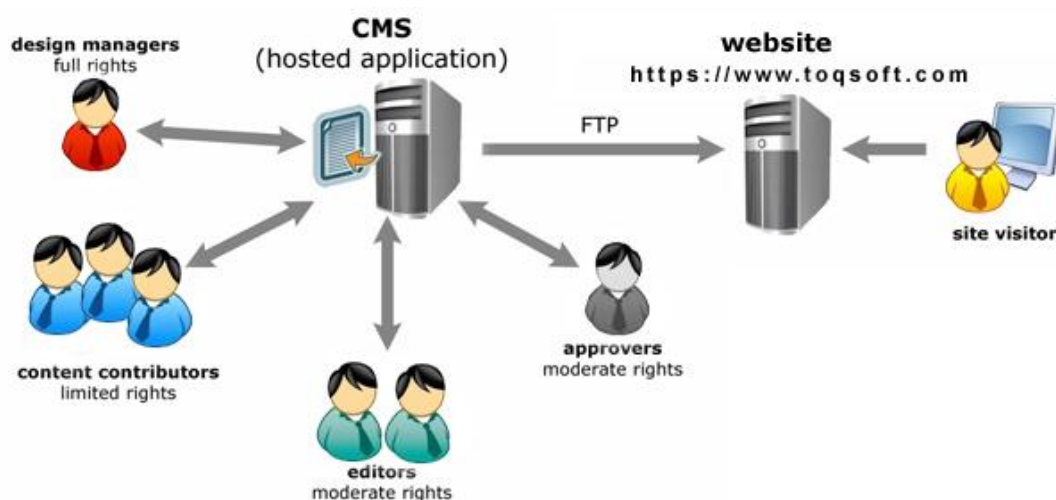
Σημείωση:

Καθώς ο όρος CMS είναι πολύ γενικός και περιλαμβάνει έναν αριθμό υποκατηγοριών με αντίστοιχα πεδία χρήσης, η συγκεκριμένη κατηγορία που περιγράφουμε ονομάζεται Web CMS (WCMS) αλλά συχνά αναφέρεται απλά ως CMS αφού είναι η πιο κοινή.

2.3.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΧΡΗΣΗΣ ΕΝΟΣ CMS

Η χρήση ενός CMS προσφέρει μια μεγάλη ποικιλία δυνατοτήτων και πλεονεκτημάτων στον ιδιοκτήτη ή/και διαχειριστή ενός ιστότοπου. Μερικά από τα σημαντικότερα είναι:

- Μειωμένο κόστος συντήρησης και διαχείρισης των σελίδων καθώς και μειωμένος φόρτος στον εξυπηρετητή λόγω της φύσης των δυναμικών ιστοσελίδων.
- Διαχωρισμός του περιεχομένου από τον σχεδιασμό και την παρουσίαση του ιστότοπου δίνοντας την δυνατότητα εύκολης αλλαγής της εμφάνισής του χωρίς να χρειαστεί να αλλάξει το περιεχόμενο.
- Δυνατότητα ταυτόχρονης επεξεργασίας και ενημέρωσης περιεχομένου από πολλούς χρήστες και αποθήκευσης διαφορετικών εκδόσεων του ίδιου αρχείου.
- Αυτοματοποίηση διαδικασιών όπως την διασύνδεση των σελίδων μεταξύ τους ανανεώνοντας τους υπερσυνδέσμους που οδηγούν σε αυτές.
- Δυνατότητα επεκτάσεων και προσθήκη επιπλέον δυνατοτήτων αναλόγως των αναγκών του ιστότοπου.
- Εύκολη και άμεση εφαρμογή αναβαθμίσεων του συστήματος και των επιμέρους στοιχείων του χωρίς να χρειαστεί να μείνει offline ο ιστότοπος.
- Προσδιορισμός ρόλων και δικαιωμάτων πρόσβασης σε λειτουργίες και περιεχόμενο για διαφορετικούς τύπους χρηστών.



Εικόνα 2.12: Σχηματική απεικόνιση χρηστών με διαφορετικά δικαιώματα .

2.3.3 ΕΙΔΗ CMS

Τα CMS διακρίνονται σε ορισμένες κατηγορίες ανάλογα με κάποια βασικά χαρακτηριστικά τα οποία παρουσιάζουν. Κατηγοριοποιούνται κυρίως ανάλογα με το είδος του παρόχου τους και ανάλογα με το που βρίσκεται ο χώρος που αποθηκεύεται και διαχειρίζεται το CMS.

Με βάση τον χώρο αποθήκευσης και διαχείρισης:

- *ASP (Application Service Provider) CMS:* Συστήματα Υποστήριξης Παρόχου Υπηρεσίας των οποίων ο κατασκευαστής φιλοξενεί τα δεδομένα και το λογισμικό στους διακομιστές της εταιρίας του αναλαμβάνοντας την συνεχή συντήρηση του εξοπλισμού και την εξέλιξη του CMS για τον πελάτη.
- *Licensed CMS:* Συστήματα με παροχή άδειας όπου ο πάροχος του παρέχει την άδεια χρήσης του λογισμικού αλλά εμπλέκεται στην διαδικασία εγκατάστασης και διαχείρισής του, τα οποία είναι ευθύνη του χρήστη.

Με βάση το είδος του παρόχου:

- *Commercial:* Εμπορικό λογισμικό κλειστού κώδικα που προέρχεται από κερδοσκοπικές ή μη εταιρείες οι οποίες το πουλάνε και στη συνέχεια παρέχουν τεχνική υποστήριξη.



Εικόνα 2.13: Εμπορικά CMS.

- *Open Source:* Λογισμικό ανοιχτού κώδικα που αναπτύσσεται και συντηρείται από μια κοινότητα προγραμματιστών και διανέμεται ελεύθερα. Η δωρεάν διανομή του δεν συνεπάγεται και χωρίς κόστος, αφού δεν παρέχεται στον χρήστη κάποιου είδους εμπορική υποστήριξη και συχνά υπάρχει έλλειψη επαρκούς τεκμηρίωσης.



Εικόνα 2.14: CMS Ανοιχτού κώδικα.

- *Managed Open Source*: συνδυασμός των δύο προηγούμενων προσεγγίσεων όπου ένας πάροχος υιοθετεί μια λύση ανοιχτού κώδικα ως βασική του πλατφόρμα και στη συνέχεια την προσφέρει εμπορικά σε συνδυασμό με συμπληρωματικές υπηρεσίες τεχνικής υποστήριξης.

Είναι φανερό λοιπόν πως υπάρχουν πολλές επιλογές στον χώρο των CMS και η τελική απόφαση μπορεί να μην είναι πάντα εύκολη ή προφανής. Κάθε περίπτωση είναι διαφορετική και για την επιλογή της κατάλληλης πλατφόρμας, ο χρήστης θα πρέπει να αναλογιστεί παραμέτρους όπως:

- Η οικειότητά του με την τεχνολογία και το διαδίκτυο.
- Ο βασικός στόχος του ιστότοπου.
- Το είδος του περιεχομένου και η πολυπλοκότητα των λειτουργιών του ιστότοπου.
- Τον βαθμό αυτονομίας στην διαχείρισή του.

2.4 WORDPRESS

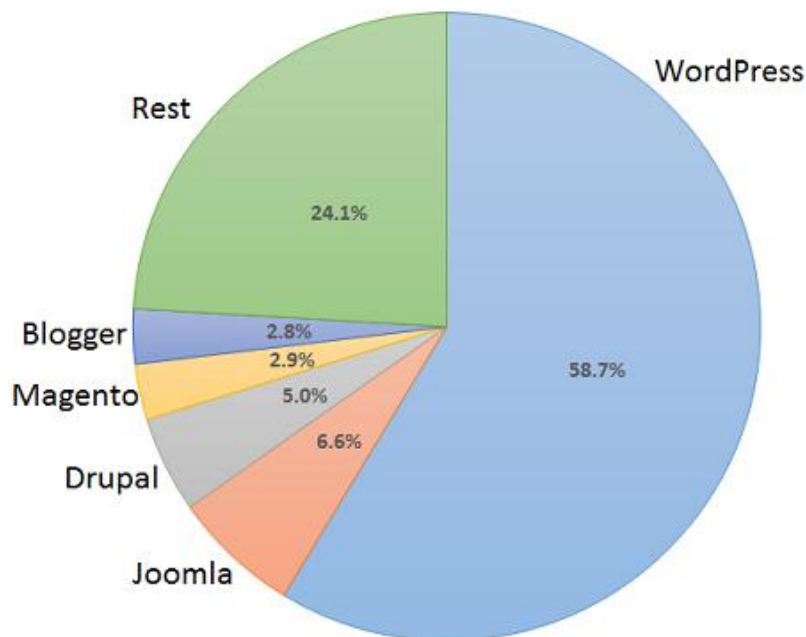


2.4.1 ΤΙ ΕΙΝΑΙ ΤΟ WORDPRESS

Το WordPress είναι ένα ελεύθερο σύστημα διαχείρισης περιεχομένου ανοιχτού κώδικα, βασισμένο σε PHP και MySQL. Αναπτύχθηκε αρχικά ως πλατφόρμα δημιουργίας και δημοσίευσης ιστολογίων (blogs), όμως τα τελευταία χρόνια έχει εξελιχθεί σε μια ολοκληρωμένη λύση CMS που μπορεί να καλύψει τις ανάγκες για την δημιουργία και διαχείριση κάθε είδους ιστότοπου.

Η φιλικότητα στη χρήση αλλά και οι άριστες επιδόσεις του WordPress όσο αφορά την βελτιστοποίηση των ιστοσελίδων για τις μηχανές αναζήτησης (**Search Engine Optimization**), έχουν συμβάλει καθοριστικά στην ευρεία του διάδοση του και στην εδραίωσή του ως το κυρίαρχο σύστημα διαχείρισης περιεχομένου αυτήν την στιγμή στο Διαδίκτυο, με μεγάλη διαφορά.

Σύμφωνα μάλιστα με την τελευταία έρευνα του [Νοεμβρίου 2015](#) από το W3Techs, το WordPress χρησιμοποιείται από το 58.7% των ιστότοπων που είναι βασισμένα σε κάποιο CMS, και από το 25% του συνόλου των ιστότοπων παγκοσμίως!



Εικόνα 2.15: Μερίδιο αγοράς ιστότοπων που χρησιμοποιούν CMS [Νοέμβριος 2015].

2.4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ WORDPRESS

Η τεράστια απήχηση του WordPress δεν είναι τυχαία, καθώς έχει πολλά πλεονεκτήματα έναντι ανταγωνιστικών CMS. Μερικά από τα πιο σημαντικά είναι τα εξής:

- ❖ *Ευκολία χρήσης:* Στον πυρήνα του, το WordPress είναι ένα ιδιαίτερα απλό σύστημα διαχείρισης περιεχομένου. Τόσο η εγκατάσταση και το στήσιμο ενός βασικού ιστότοπου, όσο και η καθημερινή διαχείριση του περιεχομένου του δεν χρειάζονται ιδιαίτερες γνώσεις, κάτι που κάνουν την επιλογή του για αυτόν τον σκοπό εξαιρετικά εύκολη.
- ❖ *SEO:* Οι μηχανές αναζήτησης λατρεύουν το WordPress, λόγω του καλογραμμένου του κώδικα. Χρησιμοποιώντας το για την ανάπτυξη ενός ιστότοπου έχουμε καλές πιθανότητες να έχει μεγαλύτερη προτεραιότητα σε αποτελέσματα αναζήτησης.
- ❖ *Ευελιξία και επεκτασιμότητα:* Είναι εξαιρετικά ευέλικτο και ευπροσάρμοστο στις εκάστοτε ανάγκες μας. Το τεράστιο πλήθος των επεκτάσεων και θεμάτων που είναι διαθέσιμα για το WordPress, σημαίνει πως μπορούμε να δημιουργήσουμε από το πιο απλό ιστολόγιο μέχρι το πιο πολύπλοκο ηλεκτρονικό κατάστημα.
- ❖ *Μεγάλη κοινότητα τεχνικής υποστήριξης:* Η δημοφιλία του συνεπάγεται και μια πολύ μεγάλη κοινότητα προγραμματιστών και χρηστών σε όλο τον κόσμο που προσφέρουν την υποστήριξή τους για κάθε είδους πρόβλημα που μπορεί να αντιμετωπίσουμε χωρίς κόστος. Μερικά από τα forums που μπορεί κανείς να βρει δωρεάν υποστήριξη είναι το wordpress.org/support (επίσημο support forum) και το μεγαλύτερο ανεπίσημο forum για WordPress, το wordpress.stackexchange.com.

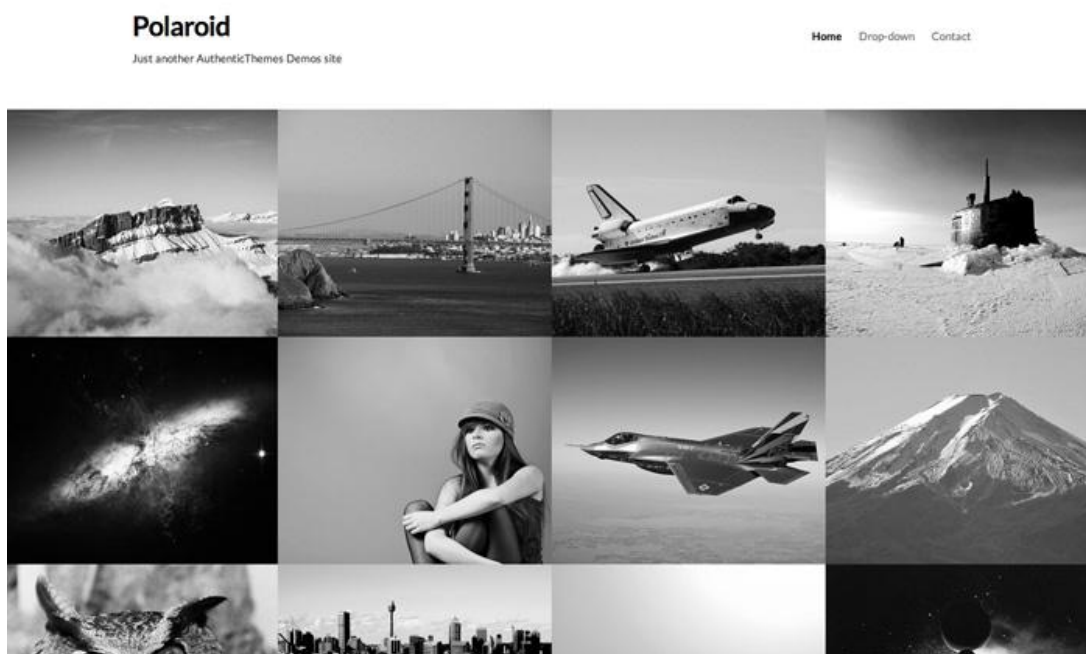


Εικόνα 2.16: Η κοινότητα του WordPress.

2.4.3 ΘΕΜΑΤΑ (THEMES)

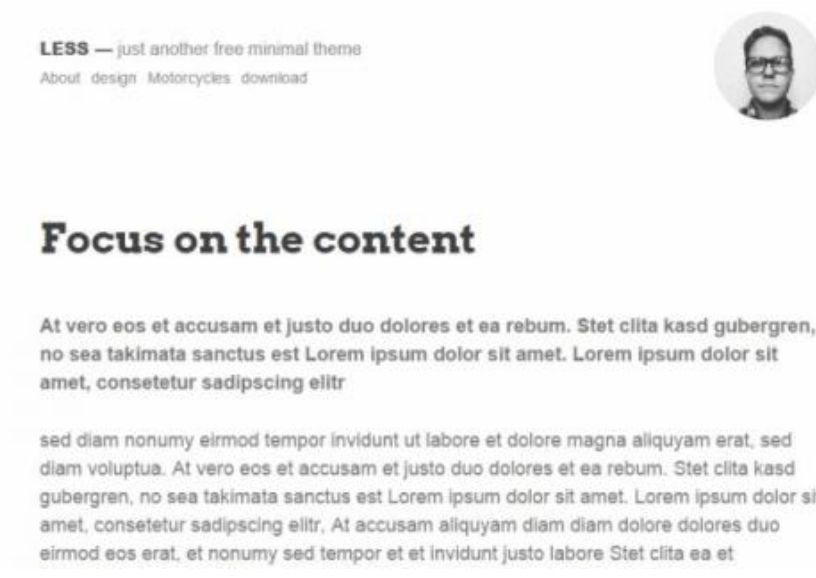
Τα θέματα δίνουν την δυνατότητα στους χρήστες του WordPress να αλλάζουν την εμφάνιση και χρηστικότητα ενός ιστότοπου, χωρίς να επηρεαστεί το περιεχόμενο. Μπορεί να αλλάξει εντελώς την παρουσίαση και μορφοποίηση των ιστοσελίδων με ενοποιημένο τρόπο και συνήθως επιτρέπει την παραμετροποίηση του ως ένα βαθμό χωρίς την ανάγκη παρέμβασης στον κώδικα. Εκτός από τα απαραίτητα αρχεία κώδικα σε PHP, μπορούν να περιλαμβάνουν αρχεία εικόνων, CSS και Javascript.

Στο Διαδίκτυο υπάρχει μια τεράστια ποικιλία διαθέσιμων θεμάτων (δωρεάν ή επί πληρωμή). Από τα πιο σχεδιαστικά απλά, μέχρι τα πιο σύνθετα με εντυπωσιακά εφέ και υψηλή διαδραστικότητα. Η επιλογή του κατάλληλου εξαρτάται από τις ανάγκες του χρήστη και τον σκοπό του ιστότοπου στον οποίο εφαρμόζεται. Για παράδειγμα αν αυτός ο σκοπός είναι η παρουσίαση φωτογραφιών, είναι προφανές πως ένα θέμα που δίνει μεγάλη έμφαση στις εικόνες, θα είναι το πιο κατάλληλο.



Εικόνα 2.17: Θέμα με έμφαση στις εικόνες.

Αντίθετα αν ο σκοπός του ιστότοπου είναι η εύκολη και γρήγορη πρόσβαση σε κάποιες πληροφορίες, η επιλογή του θέματος θα γίνει με γνώμονα την έμφαση στο γραπτό περιεχόμενο, χωρίς την ύπαρξη περιττών εφέ που μπορεί να εντυπωσιάζουν, αλλά εν τέλει αποσπούν τον επισκέπτη.



Εικόνα 2.18: Θέμα με έμφαση στο κείμενο.

Φυσικά υπάρχουν πολλά ακόμα κριτήρια για να επιλέξει κανείς. Κάθε περίπτωση είναι μοναδική και παίζει μεγάλο ρόλο και η υποκειμενική γνώμη του καθενός.

2.4.4 ΠΡΟΣΘΕΤΑ (PLUGINS)

Τα πρόσθετα είναι τα εργαλεία με τα οποία μπορούμε να επεκτείνουμε τις δυνατότητες και την λειτουργικότητα του WordPress. Ένα πρόσθετο μπορεί να είναι από κάτι απλό που να βοηθά στον εντοπισμό σχολίων spam, μέχρι κάτι που να προσθέτει λειτουργίες ηλεκτρονικού εμπορίου (e-commerce) με όλα τα χαρακτηριστικά ασφαλείας και τις δυνατότητες αλληλεπίδρασης με τον χρήστη που αυτό συνεπάγεται.

Είναι προφανές λοιπόν πως τα πρόσθετα είναι υπεύθυνα για την τεράστια ευελιξία της πλατφόρμας και πως επιτρέπουν στον χρήστη να προσαρμόσει τον ιστότοπό του απολύτως στις ανάγκες του, όποιες και να είναι αυτές. Υπάρχουν χιλιάδες δωρεάν και επί πληρωμή πρόσθετα διαθέσιμα στην επίσημη ιστοσελίδα του WordPress, και μπορούν να αναζητηθούν απευθείας από τον πίνακα ελέγχου.

2.4.5 WIDGETS

Widgets είναι μικρές εφαρμογές, συχνά μερικών γραμμών κώδικα μόνο, που χρησιμοποιούνται για την εμφάνιση περιεχομένου ή κάποιας λειτουργίας στις ακριανές μπάρες (sidebars) μιας σελίδας. Το WordPress έχει κάποια ήδη προεγκατεστημένα, όπως για την εμφάνιση κατηγοριών δημοσιεύσεων, λειτουργία αναζήτησης κ.α. Πολλές φορές κάποιο plugin μπορεί να προσθέσει ένα δικό του widget.

Είχαν σχεδιαστεί αρχικά για να παρέχουν έναν απλό και εύκολο τρόπο για τον έλεγχο της δομής και του σχεδιασμού ενός θέματος, καθώς δεν απαιτούν ειδικές γνώσεις για την χρήση τους. Μερικά widgets δίνουν δυνατότητες προσαρμογής τους και επιλογές για το τι ακριβώς θέλουμε να εμφανίσουν και τι όχι.



Εικόνα 2.19: Widget ημερολογίου και οι επιλογές του στον πίνακα ελέγχου.

2.4.6 ΔΟΜΗ ΜΙΑΣ ΣΕΛΙΔΑΣ WORDPRESS

Το WordPress χρησιμοποιεί αρχεία προτύπων (template files) για να συνθέσει την εμφάνιση και το περιεχόμενο μιας σελίδας. Δηλαδή όταν ο επισκέπτης βρίσκεται για παράδειγμα στην αρχική σελίδα του ιστότοπου, αυτό που βλέπει είναι το αποτέλεσμα πολλών template αρχείων PHP, τα οποία συνδυάζονται για να την δημιουργήσουν δυναμικά.



Εικόνα 2.17: Σύνθεση της αρχικής σελίδα index.php.

Η βασική δομή μιας σελίδας WordPress αποτελείται τουλάχιστον από τα παρακάτω στοιχεία:

- **Header (Κεφαλίδα) – header.php:**
Παραδοσιακά βρίσκεται στην κορυφή της σελίδας και περιέχει, μεταξύ άλλων, τον τίτλο της.
- **Sidebar (Ακριανή μπάρα) – sidebar.php:**
Όπως λέει και το όνομά του, βρίσκεται στα πλαϊνά της σελίδας και συνήθως περιέχει widgets. Αξίζει να σημειωθεί πως πολλά θέματα χρησιμοποιούν πάνω του ενός sidebar.
- **Content (Περιεχόμενο) – index.php / single.php / page.php κλπ.:**
Αποτελεί το σημαντικότερο μέρος διότι περιέχει το λεγόμενο WordPress Loop, το οποίο είναι ένα κομμάτι PHP κώδικα που καθορίζει το δυναμικό περιεχόμενο της σελίδας.
- **Footer (Υποσέλιδο) – sidebar.php:**
Βρίσκεται στο κάτω μέρος της σελίδας και συνήθως περιέχει κάποιες γενικότερες πληροφορίες σχετικά με τον ιστότοπο.

2.4.7 ΒΑΣΙΚΟΙ ΟΡΟΙ ΤΟΥ WORDPRESS

Σε αυτό το σημείο θα εξηγήσουμε εν συντομία μερικούς από τους βασικούς όρους του WordPress που θα συναντήσουμε κατά την υλοποίηση του ιστότοπου:

Post Types

Με τον όρο αυτό αναφερόμαστε στους 5 τύπους των δομών περιεχομένου που χρησιμοποιεί το WordPress: posts (δημοσιεύσεις), pages (σελίδες), attachments (επισυνάψεις), navigation menus (μενού πλοήγησης) και revisions (αναθεωρήσεις).

Custom Post Types

Εκτός από τους παραπάνω 5 τύπους, έχουμε την δυνατότητα να δημιουργήσουμε και δικούς μας προσαρμοσμένους τύπους (custom post types), οι οποίοι θα έχουν τα δικά τους χαρακτηριστικά και ελέγχονται από τον πίνακα ελέγχου.

Taxonomy

Τα 'Taxonomies' είναι ένας τρόπος ομαδοποίησης των στοιχείων που αφορούν ή/και περιγράφουν posts, links ή custom post types. Υπάρχουν 4 ενσωματωμένα taxonomies: category, post_tag, post_format και link_category. Το WordPress επιτρέπει επίσης και την δημιουργία προσαρμοσμένων taxonomies (custom_taxonomies) .

WordPress Loop

Ο κώδικας που χρησιμοποιεί το WordPress για να μπορεί να εμφανίζει τα posts types. Είναι ένας βρόχος επανάληψης που βρίσκεται στο αρχείο index.php και στα πρότυπα αρχεία (template files) που μπορεί να χρησιμοποιούν σελίδες του ιστότοπου. Η γενική του μορφή είναι η εξής:

```
<?php if ( have_posts() ) : ?>
    <?php while ( have_posts() ) : the_post(); ?>
        <!-- ΠΕΡΙΕΧΟΜΕΝΟ -->
    <?php endwhile; ?>
<?php endif; ?>
```

Η δομή *if* ελέγχει αν υπάρχουν 'posts' και για όσο αυτό ισχύει εκτελούνται οι εντολές εντός του βρόχου επανάληψης *while*.

Template Files

Τα 'template files' (πρότυπα αρχεία) χρησιμοποιούνται για να μπορούμε να αλλάζουμε και να προσαρμόζουμε ξεχωριστά την δομή και εμφάνιση των σελίδων μας σύμφωνα με τις ανάγκες μας.

ΚΕΦΑΛΑΙΟ 3: ΥΛΟΠΟΙΗΣΗ ΙΣΤΟΤΟΠΟΥ

3.1 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ

3.1.1 ΣΚΟΠΟΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Πριν ξεκινήσουμε το κομμάτι της υλοποίησης, θα πρέπει να έχουμε ξεκαθαρίσει επακριβώς τον σκοπό του ιστότοπου, καθώς και τις λειτουργίες που θα πρέπει να περιλαμβάνει.

Σκοπός: Η δημιουργία ενός ιστότοπου όπου θα παρουσιάζεται η ερευνητική δραστηριότητα του τμήματος Μηχανικών Η/Υ Συστημάτων και οι πτυχιικές εργασίες των φοιτητών του.

Κύριες λειτουργίες:

- Δυνατότητα ανάρτησης ανακοινώσεων.
- Δυνατότητα προσθήκης και επεξεργασίας ερευνητικών προγραμμάτων.
- Δυνατότητα προσθήκης, επεξεργασίας και αναζήτησης εκπονημένων πτυχιικών εργασιών.
- Δυνατότητα ανάρτησης νέων θεμάτων πτυχιικών εργασιών και αυτόματη δημιουργία σχετικής ανακοίνωσης.

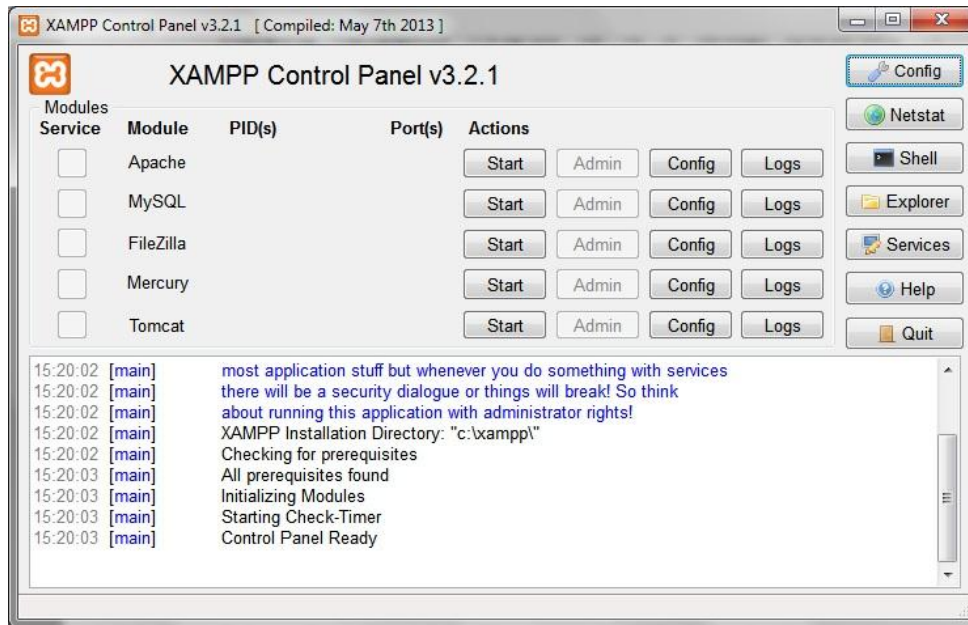
3.1.2 ΕΡΓΑΛΕΙΑ

Για την ανάπτυξη του ιστότοπου θα χρησιμοποιήσουμε τα παρακάτω ελεύθερα εργαλεία:

ΧΑΜΡΡ

Τα αρχικά της ονομασίας του προέρχονται από: **X** (αναφέρεται ως Cross-platform), **A**pache HTTP server, **M**ySQL/MariaDB, **P**HP, **P**earl.

Πρόκειται για ένα ελεύθερο και open source πακέτο λογισμικού διακομιστή ιστού, το οποίο αποτελείται κυρίως από τον Apache HTTP server, την MySQL (ή εναλλακτικά MariaDB) και τους διερμηνείς των γλωσσών PHP και Perl.

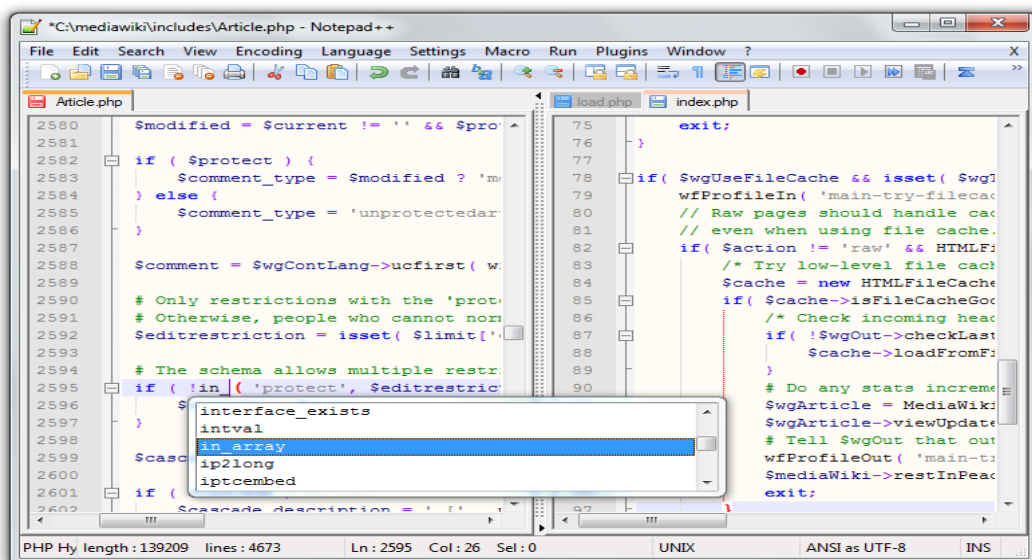


Εικόνα 3.1: Ο πίνακας ελέγχου του XAMPP.

Περιλαμβάνει επίσης και κάποια επιπλέον εργαλεία όπως το phpMyAdmin που χρησιμοποιείται για την εύκολη διαχείριση της βάσης δεδομένων, το FileZilla το οποίο είναι λογισμικό FTP (File Transfer Protocol) κ.α.

Notepad++

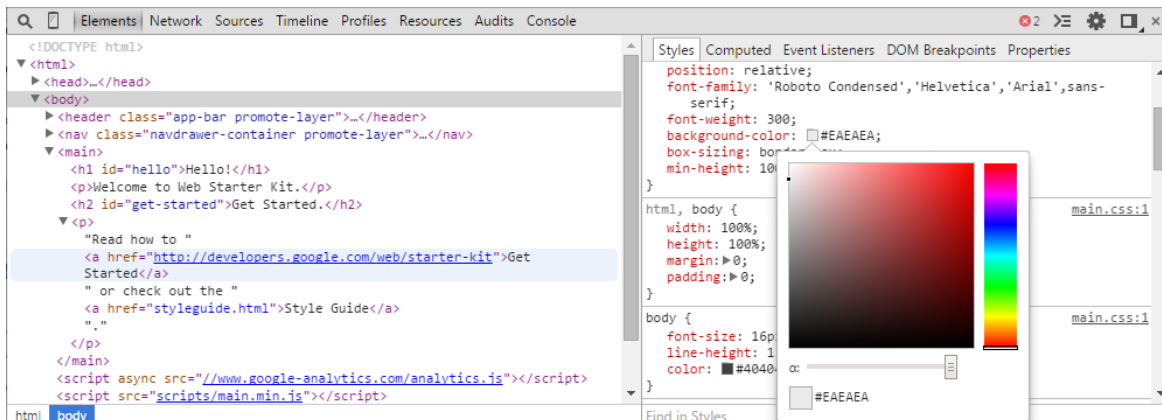
Πρόκειται για έναν επεξεργαστή κειμένου και πηγαίου κώδικα για Windows. Είναι γραμμένο σε C++ και περισσότερες από 50 γλώσσες προγραμματισμού, σεναριακού προγραμματισμού και σήμανσης.



Εικόνα 3.2: Παράθυρο επεξεργασίας του Notepad++.

Chrome Developer Tools

Πρόκειται για μια εργαλειοθήκη επεξεργασίας και απασφαλμάτωσης (debugging) ιστοσελίδων, η οποία είναι ενσωματωμένη στον φυλλομετρητή Google Chrome. Μας δίνει πρόσβαση στις εσωτερικές διεργασίες επικοινωνίας του browser με τον server αλλά και στον παραγόμενο κώδικά των ιστοσελίδων, με την δυνατότητα να για 'live' προσωρινών τροποποιήσεων στη σελίδα.

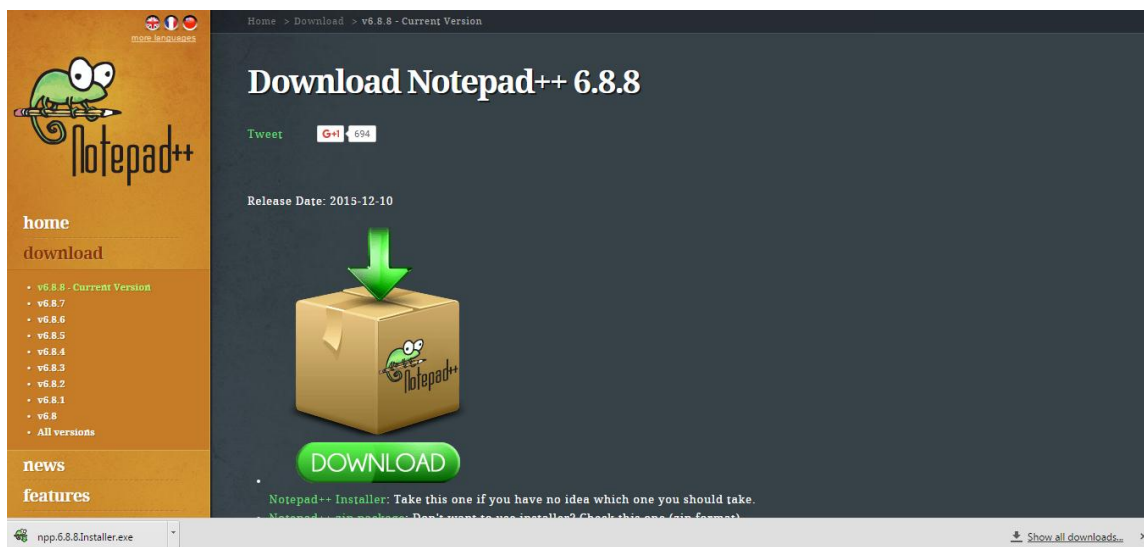


Εικόνα 3.3: Παράθυρο Chrome DevTools.

3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ

3.2.1 ΕΓΚΑΤΑΣΤΑΣΗ ΝΟΤΕΡΑΔ++

Αρχικά κάνουμε λήψη του εκτελέσιμου αρχείου εγκατάστασης του Notepad++ από την σελίδα <https://notepad-plus-plus.org/download/> :



Εικόνα 3.4: Λήψη του Notepad++.

Στην συνέχεια τρέχουμε το αρχείο και πραγματοποιούμε την εγκατάσταση του λογισμικού



Εικόνα 3.5: Εγκατάσταση του Notepad++.

3.2.2 ΕΓΚΑΤΑΣΤΑΣΗ ΧΑΜΡΡ

Πρώτα πραγματοποιούμε λήψη του αρχείου εγκατάστασης του ΧΑΜΡΡ από την σελίδα <https://www.apachefriends.org/> κάνοντας κλικ στον σύνδεσμο με την έκδοση για Windows :

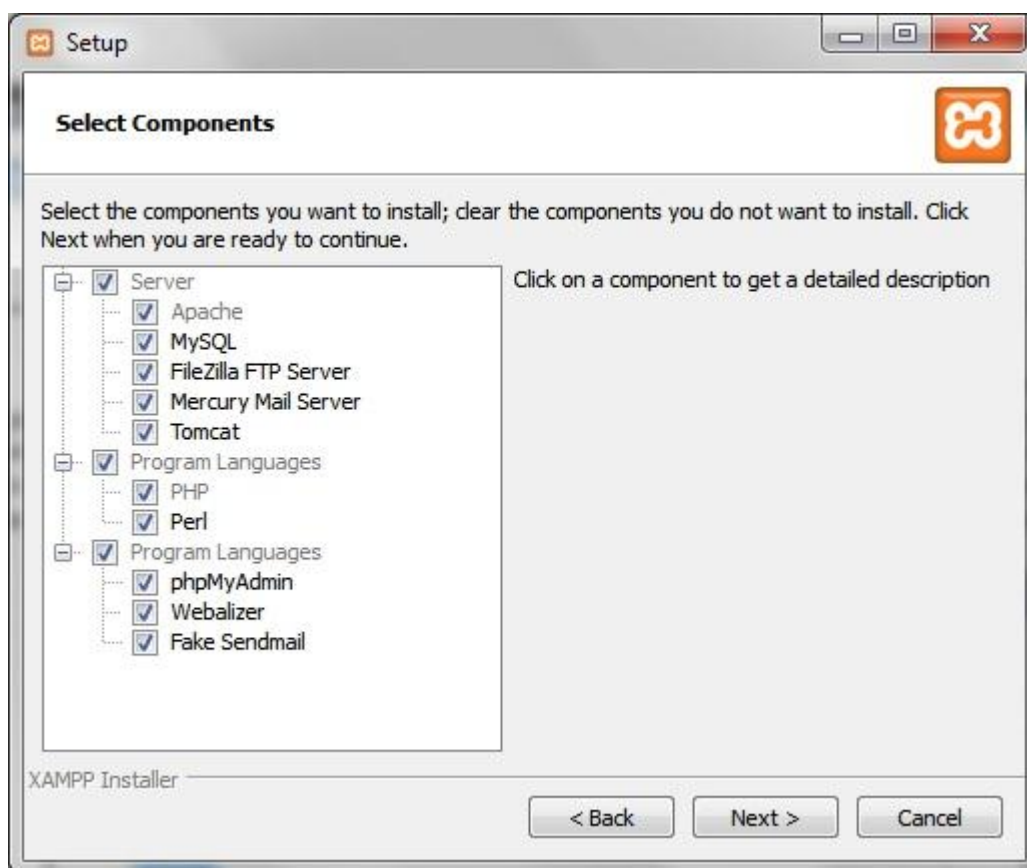


Εικόνα 3.7: Λήψη του ΧΑΜΡΡ.

Αφού ολοκληρωθεί η λήψη, τρέχουμε το αρχείο εγκατάστασης. Κατά την διάρκεια της μας ζητείται η επιλογή των στοιχείων που θέλουμε να εγκαταστήσουμε. Για την ανάπτυξη του ιστότοπου της παρούσας εργασίας, θα χρησιμοποιηθούν τα εξής:

- Apache
- MySQL
- PHP
- phpMyAdmin

Στην προκειμένη περίπτωση όμως θα εγκαταστήσουμε όλα τα στοιχεία καθώς ο επιπλέον χώρος που απαιτείται είναι αμελητέος και με αυτόν τον τρόπο θα μπορούμε πιο εύκολα να καλύψουμε τυχόν ανάγκες που μπορεί να προκύψουν και ίσως να μην έχουμε αρχικά προβλέψει (π.χ. χρήση FTP για την εγκατάσταση της ιστοσελίδας σε live server που θα επιτρέπει πρόσβαση μέσω του Διαδικτύου).



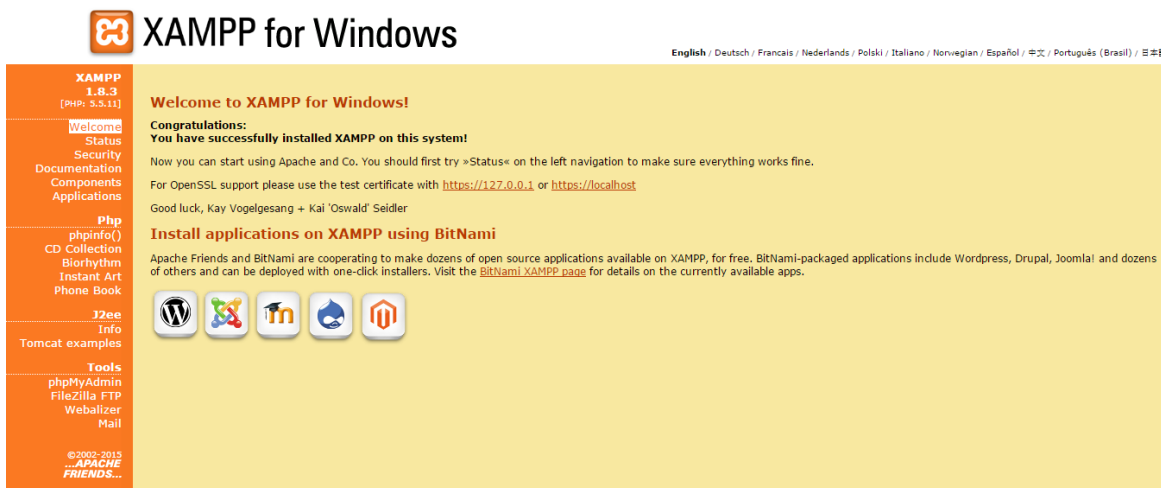
Εικόνα 3.8: Εγκατάσταση του XAMPP.

Αφού ολοκληρωθεί η εγκατάσταση, εκτελούμε το πρόγραμμα και εμφανίζεται ο πίνακας ελέγχου (εικόνα 3.1) όπου μπορούμε να δούμε τις διαθέσιμες υπηρεσίες. Για να θέσουμε σε λειτουργία τον τοπικό εξυπηρετητή (local host) που θα χρησιμοποιήσουμε για την ανάπτυξη του ιστότοπου, κάνουμε κλικ στο κουμπί Start του Apache και της βάσης δεδομένων MySQL. Αν δεν υπάρξει κάποιο πρόβλημα, θα δούμε τις αντίστοιχες ετικέτες τα γίνονται πράσινες, που σημαίνει πως οι υπηρεσίες εκτελούνται.

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	4352 3644	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	4192	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Εικόνα 3.9: Εκτέλεση των υπηρεσιών Apache HTTP και MySQL.

Αν λοιπόν πλοηγηθούμε με έναν browser στην τοπική διεύθυνση <https://localhost/> ή <https://127.0.0.1/> θα εμφανιστεί το παράθυρο που βλέπουμε στην εικόνα 3.10:



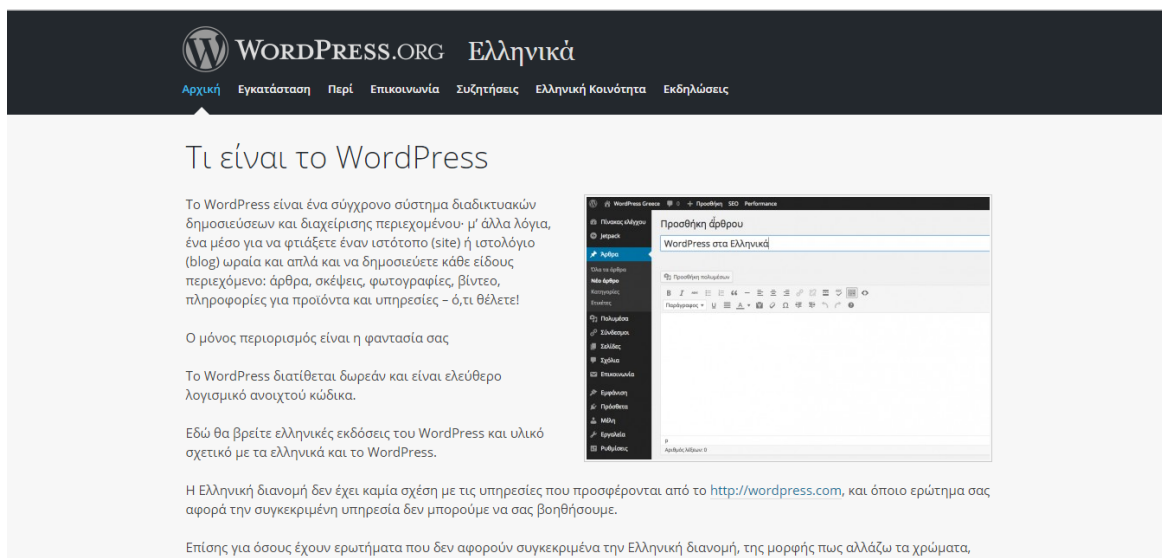
Εικόνα 3.10: Dashboard του XAMPP (<http://localhost/xampp>).

Από δω μας δίνονται αρκετές επιλογές, όπως η απευθείας εγκατάσταση κάποιου CMS διαμέσου του XAMPP, η πρόσβαση σε εργαλεία διαχείρισης (phpMyAdmin, FileZilla κλπ.), η προβολή της κατάστασης των διαφόρων υπηρεσιών αλλά και μια σειρά από documentations.

Σημείωση: Κατά την διάρκεια της υλοποίησης αυτής της εργασίας χρησιμοποιήθηκε η έκδοση 1.8.3 του XAMPP. Η εμφάνιση και μορφή του dashboard σε νεότερες εκδόσεις μπορεί να διαφέρει.

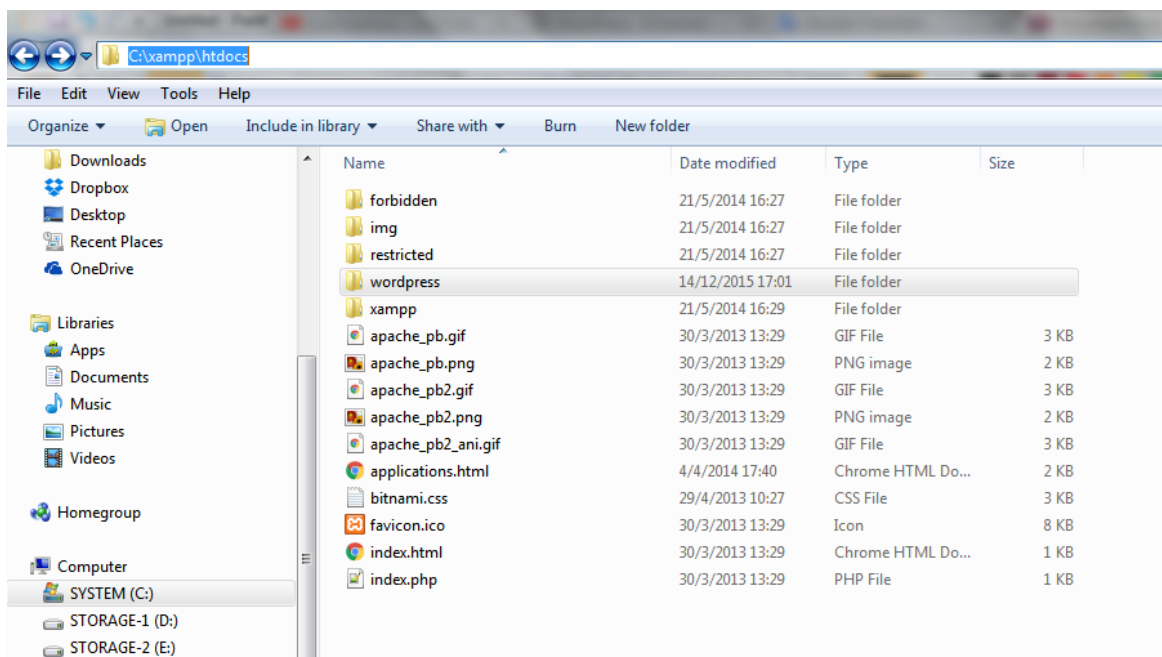
3.2.3 ΕΓΚΑΤΑΣΤΑΣΗ WORDPRESS

Για να εγκαταστήσουμε το WordPress στον τοπικό μας server θα πρέπει πρώτα να το κατεβάσουμε από το <https://wordpress.org>



Εικόνα 3.11: Σελίδα της ελληνικής έκδοσης του WordPress.

Το αρχείο που θα λάβουμε δεν θα είναι κάποιο εκτελέσιμο, αλλά ένας φάκελος σε συμπιεσμένη μορφή zip, ο οποίος περιέχει όλα τα απαραίτητα αρχεία μιας εγκατάστασης WordPress. Τα περιεχόμενα του συμπιεσμένου αυτού αρχείου θα τα εξάγουμε σε έναν φάκελο που θα ονομάσουμε 'wordpress' στον φάκελο 'htdocs' του XAMPP, ο οποίος είναι το *web root directory* του server μας.



Εικόνα 3.12: Web root directory του τοπικού server.

Με τον όρο *web root directory* εννοούμε τον φάκελο στον οποίο αποθηκεύονται όλες οι ιστοσελίδες στον server. Είναι δηλαδή ο φάκελος στον οποίο θα έχει πρόσβαση κάποιος όταν πλοηγείται στην διεύθυνση του συγκεκριμένου ιστότοπου. Διαφέρει από εξυπηρετητή σε εξυπηρετητή

Για να συνεχίσουμε την εγκατάσταση εισάγουμε στον browser την διεύθυνση <http://localhost/wordpress/> και αυτόματα θα φορτωθεί το αρχείο `setup-config.php` το οποίο θα μας καθοδηγήσει για την ολοκλήρωση της διαδικασίας.

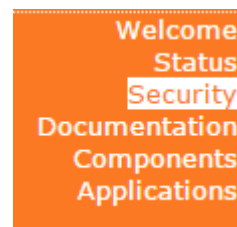


Εικόνα 3.13: Το πρώτο βήμα της εγκατάστασης του WordPress.

Σε αυτό το σημείο ο οδηγός εγκατάστασης μας ενημερώνει πως θα χρειαστεί το όνομα, τον χρήστη (user), το συνθηματικό (password) και τον φιλοξενητή (host) της βάσης δεδομένων.

Το προεπιλεγμένο όνομα χρήστη μιας νέας εγκατάστασης MySQL είναι 'root' και το συνθηματικό αν δεν οριστεί, θα είναι κενό. Στην προκειμένη περίπτωση, επειδή η εγκατάστασή μας είναι τοπική με μοναδικό σκοπό την ανάπτυξη του ιστότοπου, δεν είναι απαραίτητο αυτά να αλλάξουν. Σε κάθε άλλη περίπτωση όμως θα ήταν ιδιαίτερα σημαντικό για την ασφάλεια της βάσης.

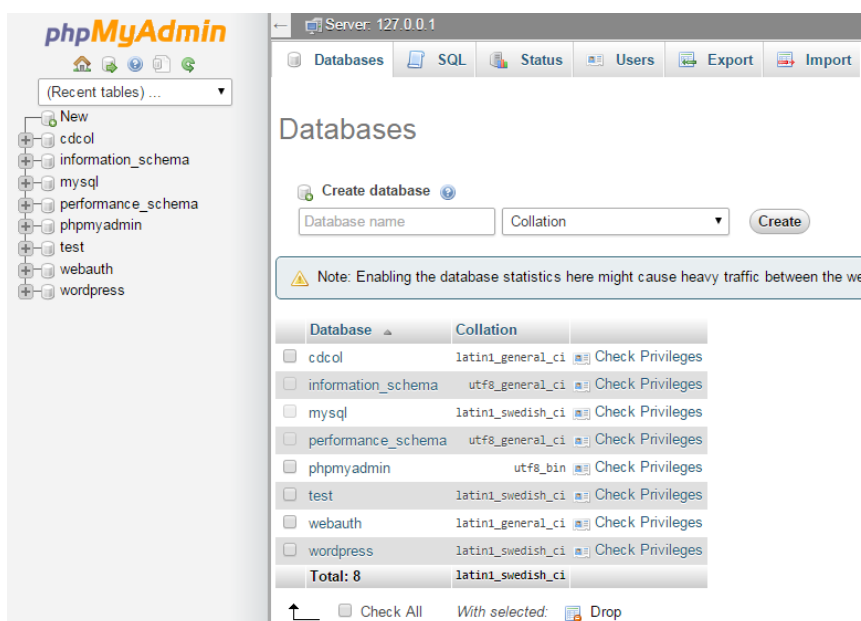
Αν θέλαμε να ορίσουμε password θα πηγαίναμε στο root directory <http://localhost/xampp> (εικόνα 3.10) και θα κάναμε κλικ στην επιλογή 'Security' του μενού που βρίσκεται στα αριστερά. Όπως βλέπουμε παρακάτω, η επόμενη οθόνη θα είχε τα πεδία που πρέπει να συμπληρωθούν για την δημιουργία νέου συνθηματικού.



Εικόνα 3.14: Security

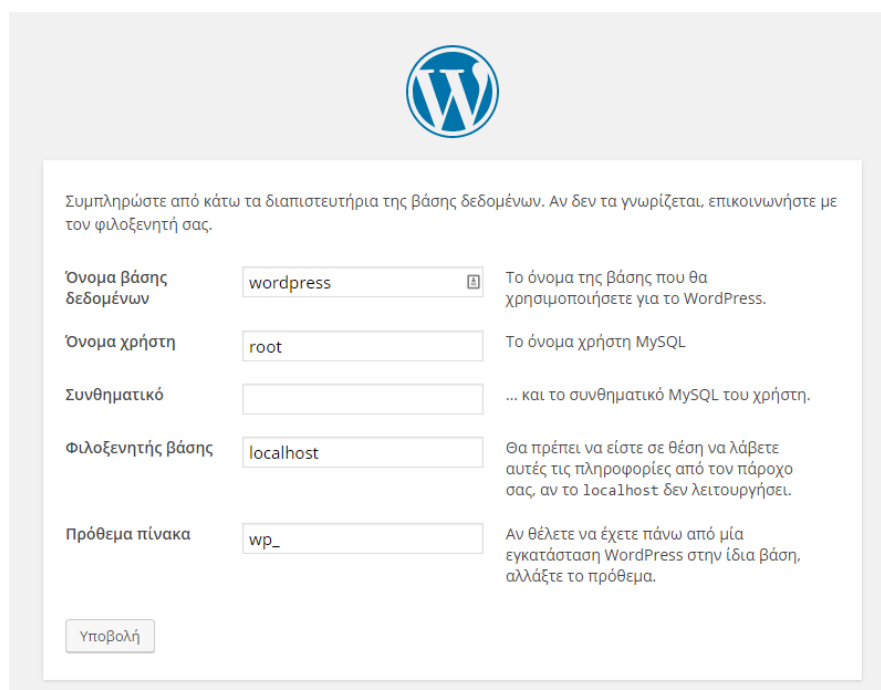
Εικόνα 3.15: Ορισμός MySQL password.

Στην συνέχεια θα πρέπει να δημιουργήσουμε την βάση δεδομένων που θα χρησιμοποιήσουμε. Για να το κάνουμε αυτό θα χρησιμοποιήσουμε το εργαλείο διαχείρισης *phpMyAdmin*. Μπορούμε να το επιλέξουμε από το ίδιο μενού ή να πλοηγηθούμε απευθείας στο <http://localhost/phpmyadmin> από την γραμμή διευθύνσεων του browser. Αφού συμπληρώσουμε το όνομα χρήστη και το συνθηματικό που ορίσαμε προηγουμένως (root και κενό αφού τα αφήσαμε ως έχουν αυτή τη φορά), πηγαίνουμε στην καρτέλα 'Databases' και πληκτρολογούμε το όνομα της βάσης που θέλουμε να δημιουργήσουμε. Στην περίπτωση μας θα βάλουμε ως όνομα το 'wordpress'.



Εικόνα 3.15: Δημιουργία βάσης δεδομένων.

Έχοντας δημιουργήσει και την βάση μας, είμαστε έτοιμοι να συνεχίσουμε την εγκατάσταση του WordPress. Συμπληρώνουμε τα στοιχεία και πατάμε 'Υποβολή'.




The screenshot shows the WordPress database configuration screen. At the top is the WordPress logo. Below it, a text box says: "Συμπληρώστε από κάτω τα διαπιστευτήρια της βάσης δεδομένων. Αν δεν τα γνωρίζεται, επικοινωνήστε με τον φιλοξενητή σας." Below this are five rows of input fields with labels and instructions:

Όνομα βάσης δεδομένων	<input type="text" value="wordpress"/>	Το όνομα της βάσης που θα χρησιμοποιήσετε για το WordPress.
Όνομα χρήστη	<input type="text" value="root"/>	Το όνομα χρήστη MySQL
Συνθηματικό	<input type="text"/>	... και το συνθηματικό MySQL του χρήστη.
Φιλοξενητής βάσης	<input type="text" value="localhost"/>	Θα πρέπει να είστε σε θέση να λάβετε αυτές τις πληροφορίες από τον πάροχο σας, αν το localhost δεν λειτουργήσει.
Πρόθεμα πίνακα	<input type="text" value="wp_"/>	Αν θέλετε να έχετε πάνω από μία εγκατάσταση WordPress στην ίδια βάση, αλλάξτε το πρόθεμα.

At the bottom left is a button labeled "Υποβολή".

Εικόνα 3.16: Συμπλήρωση στοιχείων βάσης.

Στη συνέχεια της εγκατάστασης θα χρειαστεί να συμπληρώσουμε τον τίτλο του ιστότοπου και τα στοιχεία του λογαριασμό διαχειριστή.



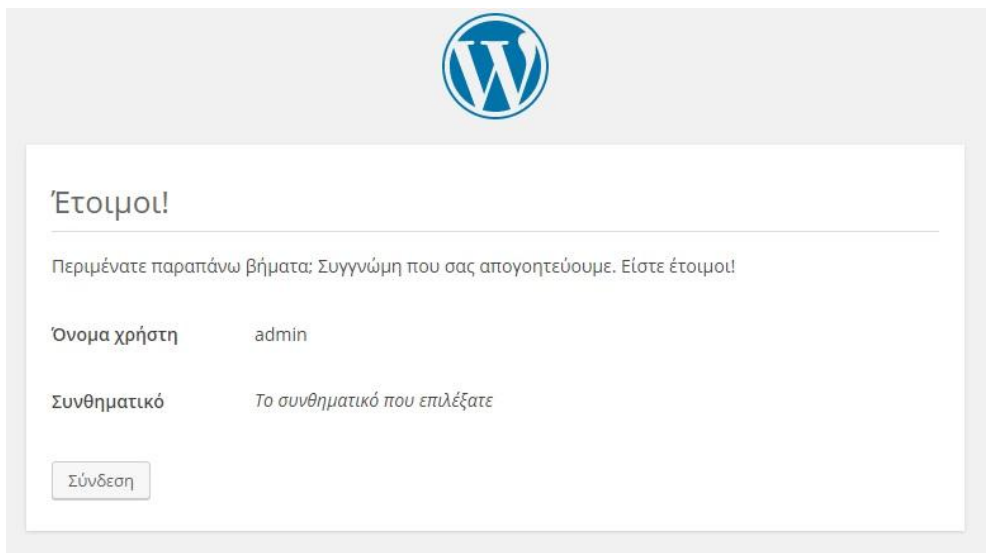
The screenshot shows the WordPress site title and admin user configuration screen. At the top is the WordPress logo. Below it, a heading says "Καλώς ήρθατε". Below the heading is a paragraph: "Καλώς ήρθατε στην διάσημη 5-λεπτη διαδικασία εγκατάστασης του WordPress! Μόνο συμπληρώστε τις πληροφορίες παρακάτω και είστε στο δρόμο να χρησιμοποιήσετε την πιο επεκτάσιμη και πανίσχυρη προσωπική πλατφόρμα δημοσιεύσεων στον κόσμο." Below this is a heading "Απαραίτητα στοιχεία". Below the heading is a text box: "Συμπληρώστε τα παρακάτω στοιχεία. (Γίνεται να τα αλλάξετε αργότερα.)". Below this are three rows of input fields with labels and instructions:

Τίτλος ιστότοπου	<input type="text"/>	
Όνομα χρήστη	<input type="text" value="admin"/>	*
Συνθηματικό	<input type="text" value="221k*xMImoyCx#8UjX"/>	<input type="button" value="Απόκρυψη"/>

Below the password field is a green button labeled "Δυνατό". Below this is a text box: "Προσοχή: Χρειάζεστε το συνθηματικό για σύνδεση. Παρακαλούμε αποθηκεύστε το σε μια ασφαλή τοποθεσία."

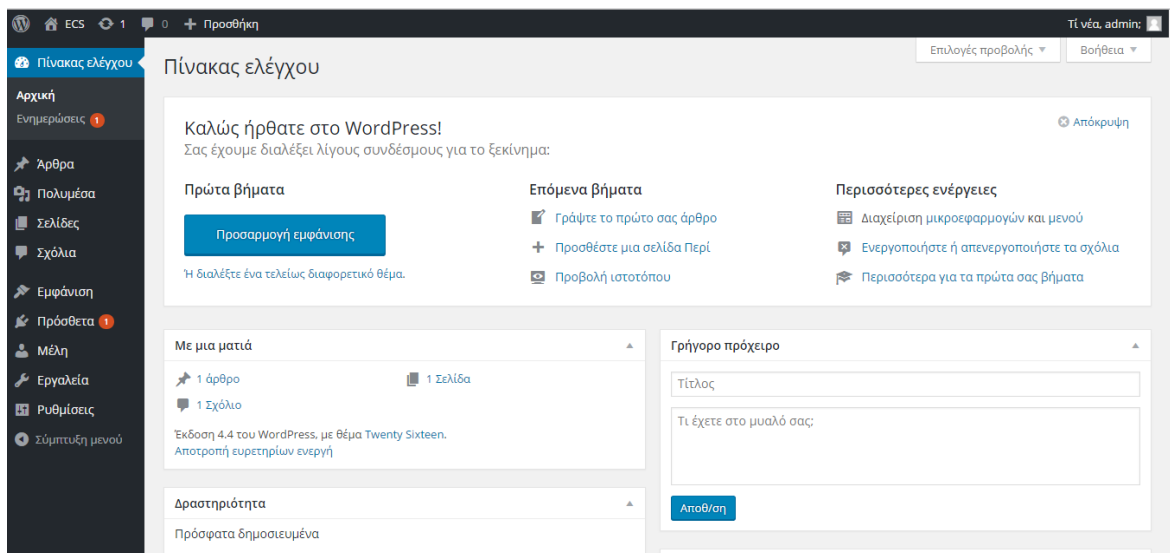
Εικόνα 3.17: Συμπλήρωση στοιχείων ιστότοπου (μπορούν να αλλάξουν αργότερα).

Μετά την ολοκλήρωση του τελευταίου βήματος, είμαστε πλέον έτοιμοι να συνδεθούμε στον νέο μας ιστότοπο!



Εικόνα 3.18: Ολοκλήρωση εγκατάστασης.

Το πρώτο πράγμα που θα αντικρίσουμε μόλις συνδεθούμε, είναι ο πίνακας ελέγχου μέσω του οποίου θα μπορέσουμε να διαχειριζόμαστε τον ιστότοπο και το περιεχόμενό του.

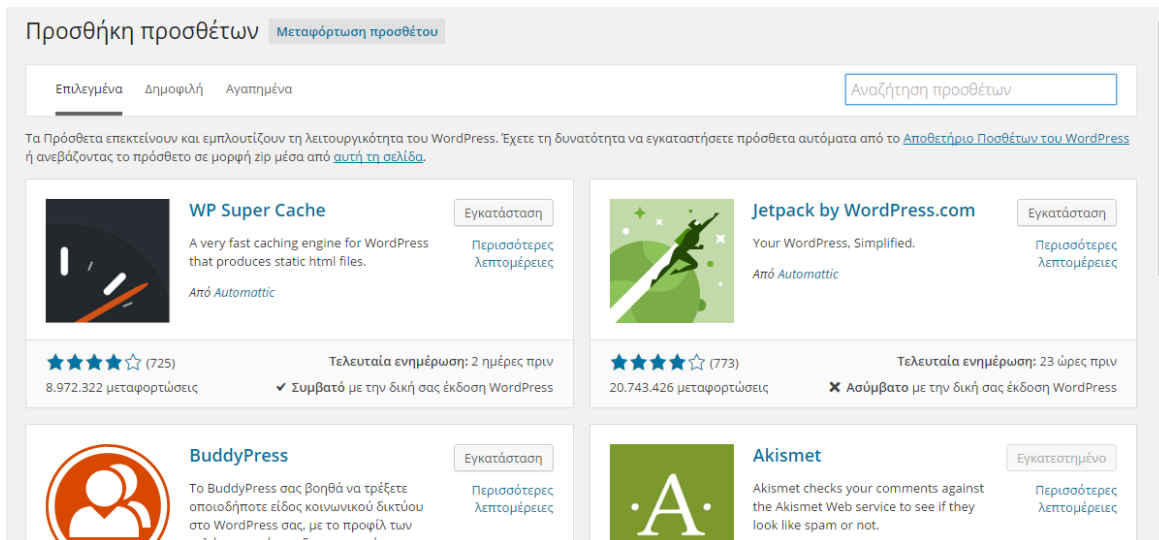


Εικόνα 3.19: Πίνακας Ελέγχου.

3.2.4 ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΣΘΕΤΩΝ

Σε αυτό το στάδιο θα εγκαταστήσουμε μια σειρά από πρόσθετα (plugins) τα οποία θα βοηθήσουν στην ανάπτυξη ορισμένων λειτουργιών, θα προσθέσουν κάποιες χρήσιμες λειτουργίες στον πίνακα ελέγχου και γενικά θα κάνουν πιο εύκολη την διαχείριση του ιστότοπου.

Αναζητούμε τα πρόσθετα που θέλουμε να εγκαταστήσουμε πηγαίνοντας στο Πρόσθετα→Νέο Πρόσθετο.



Εικόνα 3.20: Σελίδα αναζήτησης πρόσθετων.

Από εδώ αφού βρούμε αυτό που θέλουμε μέσω του πεδίου αναζήτησης, κάνουμε κλικ στο κουμπί 'Εγκατάσταση' και στη συνέχεια το ενεργοποιούμε.

Αναλυτικά θα έχουμε τα εξής plugins:

Custom Post Type UI:

Εργαλείο που επιτρέπει την δημιουργία προσαρμοσμένων τύπων δημοσιεύσεων (post types) και taxonomies και θα μας είναι απαραίτητο για την υλοποίηση ειδικών ειδών δημοσιεύσεων για τις πτυχιακές εργασίες και τα ερευνητικά προγράμματα.

Advanced Custom Fields:

Ένα πολύ δυνατό εργαλείο που επεκτείνει σημαντικά τις δυνατότητες του WordPress και μας επιτρέπει να δημιουργούμε τα δικά μας προσαρμοσμένα πεδία δεδομένων. Θα μας είναι απαραίτητο στην συνέχεια της υλοποίησης του ιστότοπου, όπου θα χρειαστεί να έχουμε επιπλέον πεδία εισαγωγής στοιχείων για τα νέα post types.

TinyMCE Advanced:

Προθέτει επιπλέον επιλογές και δυνατότητες στον επεξεργαστή κειμένου του WordPress διευκολύνοντας τους συγγραφείς στην διαμόρφωση του περιεχομένου των σελίδων και άρθρων.

Duplicator:

Εργαλείο για την δημιουργία backup κλώνων ολόκληρης της εγκατάστασης του WordPress και των περιεχομένων της βάσης δεδομένων.

Force Post Category:

Αναγκάζει τον συγγραφέα να επιλέγει κατηγορία για κάθε δημοσίευση για λόγους συνοχής.

Black Studio TinyMCE Widget:

Προσθέτει ένα νέο τύπο widget με τις δυνατότητες του ενσωματωμένου επεξεργαστή κειμένου του WordPress.

Dynamic Widgets:

Δίνει την δυνατότητα να ορίζουμε σε ποιες σελίδες και με υπό ποιες προϋποθέσεις θα εμφανίζονται συγκεκριμένα widgets.

Preserved HTML Editor Markup Plus:

Δίνει την δυνατότητα στον συγγραφέα ενός άρθρου να εναλλάσσεται μεταξύ του WYSIWYG επεξεργαστή κειμένου και του απλού επεξεργαστή, όπου μπορεί να χρησιμοποιεί HTML χωρίς να 'χαλάει' ο κώδικας.

List Custom Taxonomy Widget (Modified):

Widget που επιτρέπει την εμφάνιση κατηγοριών συνδεδεμένων με custom post types.

WP Google Maps:

Γρήγορος και εύκολος τρόπος για να προσθέτουμε έναν χάρτη της Google σε όποια σελίδα και άρθρο επιθυμούμε.

3.3 ΥΛΟΠΟΙΗΣΗ ΖΗΤΟΥΜΕΝΩΝ

3.3.1 ΕΠΙΛΟΓΗ ΘΕΜΑΤΟΣ

Η επιλογή του κατάλληλου θέματος είναι πολύ σημαντική αφού αυτό θα καθορίσει την συνολική εμφάνιση του ιστότοπου. Μπορούμε να δούμε και να ενεργοποιήσουμε κάποιο από τα προεπιλεγμένα θέματα του WordPress πηγαίνοντας στο Εμφάνιση → Θέματα του μενού. Εδώ μπορούμε επίσης να αναζητήσουμε και νέα θέματα πατώντας την επιλογή 'Προσθήκη Νέου'.

Για τον ιστότοπο της εργασίας επέλεξα το θέμα 'WP-Forge' το οποίο ως σχεδιαστικά μινιμαλιστικό, αποτελεί εξαιρετική βάση πάνω στην οποία μπορούμε να κατασκευάσουμε τις σελίδες μας. Η απλότητα της σχεδίασής του ταιριάζει απόλυτα στο ύφος και στυλ ενός ιστότοπου ακαδημαϊκού ιδρύματος στο οποίο θέλουμε να είναι στο προσκήνιο το περιεχόμενο, στην προκειμένη περίπτωση η ερευνητική δραστηριότητα και οι πτυχιακές εργασίες. Επιπλέον μας δίνει την απαραίτητη ευελιξία να το φέρουμε στα μέτρα μας κάνοντας τροποποιήσεις ή προσθέτοντας επιπλέον λειτουργίες.



Εικόνα 3.21: Θέμα WP-Forge.

Εφόσον κατεβάσουμε λοιπόν το θέμα από το Διαδίκτυο και όχι μέσω του πίνακα ελέγχου, θα πρέπει να το εγκαταστήσουμε χειροκίνητα. Το αρχείο βρίσκεται σε συμπιεσμένη μορφή zip και πρέπει να εξάγουμε το περιεχόμενό του στο `\wordpress\wp-content\themes`. Αφού γίνει αυτό, το θέμα θα εμφανιστεί και στον πίνακα ελέγχου.

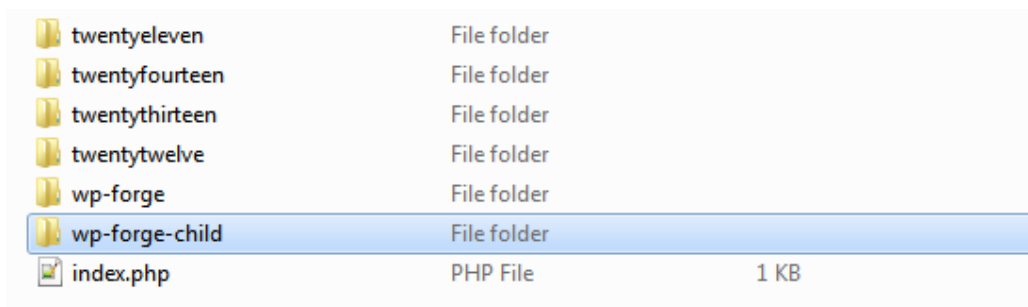
3.3.2 ΔΗΜΙΟΥΡΓΙΑ CHILD THEME

Αν και το WordPress μας δίνει αρκετές επιλογές τροποποίησης των θεμάτων μέσω του πίνακα ελέγχου, για να έχουμε τον απόλυτο έλεγχο θα πρέπει να παρέμβουμε στον κώδικα του. Όμως όποιες παρεμβάσεις κάναμε απευθείας στα αρχεία του θα χανόντουσαν κάθε φορά που έβγαινε μια νέα αναβάθμιση για το θέμα αυτό, κάτι που φυσικά δεν το θέλουμε.

Γι' αυτόν τον λόγο θα δημιουργήσουμε ένα θέμα απόγονο (child theme), το οποίο θα κληρονομεί τα χαρακτηριστικά και την λειτουργικότητα του θέματος-γονέα, αλλά θα μπορεί ταυτόχρονα να παρακάμψει, αντικαταστήσει ή τροποποιήσει οποιοδήποτε από αυτά επιλέξουμε εμείς.

Βήματα:

- 1) Στον φάκελο που περιέχει τα θέματα (\wordpress\wp-content\themes) δημιουργούμε έναν νέο φάκελο. Είναι κοινή πρακτική να έχει το όνομα του γονέα με την προσθήκη '-child' στο τέλος.



Εικόνα 3.22: Θέμα απόγονος.

- 2) Μέσα στον νέο φάκελο, το μόνο αρχείο που είναι απολύτως απαραίτητο για να αναγνωρίσει το WordPress το θέμα απόγονο, είναι το styles.css και θα πρέπει στην αρχή του να περιέχει τον εξής κώδικα:

```
/*  
Theme Name: WP-Forge Child  
Template: wp-forge  
Description: WP-Forge child theme.  
Author: Adamidis  
Version: 0.1  
*/  
@import url('../wp-forge/style.css');
```

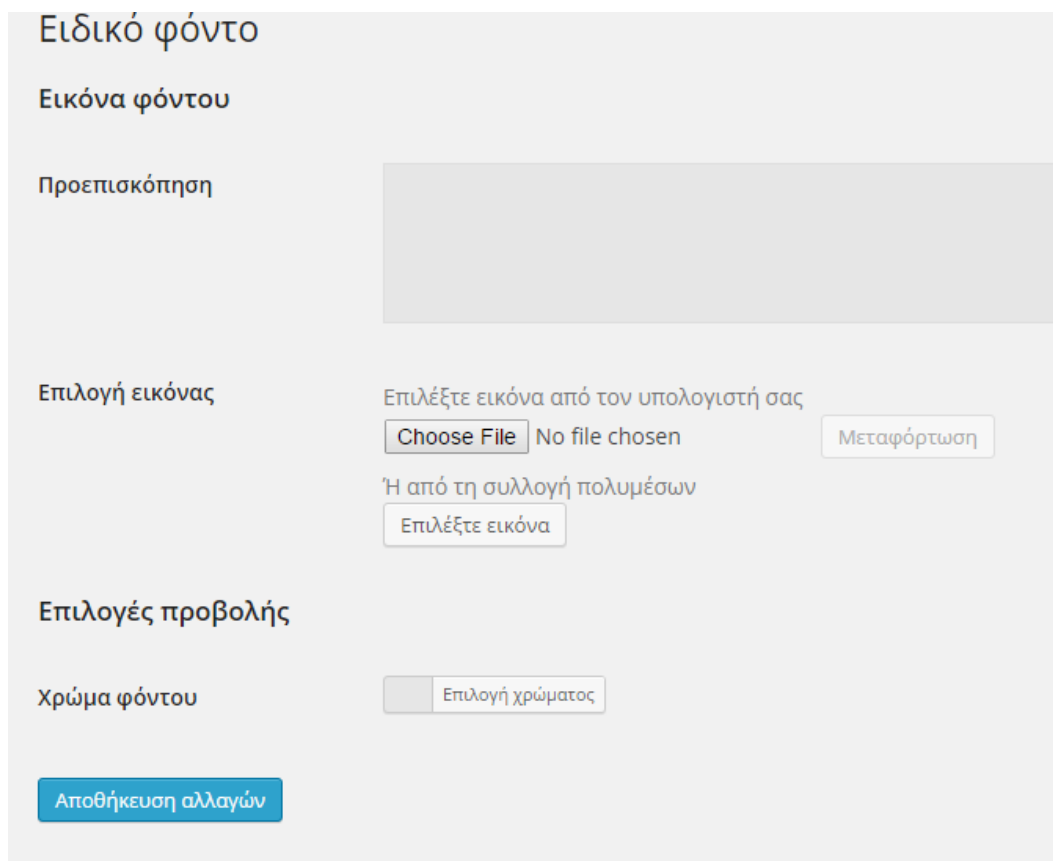
Τα απολύτως απαραίτητα στοιχεία είναι η ετικέτα 'Template' που προσδιορίζει το γονικό θέμα και η εντολή 'import' που εισάγει τον css κώδικά του. Τα υπόλοιπα είναι επιπλέον πληροφορίες που προσαρμόζουμε στα δικά μας δεδομένα, έχοντας την δυνατότητα να προσθέσουμε κι άλλα. Κάτω από αυτά θα μπουν οι δικές μας τροποποιήσεις στην εμφάνιση του θέματος.

Τώρα μπορούμε να ενεργοποιήσουμε το νέο μας θέμα απόγονο από τον πίνακα ελέγχου του WordPress. Επιπλέον όποια επιπλέον αρχεία με κώδικα δημιουργήσουμε από εδώ και πέρα για τον ιστότοπό μας θα το βάζουμε στον φάκελο του child theme που φτιάξαμε.

3.3.3 ΔΙΑΜΟΡΦΩΣΗ ΘΕΜΑΤΟΣ

Σε αυτό το στάδιο είμαστε έτοιμοι να διαμορφώσουμε το θέμα μας από την καρτέλα 'Εμφάνιση' του πίνακα ελέγχου.

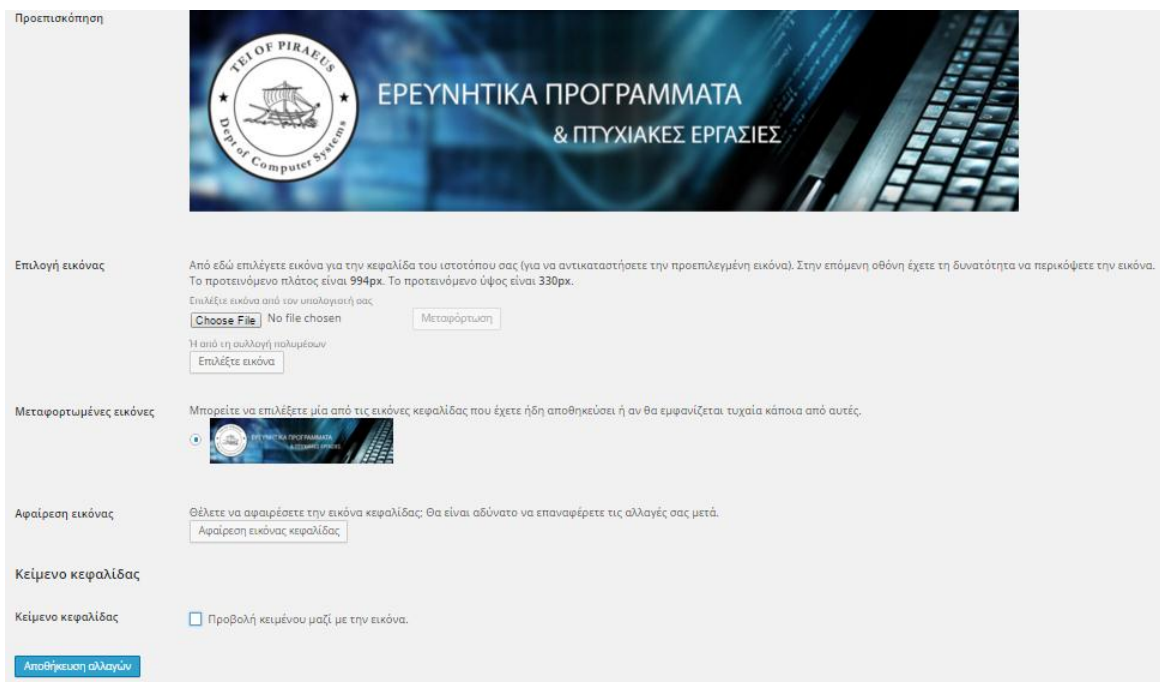
Η επιλογή 'Φόντο' μας επιτρέπει, όπως λέει όνομά της, να αλλάξουμε το χρώμα του φόντου (background) του ιστότοπου ή ακόμα και να βάλουμε μια εικόνα αν θέλουμε. Στο τέλος πατάμε 'Αποθήκευση αλλαγών'.



Εικόνα 3.23: Φόντο ιστότοπου.

Η επιλογή 'Κεφαλίδα' μας επιτρέπει να διαμορφώσουμε την εικόνα και το κείμενο που θέλουμε να εμφανίζεται στο πάνω μέρος του ιστότοπου. Μπορούμε φυσικά να επιλέξουμε να εμφανίζεται μόνο η εικόνα, το κείμενο ή και τα δύο μαζί. Στη δική μας περίπτωση επιλέγουμε μόνο μια εικόνα για μεταφόρτωση (η οποία περιέχει

και κείμενο) που την δημιουργήσαμε στο Photoshop ειδικά γι' αυτόν τον σκοπό. Στην συνέχεια πατάμε 'Αποθήκευση αλλαγών'.



Εικόνα 3.24: Κεφαλίδα ιστότοπου.

3.3.4 ΔΗΜΙΟΥΡΓΙΑ ΣΕΛΙΔΩΝ

Σε αυτό το στάδιο θα δημιουργήσουμε τις κύριες σελίδες του ιστότοπου οι οποίες θα αντιστοιχούν και στα στοιχεία που θα βρίσκονται στο κυρίως μενού πλοήγησης (Navigation menu). Σε πρώτο στάδιο τις φτιάχνουμε χωρίς περιεχόμενο, απλά για να έχουμε μια γενική ιδέα για την δομή του ιστότοπου και στη συνέχεια θα παρουσιαστούν αναλυτικά. Για να δημιουργήσουμε μια σελίδα πηγαίνουμε στο *Σελίδες → Προσθήκη νέας*.

Σημείωση: Κατά την διάρκεια της συγγραφής σελίδων και δημοσιεύσεων μέσω του επεξεργαστή κειμένου του πίνακα ελέγχου, έχουμε την δυνατότητα να δούμε πως θα φαίνεται το τελικό αποτέλεσμα οποιαδήποτε στιγμή πριν την τελική ενημέρωση/δημοσίευση πατώντας το κουμπί προεπισκόπησης:

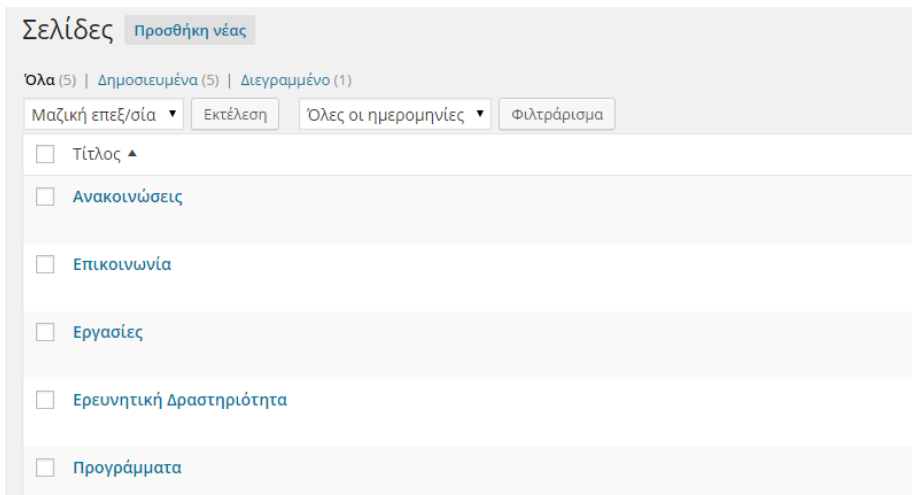
Προεπισκόπηση

Επίσης μπορούμε να αποθηκεύσουμε προσωρινά ένα προσχέδιο και να συνεχίσουμε την επεξεργασία του κάποια άλλη στιγμή πατώντας το κουμπί αποθήκευσης προσχεδίου:

Αποθήκευση προσχεδίου

Οι σελίδες και ο σκοπός τους θα είναι η εξής:

- Ερευνητική Δραστηριότητα
Θα περιέχει κάποιες γενικές πληροφορίες για την ερευνητική δραστηριότητα του τμήματος.
- Ανακοινώσεις
Εδώ θα δημοσιεύονται οι όποιες ανακοινώσεις σχετικές με ερευνητικά προγράμματα και πτυχιακές εργασίες.
- Προγράμματα
Θα περιέχει όλα τα ερευνητικά προγράμματα του τμήματος με τις βασικές τους πληροφορίες.
- Εργασίες
Θα περιέχει μια λίστα με τα διαθέσιμα θέματα πτυχιακών προς εκπόνηση καθώς και μια φόρμα αναζήτησης για ολοκληρωμένες πτυχιακές εργασίες.
- Επικοινωνία
Σελίδα με βασικά στοιχεία επικοινωνίας για την σχολή και το τμήμα.

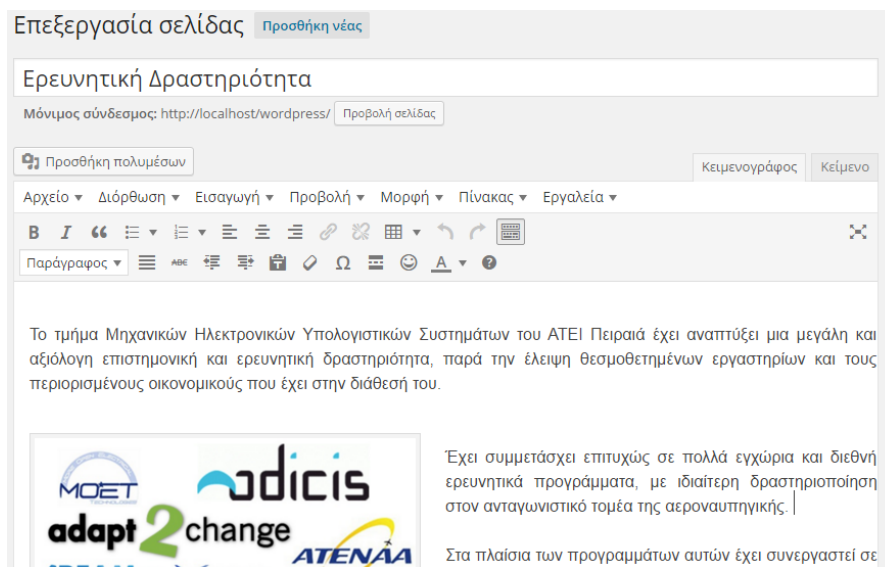


Εικόνα 3.25: Δημιουργία & επεξεργασία σελίδων.

Αξίζει να σημειωθεί πως οι ιστοσελίδες 'Προγράμματα' και 'Εργασίες' θα δημιουργούνται δυναμικά κατά με κώδικα που θα παρουσιάσουμε λίγο παρακάτω.

3.3.5 ΑΡΧΙΚΗ ΣΕΛΙΔΑ (ΕΡΕΥΝΗΤΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ)

Αυτή θα είναι μια απλή στατική σελίδα που θα παίζει τον ρόλο της αρχικής και θα περιέχει κάποιες γενικές πληροφορίες που θέλουμε να δει ο επισκέπτης με το εισέρχεται για πρώτη φορά στον ιστότοπο. Για να την επεξεργαστούμε κάνουμε κλικ πάνω στον τίτλο της (εικόνα 3.25) και χρησιμοποιούμε τον ενσωματωμένο επεξεργαστή κειμένου.



Εικόνα 3.26: Επεξεργασία σελίδας 'Ερευνητική Δραστηριότητα'.

Μόλις ολοκληρώσουμε την συγγραφή του κειμένου θα κάνουμε κλικ στην ετικέτα 'Ενημέρωση' για δημοσιεύσουμε το τελικό περιεχόμενο.

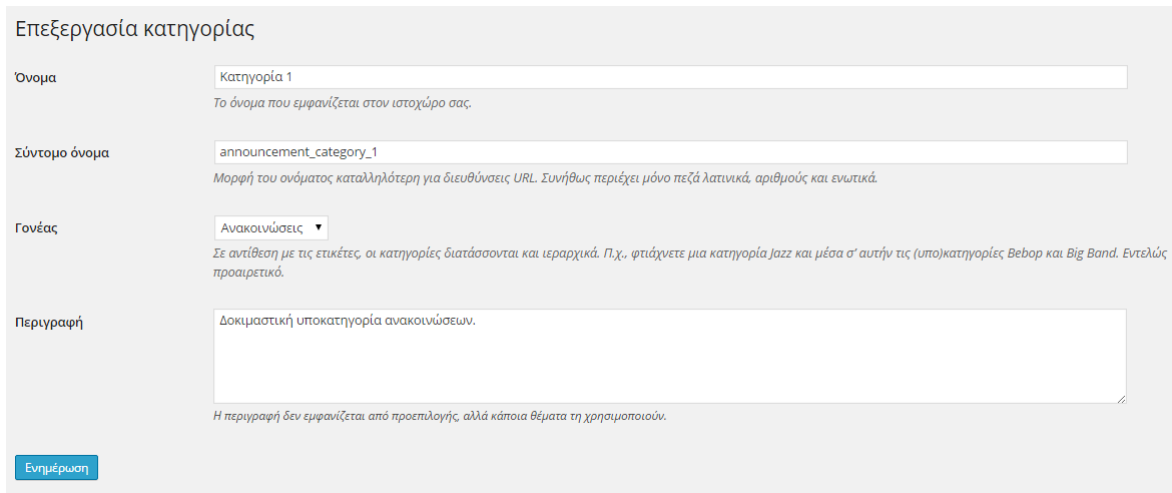
Στην συνέχεια θα πάμε στο Εμφάνιση → Προσαρμογή → Front Page Section, θα επιλέξουμε η αρχική μας σελίδα να είναι στατική και θα ορίσουμε ως πρώτη σελίδα την 'Ερευνητική Δραστηριότητα'. Ακριβώς από κάτω υπάρχει η επιλογή για τον ορισμό της σελίδας των άρθρων δηλαδή των ανακοινώσεων, οπότε θα επιλέξουμε την σελίδα 'Ανακοινώσεις' και πατάμε το κουμπί 'Αποθήκευση & δημοσίευση'.

3.3.6 ΣΕΛΙΔΑ ΑΡΘΡΩΝ (ΑΝΑΚΟΙΝΩΣΕΙΣ)

Έχοντας ορίσει την σελίδα που θα δημοσιεύονται οι ανακοινώσεις στο προηγούμενο βήμα, το μόνο που έχουμε να κάνουμε σε αυτό το σημείο είναι να δημιουργήσουμε μερικές δοκιμαστικές ανακοινώσεις για να σιγουρευτούμε πως δεν έχει γίνει κάποιο λάθος.

Για να το κάνουμε αυτό πηγαίνουμε στο Άρθρα → Νέο άρθρο. Αφού έχουμε εγκαταστήσει το πρόσθετο 'Force Post Category' αα πρέπει να ορίσουμε και μια κατηγορία για το άρθρο. Αυτό μπορεί να γίνει είτε πατώντας '+Προσθήκη

κατηγορίας' από την μονάδα 'Κατηγορίες' στα δεξιά του επεξεργαστή κειμένου, είτε από το Άρθρα → Κατηγορίες. Στην περίπτωση μας θα δημιουργήσουμε την γονική κατηγορία 'Ανακοινώσεις' και τις υποκατηγορίες 'Κατηγορία 1' και 'Κατηγορία 2' στις οποίες θα κατατάξουμε τις δοκιμαστικές μας ανακοινώσεις. Προαιρετικά έχουμε την δυνατότητα να επεξεργαστούμε το σύντομο όνομα που θα εμφανίζεται στο URL και να προσθέσουμε μια περιγραφή για κάθε κατηγορία.



Επεξεργασία κατηγορίας

Όνομα
Το όνομα που εμφανίζεται στον ιστοχώρο σας.

Σύντομο όνομα
Μορφή του ονόματος καταλληλότερη για διευθύνσεις URL. Συνήθως περιέχει μόνο πεζά λατινικά, αριθμούς και ενωτικά.

Γονέας
Σε αντίθεση με τις ετικέτες, οι κατηγορίες διατάσσονται και ιεραρχικά. Π.χ., φτιάχνετε μια κατηγορία Jazz και μέσα σ' αυτήν τις (υπο)κατηγορίες Bebop και Big Band. Εντελώς προαιρετικό.

Περιγραφή
Η περιγραφή δεν εμφανίζεται από προεπιλογή, αλλά κάποια θέματα τη χρησιμοποιούν.

Εικόνα 3.27: Επεξεργασία κατηγορίας.

Έχοντας δημοσιεύσει 3 δοκιμαστικές ανακοινώσεις, η σελίδα των δημοσιεύσεων θα έχει την παρακάτω μορφή:



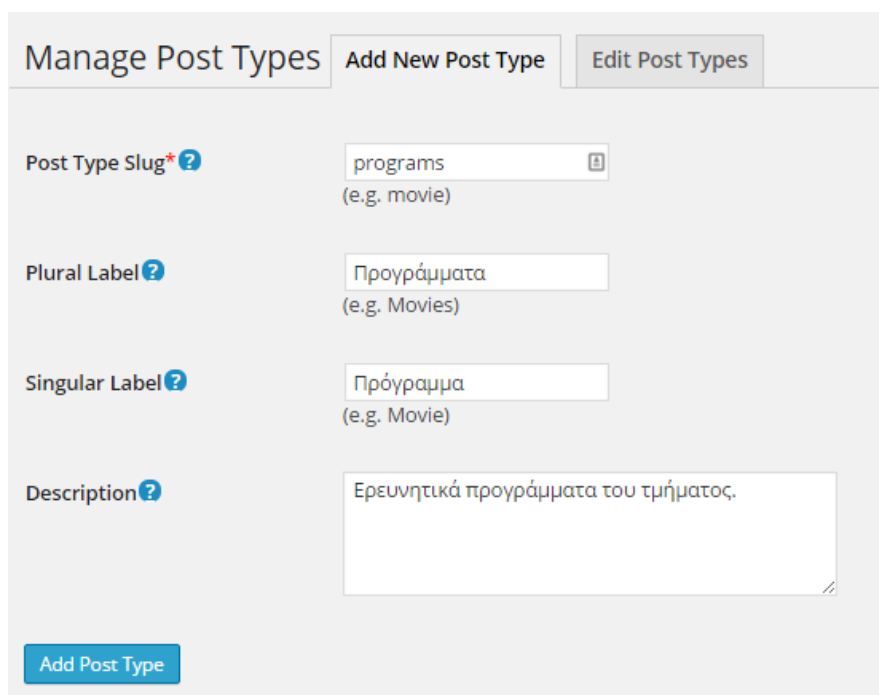
Εικόνα 3.28: Σελίδα ανακοινώσεων.

3.3.7 ΣΕΛΙΔΑ ΠΡΟΓΡΑΜΜΑΤΩΝ

Αυτή η σελίδα θα περιέχει με ένα-ένα τα ερευνητικά προγράμματα του τμήματος. Όμως θέλουμε να μπορούν να προστίθενται εύκολα και γρήγορα από τον διαχειριστή και να εμφανίζονται αυτομάτως και με ομοιόμορφο τρόπο. Γι' αυτόν τον σκοπό θα κινηθούμε αρκετά διαφορετικά σε σχέση με τις προηγούμενες σελίδες και θα γράψουμε κώδικα για να προσαρμόσουμε το αποτέλεσμα όπως εμείς ακριβώς θέλουμε. Τα βήματα που θα κάνουμε είναι λοιπόν τα εξής:

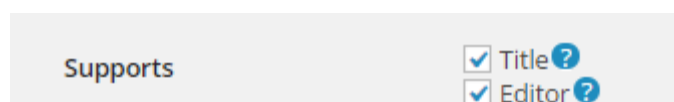
3.3.7α ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE

Πηγαίνουμε στην επιλογή CPT UI → Add/Edit Post Types του πίνακα ελέγχου (μέρος του πρόσθετου Custom Post Type UI που είχαμε εγκαταστήσει νωρίτερα) και θα δημιουργήσουμε post type με όνομα 'Προγράμματα'.



Εικόνα 3.29: Δημιουργία του post type 'programs'.

Κάνουμε scroll down και στο σημείο που λέει 'Supports' επιλέγουμε ποια από τα default στοιχεία εισαγωγής δεδομένων του WordPress θέλουμε να εμφανίζονται όταν θα προσθέτουμε ένα νέο ερευνητικό πρόγραμμα. Τα μόνα που χρειαζόμαστε από εδώ είναι ο τίτλος (title) και ο επεξεργαστής κειμένου (editor) καθώς θα προσθέσουμε στο επόμενο βήμα επιπλέον δικά μας πεδία.



Εικόνα 3.30: Στοιχεία εισαγωγής.

Στο πεδίο Menu Icon θα συμπληρώσουμε την κλάση του εικονιδίου που θέλουμε να έχει η επιλογή των προγραμμάτων στον πίνακα ελέγχου. Αυτά μπορούμε να τα βρούμε εδώ: <https://developer.wordpress.org/resource/dashicons> .

Θα χρησιμοποιήσουμε το εικονίδιο του μεγεθυντικού φακού για να υποδηλώσουμε την έρευνα, οπότε στο πεδίο θα γράψουμε 'dashicons-search'.

3.3.7β ΔΗΜΙΟΥΡΓΙΑ CUSTOM FIELDS GROUP

Πηγαίνουμε στην επιλογή Custom Fields, πατάμε το κουμπί 'Add New' για να δημιουργήσουμε μια νέα 'συλλογή' πεδίων που θα αντιστοιχούν στο νέο post type 'Προγράμματα'. Τα πεδία θα έχουν αναγκαστικά Field Label, Field Name και ένα ορισμένο Field Type το οποίο επιλέγεται από μια λίστα. Μπορούμε επίσης να ορίσουμε αν κάποιο πεδίο πρέπει να είναι συμπληρωμένο υποχρεωτικά για να προσθέσει ο διαχειριστής ένα νέο πρόγραμμα.

Αναλυτικότερα θα έχουμε τα εξής πεδία:

- Πεδίο 1 – Πλήρης τίτλος του προγράμματος
Field Label: Πλήρης Τίτλος
Field Name: full_title
Field Type: Text
- Πεδίο 2 – Περιγραφή του προγράμματος
Field Label: Περιγραφή
Field Name: description
Field Type: Text Area
- Πεδίο 3 – Εικόνα/Logo του προγράμματος
Field Label: Εικόνα Προγράμματος
Field Name: image
Field Type: image
- Πεδίο 4 – Προϋπολογισμός του προγράμματος
Field Label: Προϋπολογισμός
Field Name: budget
Field Type: Text
- Πεδίο 5 – Διάρκεια του προγράμματος
Field Label: Διάρκεια
Field Name: duration
Field Type: Text
- Πεδίο 6 – URL Ιστότοπου του προγράμματος
Field Label: URL Ιστότοπου προγράμματος
Field Name: ext_url
Field Type: Text

Στη συνέχεια θα ορίσουμε τους κανόνες, δηλαδή για ποια post types θέλουμε να εμφανίζονται αυτά τα νέα πεδία. Για να τα κάνουμε να εμφανίζονται λοιπόν σε κάθε προσθήκη νέου ερευνητικού προγράμματος με post type 'programs', θα πρέπει να υπάρχει ο κανόνας *Show this field group if Post Type is equal to programs*.

Τέλος έχουμε κάποιες επιπλέον επιλογές για τον τρόπο εμφάνισής τους στην οθόνη προσθήκης/τροποποίησης. Στην δική μας περίπτωση σε όλα τα επόμενα field groups που θα φτιάξουμε, θα επιλέξουμε ως θέση (Position) το 'High (after title)' και ως στυλ (Style) το 'Seamless (no metabox)'.

Field Order	Field Label	Field Name	Field Type
1	Πλήρης Τίτλος	full_title	Text
2	Περιγραφή	description	Text Area
3	Εικόνα Προγράμματος	image	Image
4	Προϋπολογισμός	budget	Text
5	Διάρκεια	duration	Text
6	URL Ιστότοπου Προγράμματος	ext_url	Text

Location

Rules
Create a set of rules to determine which edit screens will use these advanced

Show this field group if

Post Type is equal to programs and

Εικόνα 3.31: Προσαρμοσμένα πεδία ερευνητικών προγραμμάτων.

3.3.7γ ΚΩΔΙΚΑΣ: ΕΡΕΥΝΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ

Μέσα στον φάκελο του θέματος απογόνου που χρησιμοποιούμε, θα φτιάξουμε ένα νέο αρχείο με το όνομα 'programs-template.php'. Αυτό το αρχείο θα είναι το πρότυπο της σελίδας με τα ερευνητικά προγράμματα και θα καθορίζει την εμφάνισή της.

Στον κώδικα αρχικά ορίζουμε μια μεταβλητή που θα περιέχει SQL επερώτηση (SQL query) προς την βάση δεδομένων για να βρούμε όλα τα posts με post type = programs, δηλαδή το custom post type που φτιάξαμε στο προηγούμενο βήμα για τα ερευνητικά προγράμματα.

Στην συνέχεια θα έχουμε το WordPress Loop που θα μας εμφανίζει το ένα μετά το άλλο το περιεχόμενο αυτών των posts με την δομή που περιγράφει το παρακάτω κομμάτι κώδικα:

```
<table class="program">
  <tr>
    <td class="image" colspan="4">
      <?php $image = get_field('image');
          $url = $image['url'];
          $alt = $image['alt']; ?>
      "
width="180" height="180" />
    </td>
  </tr>
  <tr>
    <td colspan="4">
      <h2 id='full_title'><a href="<?php
the_field('ext_url');?>" <?php the_field('full_title'); ?> </a></h2>
    </td>
  </tr>
  <tr>
    <th>Προϋπολογισμός:</th>
    <td><?php the_field('budget'); ?> €</td>
    <th>Διάρκεια:</th>
    <td><?php the_field('duration'); ?></td>
  </tr>
  <tr>
    <td colspan="4" id="description">
      <p><?php the_field('description'); ?></p>
      <!-- Αν υπάρχει περιεχόμενο στον post editor προστίθεται
link προς αυτό μετά την περιγραφή -->
      <?php if( $post->post_content != "" ) { ?>
          <a class="float_r" href="<?php the_permalink();
?>">Περισσότερα...</a>
      <?php } ?>
    </td>
  </tr>
</table>
```

Βλέπουμε πως κάθε πρόγραμμα θα εμφανίζεται ως ένας πίνακας με κάποιες βασικές πληροφορίες: Μια εικόνα (συνήθως το logo της), τον πλήρη τίτλο με έναν σύνδεσμο προς τον ιστότοπο του, τον προϋπολογισμό, την διάρκεια και μια σύντομη περιγραφή. Οι πληροφορίες που αντλούνται είναι αυτές για τις οποίες δημιουργήσαμε προηγουμένως τα νέα πεδία που συνδέονται με το συγκεκριμένο post type.

Επιπλέον εφόσον ο διαχειριστής έχει επιλέξει να συμπληρώσει αναλυτικές πληροφορίες για το πρόγραμμα (στον επεξεργαστή κειμένου του WordPress), θα εμφανίζεται και ο σύνδεσμος 'Περισσότερα...' που θα οδηγεί τον χρήστη στη σελίδα με τις πληροφορίες αυτές.

Στη συνέχεια θα προσαρμόσουμε την εμφάνιση του πίνακα εντός του αρχείου style.css με το παρακάτω κομμάτι κώδικα:

```
table.program {  
    float: left;  
    width: 95%;  
    padding: 5px 20px 20px 20px;  
    margin-left: 40px;  
}  
.program th {  
    width: 5%;  
}  
.image img {  
    height: 80px;  
    width: 220px;  
    margin-left: auto;  
    margin-right: auto;  
    display: block;  
    padding: 0px 0px;  
    border-style: none;  
}  
h1#p_title {  
    font-size: 25px;  
    text-align: center;  
    margin: 0px 0px;  
    padding: 0px 0px;  
}
```

Το τελικό αποτέλεσμα που θα βλέπουμε στη σελίδα φαίνεται στην παρακάτω εικόνα:



Εικόνα 3.32: Εμφάνιση πίνακα ερευνητικού προγράμματος.

Σημείωση: Ο πλήρης κώδικας όλων των αρχείων του ιστότοπου βρίσκεται στο αντίστοιχο παράρτημα στο τέλος της εργασίας.

Τέλος, θα δημιουργήσουμε τα αρχεία `single-programs.php` & `content-programs.php`, τα οποία θα είναι σχεδόν ίδια με τα `single.php` και `content.php` που βρίσκονται στο γονικό θέμα. Η μόνη διαφορά θα είναι η παράλειψη των γραμμών κώδικα που εμφανίζουν τον τίτλο και την ημερομηνία ανάρτησης του κάθε 'άρθρου - προγράμματος', έτσι ώστε αυτά να μην εμφανίζονται στην πλήρη σελίδα του κάθε προγράμματος.



Εικόνα 3.33: Στοιχεία που θα αφαιρεθούν.

Αφού ολοκληρώσουμε τα παραπάνω βήματα για να χρησιμοποιηθεί το αρχείο `template` που φτιάξαμε, θα πρέπει να επεξεργαστούμε την σελίδα μέσω του πίνακα ελέγχου και να το ορίσουμε ως πρότυπο στις ιδιότητες σελίδων επιλέγοντας το όνομα που ορίσαμε στην αρχή του (Template Name: Research Programs).



Εικόνα 3.34: Τελική εμφάνιση.

3.3.8 ΣΕΛΙΔΑ ΕΡΓΑΣΙΩΝ

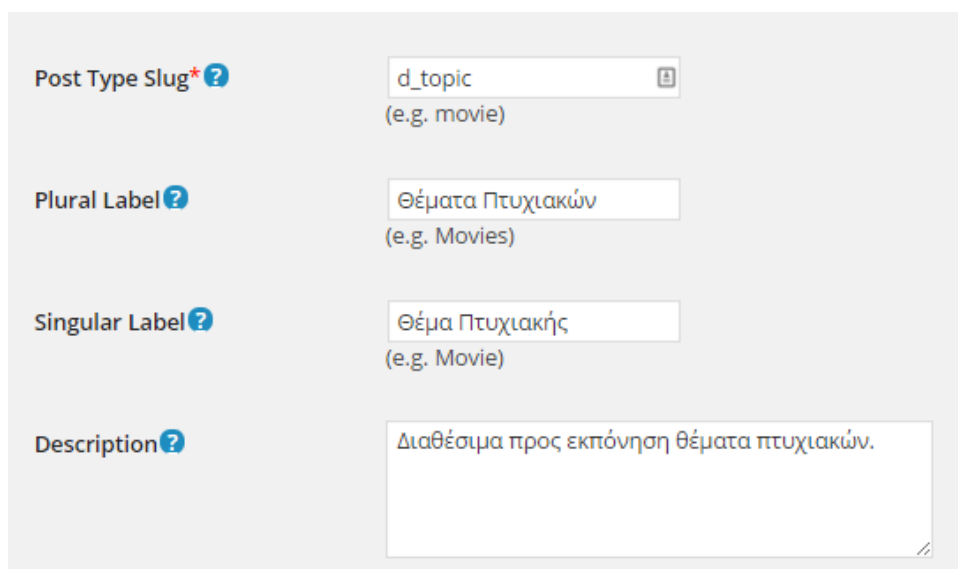
Σε αυτήν την σελίδα θα έχουμε:

- ❖ Μια δυναμική λίστα με τα διαθέσιμα θέματα πτυχιακών προς εκπόνηση τα οποία θα οδηγούν σε αντίστοιχη σελίδα με όλες τις απαραίτητες πληροφορίες για τους ενδιαφερόμενους σπουδαστές.
- ❖ Μια εξειδικευμένη λειτουργία αναζήτησης ολοκληρωμένων πτυχιακών εργασιών χρησιμοποιώντας τεχνικές AJAX για ασύγχρονη φόρτωση των αποτελεσμάτων.

Όπως και για την σελίδα με τα ερευνητικά προγράμματα, έτσι και εδώ θα πρέπει να δημιουργήσουμε τα αντίστοιχα custom post types και custom fields για τα θέματα και για τις εκπονημένες εργασίες. Για αυτά τα post types όμως θα θέλουμε επιπλέον να φτιάξουμε και ένα custom taxonomy που θα περιέχει τους εισηγητές των εργασιών.

3.3.8α ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE ΓΙΑ ΔΙΑΘΕΣΙΜΑ ΘΕΜΑΤΑ:

Στην επιλογή CPT UI → Add/Edit Post Types του πίνακα ελέγχου δημιουργούμε post type με όνομα 'Θέματα Πτυχιακών'.



The screenshot shows the 'Add/Edit Post Types' form in the CPT UI plugin. It contains the following fields:

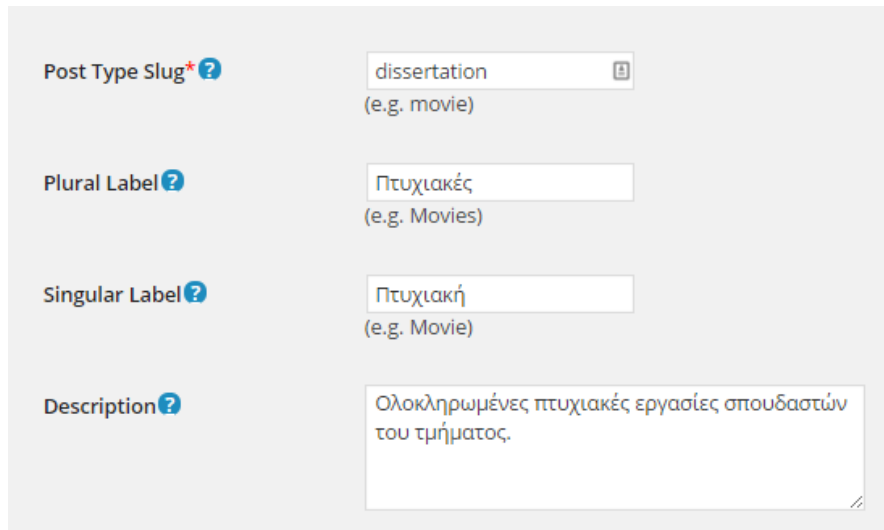
- Post Type Slug***: d_topic (with a help icon and a lock icon)
- Plural Label**: Θέματα Πτυχιακών (with a help icon)
- Singular Label**: Θέμα Πτυχιακής (with a help icon)
- Description**: Διαθέσιμα προς εκπόνηση θέματα πτυχιακών. (with a help icon)

Εικόνα 3.35: Δημιουργία του post type 'd_topic'.

Στο τμήμα 'Supports' εκτός από το 'Title' και 'Editor' θα τσεκάρουμε και την επιλογή 'Post Formats'. Αυτό θα το κάνουμε διότι θέλουμε κάθε φορά που ο χρήστης θα προσθέτει ένα νέο θέμα να δημιουργείται αυτόματα μια ανακοίνωση στην αντίστοιχη σελίδα που θα έχει διαφορετική εμφάνιση από τις υπόλοιπες. Η δομή και εμφάνιση αυτού του τύπου αναρτήσεων θα προσδιοριστεί λίγο παρακάτω. Στο Menu Icon θα συμπληρώσουμε το 'dashicons-media-default'.

3.3.8β ΔΗΜΙΟΥΡΓΙΑ CUSTOM POST TYPE ΓΙΑ ΟΛΟΚΛΗΡΩΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Στην επιλογή CPT UI → Add/Edit Post Types του πίνακα ελέγχου δημιουργούμε post type με όνομα 'Πτυχιακές'.



The screenshot shows the 'Add/Edit Post Types' form in CPT UI. It has four main sections:

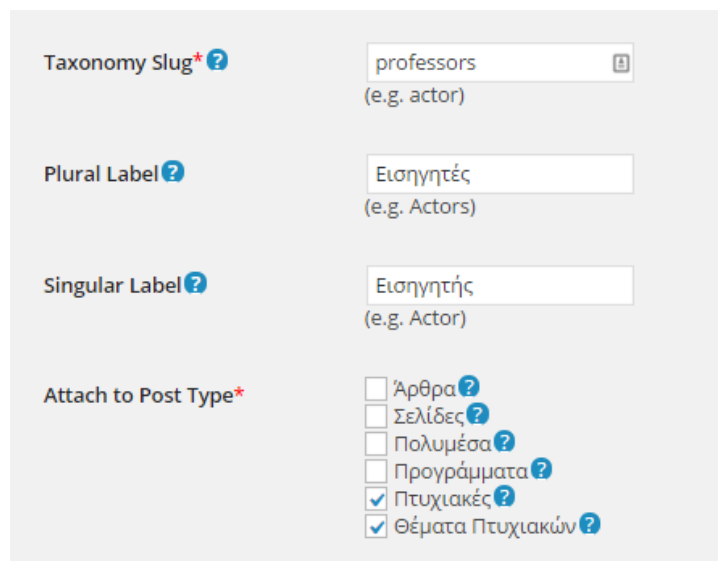
- Post Type Slug* ?**: A text input field containing 'dissertation' with a lock icon on the right. Below it, '(e.g. movie)' is written.
- Plural Label ?**: A text input field containing 'Πτυχιακές'. Below it, '(e.g. Movies)' is written.
- Singular Label ?**: A text input field containing 'Πτυχιακή'. Below it, '(e.g. Movie)' is written.
- Description ?**: A text area containing the text 'Ολοκληρωμένες πτυχιακές εργασίες σπουδαστών του τμήματος.'

Εικόνα 3.36: Δημιουργία του post type 'dissertation'.

Στο τμήμα 'Supports' θα τσεκάρουμε τις επιλογές 'Title', 'Editor', 'Author' και στο πεδίο 'Menu Icon' θα συμπληρώσουμε 'dashicons-welcome-learn-more'.

3.3.8γ ΔΗΜΙΟΥΡΓΙΑ CUSTOM TAXONOMY ΕΙΣΗΓΗΤΩΝ

Πηγαίνουμε στην επιλογή CPT UI → Add/Edit Taxonomies για να δημιουργήσουμε το taxonomy 'Εισηγητές'.



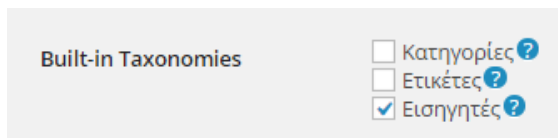
The screenshot shows the 'Add/Edit Taxonomies' form in CPT UI. It has four main sections:

- Taxonomy Slug* ?**: A text input field containing 'professors' with a lock icon on the right. Below it, '(e.g. actor)' is written.
- Plural Label ?**: A text input field containing 'Εισηγητές'. Below it, '(e.g. Actors)' is written.
- Singular Label ?**: A text input field containing 'Εισηγητής'. Below it, '(e.g. Actor)' is written.
- Attach to Post Type***: A list of checkboxes with labels and question marks:
 - Άρθρα ?
 - Σελίδες ?
 - Πολυμέσα ?
 - Προγράμματα ?
 - Πτυχιακές ?
 - Θέματα Πτυχιικών ?

Εικόνα 3.37: Δημιουργία taxonomy 'professors'

Πριν πατήσουμε 'Add Taxonomy' σιγουρευόμαστε πως έχουμε τσεκάρει τα κουτάκια με τις Πτυχιακές και τα Θέματα Πτυχιακών για να συνδέσουμε το νέο μας taxonomy με τα post types που φτιάξαμε στο προηγούμενο βήμα.

Επίσης θα πρέπει να επιστρέψουμε στην σελίδα τροποποίησης των posts types 'Πτυχιακές' και 'Θέματα Πτυχιακών' και να τσεκάρουμε το κουτί εισηγητές στο κάτω μέρος της σελίδας που λέει 'Built-in Taxonomies'.



Εικόνα 3.37: Ενσωμάτωση του taxonomy 'Εισηγητές' στα κατάλληλα post types.

3.3.8δ ΔΗΜΙΟΥΡΓΙΑ CUSTOM FIELDS GROUP

Τώρα θα δημιουργήσουμε τα επιπλέον πεδία στοιχείων που θα έχουν τα δύο νέα μας post types. Πρώτα θα έχουμε τα στοιχεία των θεμάτων προς εκπόνηση με τα πεδία:

- Πεδίο 1 – Αρχείο Θέματος της Εργασίας (π.χ. PDF)
Field Label: Αρχείο Θέματος
Field Name: topic_file
Field Type: File
Return Value: File ID
- Πεδίο 2 – Περιγραφή του θέματος
Field Label: Σύντομη Περιγραφή
Field Name: topic_description
Field Type: Text Area

Στους κανόνες επιλέγουμε 'Show this field group if Post Type is equal to d_topic'.

Field Order	Field Label	Field Name	Field Type
1	Αρχείο Θέματος	topic_file	File
2	Σύντομη Περιγραφή	topic_description	Text

Location

Rules
Create a set of rules to determine which edit screens will use these advanced custom fields

Show this field group if

Post Type is equal to d_topic and

or

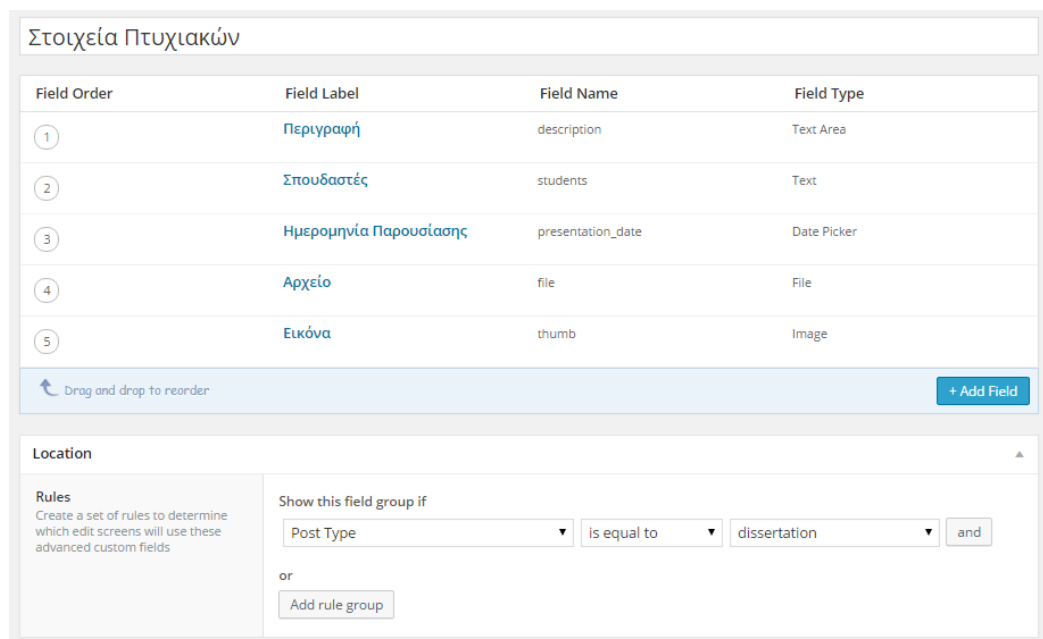
Add rule group

Εικόνα 3.38: Προσαρμοσμένα πεδία θεμάτων πτυχιακών.

Τα πεδία στοιχεία των ολοκληρωμένων εργασιών θα έχουν ως εξής:

- Πεδίο 1 – Περιγραφή της εργασίας
Field Label: Περιγραφή
Field Name: description
Field Type: Text Area
- Πεδίο 2 – Ονόματα σπουδαστών που εκπόνησαν την εργασία
Field Label: Σπουδαστές
Field Name: students
Field Type: Text
- Πεδίο 3 – Ημερομηνία Παρουσίασης
Field Label: Ημερομηνία Παρουσίασης
Field Name: presentation_date
Field Type: Date Picker
Display Format: dd/mm/yy
- Πεδίο 4 – Αρχείο εργασίας
Field Label: Αρχείο
Field Name: file
Field Type: File
Return Value: File ID
- Πεδίο 5 – Αντιπροσωπευτική εικόνα της εργασίας
Field Label: Εικόνα
Field Name: thumb
Field Type: image
Return Value: Image Object
Preview Size: Thumbnail

Στους κανόνες επιλέγουμε *'Show this field group if Post Type is equal to dissertation.'*



The screenshot shows the 'Custom Fields' editor in WordPress. At the top, the field group is titled 'Στοιχεία Πτυχιακών'. Below this is a table with the following columns: Field Order, Field Label, Field Name, and Field Type. The table contains five rows:

Field Order	Field Label	Field Name	Field Type
1	Περιγραφή	description	Text Area
2	Σπουδαστές	students	Text
3	Ημερομηνία Παρουσίασης	presentation_date	Date Picker
4	Αρχείο	file	File
5	Εικόνα	thumb	Image

Below the table, there is a 'Location' section with 'Rules'. The rule is set to 'Show this field group if' with the following configuration: Post Type is equal to dissertation. There is also an 'Add rule group' button.

Εικόνα 3.39: Προσαρμοσμένα πεδία ολοκληρωμένων πτυχιακών.

Έχοντας δημιουργήσει την υποδομή που θα χρειαστούν τα δεδομένα μας μπορούμε πλέον να φτιάξουμε τα ρηρ αρχεία μας.

3.3.8ε ΚΩΔΙΚΑΣ: ΔΙΑΘΕΣΙΜΑ ΘΕΜΑΤΑ ΠΤΥΧΙΑΚΩΝ

Όπως και για την σελίδα των προγραμμάτων, έτσι κι εδώ θα δημιουργήσουμε ένα αρχείο template το οποίο και θα ορίσουμε ως πρότυπο. Το ονομάζουμε 'dissertations-template.php' και αρχικά εντός του div περιεχομένου θα έχουμε ένα κομμάτι κώδικα που:

- Φτιάχνει ένα query προς την βάση δεδομένων για να ανακτήσει όλα τα 'posts' τύπου 'd_topic', δηλαδή τα διαθέσιμα προς εκπόνηση θέματα πτυχιακών.
- Εφόσον υπάρχουν, εμφανίζει με την μορφή λίστας τους τίτλους με συνδέσμους προς τις σελίδες λεπτομερειών και τις σύντομες περιγραφές τους.
- Αν δεν υπάρχουν, εμφανίζει αντίστοιχο μήνυμα.

```

<h5>Θέματα προς εκπόνηση:</h5>
<ul id="topics-list">
  <?php $topics = new WP_Query( 'post_type=d_topic' );
  if ( $topics->have_posts() ) {
    while ( $topics->have_posts() ): $topics->the_post();
      <?>
        <li>
          <h5 class="display_inline"><a href="<?php
the_permalink();?>"> <?php the_title();?></a> - </h5>
          <p class="display_inline f_size85"><?php
the_field('topic_description')?></p>
        </li>
      <?php
    endwhile;?>
  </ul>
  <?php
  }
  else
    echo "<h6>Δεν υπάρχουν διαθέσιμα θέματα αυτή τη στιγμή.
Επικοινωνήστε με κάποιον υπεύθυνο καθηγητή για ανάθεση.</h6>";
  <?>

```

Στην συνέχεια θα φτιάξουμε τα αρχεία 'single-d_topic.php', που ορίζει την γενική δομή της πλήρους σελίδας ενός θέματος στην οποία οδηγεί ο σύνδεσμος του τίτλου στην παραπάνω λίστα, και το 'content-d_topic.php' που περιέχει το ακριβές περιεχόμενο και καλείται εντός του 'single-d_topic.php'.

Μέσα στο *content-d_topic.php* λοιπόν θα έχουμε κώδικα που θα κάνει τα εξής:

- Εμφανίζει τον τίτλο και την σύντομη περιγραφή του θέματος ακολουθούμενο από κάποιο αρχείο με λεπτομέρειες που πιθανώς να έχει ανεβάσει ο διαχειριστής.
- Εμφανίζει το όνομα και το e-mail του εισηγητή που θα είναι ουσιαστικά το περιεχόμενο του taxonomy 'professors' που δημιουργήσαμε νωρίτερα. Τα e-mails αντλούνται από το πεδίο 'Περιγραφή' του taxonomy όπου μπορούμε να τα εισάγουμε μέσω του πίνακα ελέγχου.
- Εμφανίζει την ημερομηνία ανάρτησης του θέματος (κάτι που είχαμε αφαιρέσει για τις αντίστοιχες σελίδες των ερευνητικών προγραμμάτων) και από κάτω θα έχουμε το περιεχόμενο του επεξεργαστή κειμένου του WordPress που θα περιέχει τις όποιες άλλες λεπτομέρειες έχει προσθέσει ο διαχειριστής για το εκάστοτε θέμα.

Αφού ολοκληρώσουμε τα παραπάνω αλλά πριν επιστρέψουμε στο *dissertations-template.php* θα δημιουργήσουμε το αρχείο 'functions.php' στον φάκελο του θέματός μας. Σε αντίθεση με άλλα αρχεία του θέματος απογόνου που παρακάμπτουν τα αρχικά, το *functions.php* εμπλουτίζει το αρχικό και μας δίνει την δυνατότητα να προσθέσουμε λειτουργίες.

Μια τέτοια λειτουργία είναι και η αυτόματη ανάρτηση ενός ειδικού τύπου ανακοίνωσης στην αντίστοιχη σελίδα, για κάθε ανάρτηση ενός νέου θέματος προς εκπόνηση. Κάτι τέτοιο δεν είναι δύσκολο καθώς το μόνο που έχουμε να κάνουμε είναι πούμε στο WordPress πως θέλουμε στην σελίδα των άρθρων να εμφανίζονται εκτός από τα post types τύπου 'post', και τα τύπου 'd_topic'. Αυτό θα γίνει με την παρακάτω συνάρτηση σε php εντός του αρχείου *functions.php*:

```
//Συνάρτηση που ορίζει τα post types που θέλουμε να εμφανίζονται στην
σελίδα των άρθρων και στο feed.
function my_get_posts( $query ) {
    if ( ( is_home() && $query->is_main_query() ) || is_feed() )
        $query->set( 'post_type', array( 'post', 'd_topic' ) );

    return $query;
}
add_filter( 'pre_get_posts', 'my_get_posts' );
```

Όμως εμείς θέλουμε αυτές οι ανακοινώσεις να ξεχωρίζουν από τις κανονικές και να έχουν μια διαφορετική μορφή. Για αυτόν τον σκοπό θα χρησιμοποιήσουμε το aside (μικροάρθρο) ,ένα από τα ενσωματωμένα post formats του WordPress και θα το τροποποιήσουμε για να καλύπτει αυτή μας την ανάγκη.

Για να ενεργοποιήσουμε την υποστήριξη επιπλέον post formats στο θέμα μας θα γράψουμε μια συνάρτηση:

```
//Post Format Support
function add_post_formats() {
    add_theme_support( 'post-formats', array( 'aside' ) );
}
add_action( 'after_setup_theme', 'add_post_formats', 20 );
```

Η συνάρτηση αυτή θα προσθέσει την δυνατότητα επιλογής της μορφής ενός post στον πίνακα ελέγχου. Γι' αυτό άλλωστε είχαμε τσεκάρει και την επιλογή post formats στο τμήμα supports όταν δημιουργούσαμε το d_topic post type.

Για την διευκόλυνση του διαχειριστή θα προσθέσουμε και μία ακόμη συνάρτηση που θα ορίζει το aside ως την προεπιλεγμένη μορφή για τις ανακοινώσεις θεμάτων:


```
/* Aside (Θέματα Πτυχιακών) */
article.post-aside {
    font-size: 120%;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 3px 3px 3px rgba(0, 0, 0, .1);
    background-color: #EAF4FF;
}

a.title {
    font-size: 20px;
    font-weight: italic;
}

.mini-meta {
    font-size: 65%;
    margin: 0;
    color: #777;
}


.post-aside h2 {
    font-size: 95%;
}

.post-aside p {
    margin-top: 2%;
    margin-left: 3%;
}
}
```




Εικόνα 3.40: Ανακοίνωση νέου θέματος.

Θέμα προς εκπόνηση: GPS

Ανάπτυξη ενός ολοκληρωμένου Global Positioning System. 

Εισηγητές: Αλατσαθιανός Σ. E-mail: alatsa@teipir.gr

© 23 ΙΟΥΝΙΟΥ 2015  EDIT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nisl eros, pretium tempus augue quis, malesuada rutrum sem. In hac habitasse platea dictumst. Proin hendrerit, mauris quis iaculis facilisis, augue purus fringilla risus, a condimentum orci massa vel urna. Curabitur pretium tellus in libero porta dictum.

Εικόνα 3.41: Σελίδα λεπτομερειών νέου θέματος.

3.3.8στ ΚΩΔΙΚΑΣ: ΑΝΑΖΗΤΗΣΗ ΠΤΥΧΙΑΚΩΝ ΜΕ AJAX

Η επόμενη κύρια λειτουργία της σελίδας είναι η δυνατότητα αναζήτησης ολοκληρωμένων πτυχιακών εργασιών. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε:

- AJAX για την ασύγχρονη φόρτωση των αποτελεσμάτων έτσι ώστε να μην είναι απαραίτητο να φορτώνεται ξανά ολόκληρη η σελίδα σε κάθε αναζήτηση.
- JQuery για να μπορούμε να χρησιμοποιήσουμε τις έτοιμες βιβλιοθήκες εφέ στην παρουσίαση των αποτελεσμάτων.

Στο αρχείο *dissertations-template.php*, συνεχίζοντας από εκεί που είχαμε μείνει θα έχουμε το παρακάτω κομμάτι κώδικα:

```
<div id="dissertation-search">
    <h5>Αναζήτηση Εργασιών</h5>
    <p>
        Συμπληρώνοντας τα παρακάτω πεδία της φόρμας μπορείτε να
        αναζητήσετε τις πτυχιακές εργασίες που εκπονήθηκαν από φοιτητές του
        τμήματος. Για εμφάνιση όλων των διαθέσιμων εργασιών αφήστε τα πεδία
        κενά.
    </p>
    <form role="search" id="dissertation-search" class="searchform"
method="POST" action="">
        <label for="professor">Εισηγητής:</label>

        <select name="professor" form="dissertation-search"
id="prof_list">
            <option value=""></option>
            <?php
                $args = array(
                    'orderby' =>'name',
                    'hide_empty' => 0,
                );
                $professors= get_terms('professors',$args);
                $counter = 0;
                foreach ($professors as $professors ) {
                    $counter++;
                    echo '<option value="'. $professors->
slug.' "'>'. $professors->name.'</option>';
                }
            <?>
        </select>
        <label for="dissertation">Τίτλος Εργασίας:</label>
        <input type="text" value="" id="dissertation"
name="dissertation" class="text-search" />
        <label for="student">Όνομα Σπουδαστή:</label>
        <input type="text" value="" id="student" name="student"
class="text-search" />
        <input type="submit" id="searchsubmit" name="searchsubmit"
value="Αναζήτηση" />
        
    </form>
</div><!-- #dissertation-search -->
```


Ορίζονται 3 πεδία αναζήτησης: εισηγητή καθηγητή, τίτλου εργασίας και όνομα σπουδαστή. Το 1^ο είναι ένα drop-down μενού επιλογής που χρησιμοποιεί το taxonomy 'professors' για την δυναμική συμπλήρωση της λίστας των εισηγητών ενώ τα 2 τελευταία είναι απλά text inputs.

Στο τέλος έχουμε μια εικόνα gif που θα παίζει τον ρόλο του loader και θα είναι φανερή μόνο όσο θα βρίσκεται σε εξέλιξη μια αναζήτηση (αυτό θα επιτευχθεί αφαιρώντας την κλάση 'hidden' από το στοιχείο η οποία την κρύβει σε διαφορετική περίπτωση).

Αναζήτηση Εργασιών

Συμπληρώνοντας τα παρακάτω πεδία της φόρμας μπορείτε να αναζητήσετε τις πτυχιακές εργασίες που εκπονήθηκαν από φοιτητές του τμήματος. Για εμφάνιση όλων των διαθέσιμων εργασιών αφήστε τα πεδία κενά.

Εισηγητής:

Τίτλος Εργασίας:

Όνομα Σπουδαστή:

Αναζήτηση

Εικόνα 3.42: Φόρμα αναζήτησης εργασιών.

Στην συνέχεια θα έχουμε ένα τμήμα κώδικα ο οποίος καλεί την συνάρτηση `dissertation_search_callback()` που βρίσκεται στο `functions.php`, και θα εμφανίσει τα αποτελέσματα της αναζήτησης εντός ενός div:

```
<!-- AJAX Search Response Content -->
<h6 id="results" class="hidden">Αποτελέσματα Αναζήτησης:</h6>
<div id="ajax-response" class="hidden"> </div>
</div>
<?php
    if ( isset( $_POST['searchsubmit'] ) ) {
        dissertation_search_callback();
    }
?>
</div>
```

Για να υπάρξει όμως ασύγχρονη μεταφορά των εισαγόμενων δεδομένων και εμφάνιση των τελικών αποτελεσμάτων χωρίς την επαναφόρτωση της σελίδας θα χρησιμοποιήσουμε AJAX. Αυτό σημαίνει πως θα πρέπει να κάνουμε χρήση κώδικα JavaScript ο οποίος θα αναλάβει την επικοινωνία μεταξύ του κώδικα της σελίδας και του κώδικα της συνάρτησης που θα δούμε λίγο παρακάτω.

Αυτός ο κώδικας θα βρίσκεται σε ένα νέο αρχείο 'dissertations_search.js' μέσα στον φάκελο 'js' που δημιουργούμε εντός του φακέλου του θέματός μας. Σε αυτόν θα βάζουμε όλα τα αρχεία JavaScript, συμπεριλαμβανομένων και των βιβλιοθηκών *jQuery* και *jQuery-UI* που κατεβάζουμε από τις σελίδες <https://jquery.com/> και <https://jqueryui.com/> αντιστοίχως, για να τις χρησιμοποιήσουμε στα scripts μας.

Σημείωση: Όπως θα δούμε παρακάτω, για την jQuery θα χρησιμοποιήσουμε το CDN (Content Delivery Network) της Google και την ενσωματωμένη στο WordPress έκδοσή της, οπότε δεν είναι απαραίτητο να έχουμε επιπλέον αρχεία jquery.js και jquery-ui.js.

Έτσι λοιπόν θα έχουμε:

```
var searchForm = $("#dissertation-search");
searchForm.submit(function(e) {
  e.preventDefault();
  search_posts();
});
```

Χρησιμοποιώντας τις μεθόδους της βιβλιοθήκης jQuery, βρίσκουμε την διεύθυνση της φόρμας αναζήτησης στο DOM (Document Object Model) και την βάζουμε σε μια μεταβλητή για να μην χρειάζεται να την προσπελάσουμε για κάθε ένα ξεχωριστό πεδίο. Αυτή είναι μια μέθοδος που χρησιμοποιείται συχνά όταν θέλουμε να έχουμε πρόσβαση σε στοιχεία της σελίδας γρηγορότερα και με τους ελάχιστους δυνατούς προγραμματιστικούς πόρους. Μετά καλούμε την συνάρτηση search_posts().

Σημειωτέων πως προηγουμένως έχουμε παρεμποδίσει την default λειτουργία του κουμπιού 'submit' γιατί αυτή θα έστελνε τα δεδομένα απευθείας στην σελίδα επεξεργασίας, ενώ εμείς θέλουμε να γίνει η επικοινωνία ασύγχρονα και μετά να φορτωθούν μόνο συγκεκριμένα στοιχεία στην σελίδα μας.

Ο κώδικας της συνάρτησης διαβάζει ένα-ένα τα δεδομένα των πεδίων, τα βάζει σε έναν πίνακα 'data' και μέσω AJAX αναλαμβάνει τις παρακάτω διαμεσολαβητικές λειτουργίες πριν την αποστολή τους στην συνάρτηση dissertation_search_callback(), αλλά και μετά την λήψη των αποτελεσμάτων:

- Πριν την αποστολή των δεδομένων, κάνει την εικόνα loader της σελίδας να φαίνεται για να υποδηλώνει την επεξεργασία των στοιχείων.
- Μετά την ολοκλήρωση της επεξεργασίας θα κρύψει πάλι την εικόνα.
- Τα ληφθέντα αποτελέσματα θα 'προσθεθούν' με εφέ fade-in και με μορφή accordion (από jQuery UI) στο κατάλληλο div της σελίδας.

Σε κάθε βήμα έχουμε και ένα `console.log()` το οποίο μας βοηθά στην απασφαλμάτωση εμφανίζοντας μηνύματα και δεδομένα που μπορούμε να δούμε στην κονσόλα των Chrome Developer Tools .

```
function search_posts () {
    var dissertation = searchForm.find("#dissertation").val();
    var student = searchForm.find("#student").val();
    var professor = searchForm.find('option:selected').attr("value");
    var data = {
        action: "dissertation_search",
        dissertation: dissertation,
        student: student,
        professor: professor,
    };
    console.log("Input Search Parameters: ",data);
    $.ajax({
        cache: false,
        method: "POST",
        url : wp_localize_data.ajax_url,
        data : data,
        //dataType: '',
        beforeSend: function () {
            //Display Ajax Loader image
            $("#loading").show();
        },
        complete: function( jqXHR, textStatus ){
            //Hide Ajax Loader image
            $("#loading").hide();
            console.log("complete");
        },
        success: function(response, textStatus, jqXHR ) {
            console.log("success");
            $("#results").show();
            var $ajax_response = $( response );
            //Output response to html element with id='ajax-response'
            within template file
            $( '#ajax-response' ).fadeOut(1).html( $ajax_response
            ).fadeIn();
            //Add jQuery-ui effects on response data (dissertation
            posts)
            $( '.dissertation_post' ).accordion({
                active: false,
                collapsible: true,
                heightStyle: 'panel'
            });
        },
        error: function( jqXHR, textStatus, errorThrown ){
            console.log( 'Error Message: ' + textStatus,
            errorThrown );
        }
    });
};
```

Για να μπορούν να λειτουργήσουν τα scripts θα πρέπει να τα δηλώσουμε (register) και να τα συμπεριλάβουμε στην ουρά φόρτωσης (enqueue). Αυτό γίνεται με τα παρακάτω τμήματα κώδικα στο αρχείο *functions.php*:

Συνάρτηση που κάνει register την jQuery από το CDN της Google. Αν αυτό δεν είναι εφικτό, η τοπική έκδοση του WordPress λειτουργεί ως fallback:

```
function register_custom_jquery() {
    $url =
'https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js';
    $test_url = @fopen($url, 'r'); //Έλεγχος για το αν υπάρχει το URL
    if($test_url !== false) {
        wp_deregister_script('jquery');
        wp_register_script(
            'jquery',

'https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js',
            array(),
            '2.1.4'
        );
    }
}
add_action('init', 'register_custom_jquery');
```

Συνάρτηση που κάνει enqueue την jQuery και όλα τα jQuery-UI στοιχεία:

```
function load_all_jquery() {
    //Load jQuery
    wp_enqueue_script("jquery");
    $jquery_ui = array(
        "jquery-ui-core",
        "jquery-ui-widget",
        "jquery-ui-mouse",
        "jquery-ui-accordion",
        "jquery-ui-autocomplete",
        "jquery-ui-slider",
        "jquery-ui-tabs",
        "jquery-ui-sortable",
        "jquery-ui-draggable",
        "jquery-ui-droppable",
        "jquery-ui-selectable",
        "jquery-ui-position",
        "jquery-ui-datepicker",
        "jquery-ui-resizable",
        "jquery-ui-dialog",
        "jquery-ui-button"
    );
    foreach($jquery_ui as $script){
        //Load each jQuery UI function in the array above
        wp_enqueue_script($script);
    }
}
add_action('wp_enqueue_scripts', 'load_all_jquery');
```

Συνάρτηση που κάνει register το *dissertations_search.js* και ορίζει ότι τα δεδομένα από το script μας θα περάσουν στο αρχείο *admin-ajax.php* του WordPress, το οποίο αναλαμβάνει να εξυπηρετεί τις κλήσεις AJAX:

```
//Register Scripts
function theme_register_scripts () {
    wp_register_script( 'dissertations_search',
    get_stylesheet_directory_uri() . '/js/dissertations_search.js', array
    ('jquery'), '', true );
    wp_localize_script( 'dissertations_search', 'wp_localize_data',
    array ('ajax_url' => admin_url('admin-ajax.php')) );
}
add_action('wp_enqueue_scripts', 'theme_register_scripts', 1);
```

Συνάρτηση που κάνει enqueue το *dissertations_search.js*:

```
//Enqueue Scripts
function theme_enqueue_scripts () {
    wp_enqueue_script( 'dissertations_search',
    get_stylesheet_directory_uri() . '/js/dissertations_search.js', array
    ('jquery') );
}
add_action('wp_enqueue_scripts', 'theme_enqueue_scripts');
```

Αφού ολοκληρώσουμε αυτά τα βήματα, είμαστε έτοιμοι να δημιουργήσουμε την συνάρτηση *dissertation_search_callback()* η οποία θα λαμβάνει τα δεδομένα από την AJAX κλήση και θα κάνει τις απαραίτητες διεργασίες για να βρει τα ζητούμενα αποτελέσματα.

Αναλυτικά:

Παίρνουμε τα δεδομένα από την AJAX κλήση, αρχικοποιούμε τις μεταβλητές αναζήτησης και ορίζουμε τα arguments του πίνακα για το query.

```
function dissertation_search_callback(){
    //Get input data sent by AJAX call
    $query_data = $_POST;
    //Initialize query variables
    $dissertation_title = null;
    $students = null;
    $professors = null;
    //Arguments array for WP_Query
    $args = array(
        'post_type' => 'dissertation',
        'posts_per_page' => -1,
        'orderby' => 'meta_value',
        'meta_key'=>'students',
        'order' => 'ASC'
    )
}
```

Ελέγχουμε αν και ποια πεδία έχουν συμπληρωθεί, κάνουμε sanitize τα δεδομένα για λόγους ασφαλείας και στη συνέχεια τα βάζουμε στον πίνακα meta_query του πίνακα args που θα περιέχει τις παραμέτρους του SQL query που θα φτιαχτεί για την άντληση των αποτελεσμάτων της αναζήτησης από την βάση.

```

if ( !empty($query_data['dissertation']) ||
!empty($query_data['student']) || !empty($query_data['professor']) )
{
    if ( !empty($query_data['dissertation']) ) {
        $dissertation_title = sanitize_text_field(
$query_data['dissertation'] );
        $args['meta_query'][] = array(
            'key' => 'dissertation_title',
            'value' => $dissertation_title,
            'compare' => 'LIKE');
    }
    if ( !empty($query_data['student']) ) {
        $students = sanitize_text_field( $query_data['student']);
        $args['meta_query'][] = array(
            'key' => 'students',
            'value' => $students,
            'compare' => 'LIKE');
    }
    if ( !empty($query_data['professor']) ) {
        $professors = sanitize_text_field(
$query_data['professor'] );
        $args ['tax_query'][]= array(
            'taxonomy' => 'professors',
            'field' => 'slug',
            'terms' => $professors,
            'operator' => 'IN');
    }
}

```

Αν υπάρχουν αποτελέσματα θα επιστραφούν με την μορφή που ορίζει το πρότυπο *content-dissertation.php*, ενώ αν δεν υπάρχουν θα επιστραφεί το *content-noresults.php* που περιέχει το κατάλληλο μήνυμα. Σε κάθε περίπτωση γίνεται επαναφορά της global μεταβλητής \$post πριν το τέλος κάθε custom WP query.

```

//New WP_Query with input arguments
$dissertations = new WP_Query( $args );
if( $dissertations->have_posts() ):
    while( $dissertations->have_posts() ): $dissertations->
>the_post();
        get_template_part( 'content', 'dissertation' );
    endwhile;
    wp_reset_postdata();
    wp_die();

else:
    get_template_part('content-noresults');
endif;

wp_reset_postdata();
wp_die();
}

```

Τέλος στην περίπτωση που κανένα από τα πεδία δεν είχαν συμπληρωθεί πριν την έναρξη της αναζήτησης, η συνάρτηση θα επιστρέψει στον χρήστη το σύνολο των πτυχιακών εργασιών που υπάρχουν στην βάση δεδομένων.

```
else {
    $dissertations = new WP_Query( $args );
    while( $dissertations->have_posts() ): $dissertations->the_post();
        get_template_part( 'content', 'dissertation' );
    endwhile;

    wp_reset_postdata();
    wp_die();
}
}
```

Έχοντας γράψει τον απαραίτητο κώδικα για την λειτουργία αναζήτησης, θα δούμε και τον κώδικα που θα ορίζει την δομή και εμφάνιση της κάθε εργασίας στον χώρο των αποτελεσμάτων. Όπως προείπαμε, αυτός θα βρίσκεται στο template αρχείο content-dissertation.php και θα έχει ως εξής:

```
<h3><?php the_field('students');?> - <?php the_title(); ?></h3>
<div id="content_panel">
    <?php $file_id= get_field('file');
        $url = wp_get_attachment_url( $file_id );
        $title = get_the_title( $file_id );
        $ext = pathinfo($url, PATHINFO_EXTENSION);
        $filesize = filesize( get_attached_file( $file_id ) );
        $filesize = size_format($filesize, 2); ?>
```

Στην αρχή έχουμε τον τίτλο κάθε αποτελέσματος που θα φαίνεται πριν ανοιχτεί το content panel του στοιχείου (όπως είδαμε νωρίτερα τα αποτελέσματα θα χρησιμοποιούν το accordion από την βιβλιοθήκη jQuery-UI). Αποτελείται από το όνομα του σπουδαστή και τον τίτλο της εργασίας. Παρακάτω θα έχουμε το περιεχόμενο του πάνελ, αλλά πριν θα αποθηκεύσουμε σε μεταβλητές τα στοιχεία του προαιρετικού αρχείου που μπορεί να ανεβάσει ο χρήστης για κάθε εργασίας.

Στην συνέχεια έχουμε τα βασικά στοιχεία της εργασίας με την μορφή πίνακα.

```

<table class='details'>
  <tr>
    <th id="diss_thumbnail" rowspan=4>
      <?php $image = get_field('thumb');
      // Αν υπάρχει εικόνα την χρησιμοποιούμε στο
      μέγεθος που θέλουμε. Αν όχι χρησιμοποιούμε την default ως thumbnail
      if (!empty($image)) {
        $url = $image['url'];
        $alt = $image['alt'];
      }
      <?php
      
      <?php
      }else {<?php
      
      <?php } >
    </th>
    <th>Τίτλος</th>
    <td>
      <a href="<?php the_permalink(); >"><?php
the_title(); >&nbsp;</a>
      <a href="<?php echo $url; >"><img
class="pdf_icon"src='
      <?php
      if ($ext=='pdf') {
        bloginfo('stylesheet_directory');
        echo "/images/pdf.png"; }
      elseif ($ext=='zip'){
        bloginfo('stylesheet_directory');
        echo "/images/zip.png"; }
      else {
        bloginfo('stylesheet_directory');
        echo "/images/download.png";
      }
      <?php
      alt="download icon"/>
      </a>
      <span id="file_size"><?php echo
$filesize;></span>
    </td>
  </tr>

```

Η πρώτη σειρά περιέχει την εικόνα thumbnail που έχει ανεβάσει ο χρήστης (αν δεν υπάρχει χρησιμοποιείται μια προεπιλεγμένη), τον τίτλο της εργασίας με έναν σύνδεσμο που οδηγεί στην σελίδα με τις λεπτομέρειες, καθώς και ένα εικονίδιο (pdf, zip ή πιο γενικό για οτιδήποτε άλλο) με σύνδεσμο για κατέβασμα του αρχείου μαζί με το μέγεθός του.

Οι επόμενες σειρές περιέχουν το όνομα του σπουδαστή, την σύντομη περιγραφή, την ημερομηνία παρουσίασης και το όνομα του εισηγητή.


```

        <tr>
            <th>Σπουδαστές</th>
            <td><?php the_field('students'); ?></td>
        </tr>
        <tr>
            <th>Σύντομη Περιγραφή</th>
            <td><?php the_field('description'); ?></td>
        </tr>
        <tr>
            <th>Ημερομηνία</th>
            <td><?php $date = DateTime::createFromFormat('Ymd',
get_field('presentation_date'));
                echo $date->format('d/m/Y'); ?>
            </td>
        </tr>
        <tr>
            <td class="center_text" colspan=3><?php $taxonomies =
the_taxonomies( array( 'template' => '%s: %1' )); ?></td>
        </tr>
    </table><!-- end.details -->
</div>

```

Τέλος, για την συνολική εμφάνιση της σελίδας και του περιεχομένου της θα έχουμε το παρακάτω κομμάτι CSS κώδικα:

```

ul#topics-list li {
    list-style-type: none;
    padding-left      : 20px;
    background-image  : url('../wp-forge-child/images/arrow.png');
    background-size   : 35px 35px;
    background-position : -17px -5px;
    background-repeat : no-repeat;
}
.dissertation_post {
    margin-bottom: 15px;
    overflow: auto;
}
.hidden { display: none;}
#content_panel {padding: 8px 8px;}
table.details {
    width: 100%;
    margin-bottom: 0px;
}
table.details tr th, tr td {
    padding: 3px 3px 7px 6px;
    background-color: white;
}
#diss_thumbnail {text-align: center;}
img.d_thumb {max-width: none;}
img.pdf_icon {
    height: 25px;
    width: 25px;
}
span.file {
    float: right;
    font-size: 12px;
}

```

Εισηγητής:

Τίτλος Εργασίας:



Όνομα Σπουδαστή:

adam

Αναζήτηση

Αποτελέσματα Αναζήτησης:

▼ Adamidis Stylianos - CMS Website

	Τίτλος	CMS Website  (6,67 MB)
	Σπουδαστές	Adamidis Stylianos
	Σύντομη Περιγραφή	Ανάπτυξη Ιστοτόπου με Σύστημα Διαχείρισης Περιεχομένου WordPress.
	Ημερομηνία	31/12/2015
		Εισηγητές: Έλληνας Ι.

Εικόνα 3.43: Παράδειγμα αναζήτησης και αποτελεσμάτων με μερική εισαγωγή στοιχείων.

Όπως και με την σελίδα των προγραμμάτων, έτσι και με αυτήν θα πρέπει να επιλέξουμε το κατάλληλο πρότυπο στις ιδιότητες σελίδων πριν μπορέσουμε να δούμε το αποτέλεσμα δημοσιευμένο.

Ο κώδικας της πλήρους σελίδας μιας πτυχιακής με τις επιπλέον λεπτομέρειές της θα βρίσκεται στο *single-dissertation.php* και μετά τον τίτλο θα έχει ως εξής:

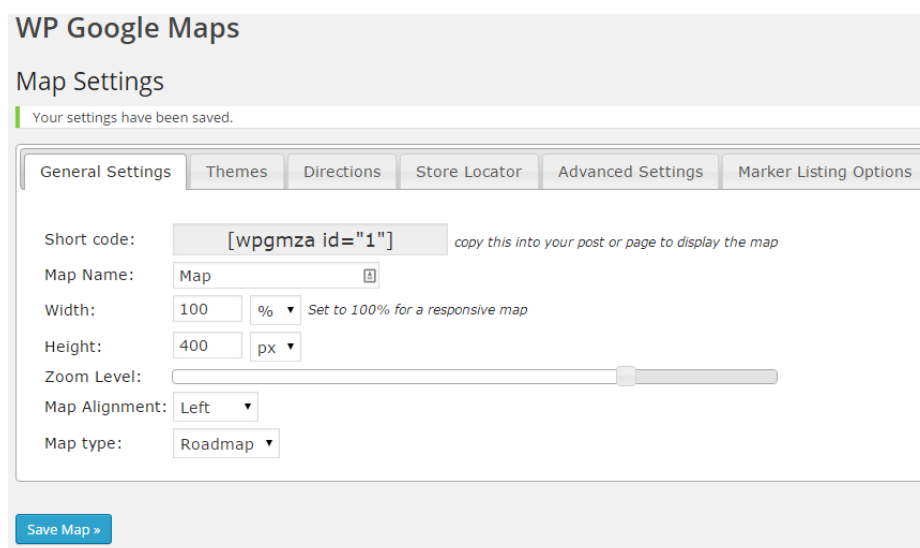
Το πεδίο της σύντομης περιγραφής της εργασίας και το εικονίδιο με τον σύνδεσμο για λήψη των αρχείων. Στη συνέχεια έχουμε το taxonomy του post, δηλαδή τον εισηγητή της πτυχιακής εργασίας, ακολουθούμενο από το πεδίο terms του taxonomy που περιέχει την e-mail διεύθυνση του καθηγητή. Κάνοντας κλικ στον σύνδεσμο της διεύθυνσης θα ανοίξει ο e-mail client του χρήστη με προ-συμπληρωμένο το θέμα με τον τίτλο της πτυχιακής και τον παραλήπτη.

3.3.9 ΣΕΛΙΔΑ ΕΠΙΚΟΙΝΩΝΙΑΣ

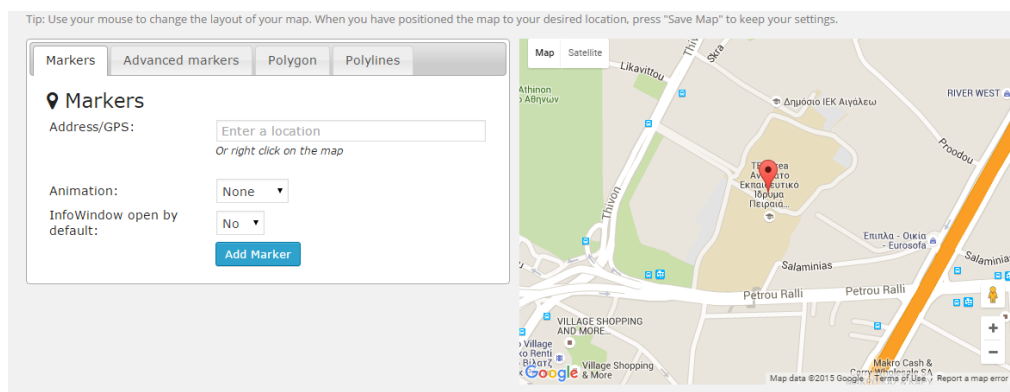
Αυτή θα είναι η κλασική σελίδα με στοιχεία επικοινωνίας, την διεύθυνση της σχολής και έναν χάρτη της Google.

Η εισαγωγή των στοιχείων θα γίνει από την επεξεργασία σελίδων του πίνακα ελέγχου και ο χάρτης με την βοήθεια του πρόσθετου 'WP Google Maps' που εγκαταστήσαμε στο ξεκίνημα της υλοποίησης.

Πάμε λοιπόν στην επιλογή 'Maps' που έχει προστεθεί στο αριστερό μενού του πίνακα ελέγχου και κάνουμε τις επιθυμητές ρυθμίσεις για τον χάρτη μας. Όταν είμαστε ικανοποιημένοι αποθηκεύουμε τον χάρτη και κάνουμε copy το short code που μας δίνεται για να το κάνουμε paste στον επεξεργαστή κειμένου της σελίδας που θέλουμε να εμφανίζεται, δηλαδή αυτήν της επικοινωνίας.



Εικόνα 3.44: Επιλογές χάρτη.



Εικόνα 3.45: Προσθήκη σημείου στον χάρτη.

Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα:

Επικοινωνία

Πληροφορίες Επικοινωνίας:

A.E.I. ΠΕΙΡΑΙΑ Τ.Τ.

Κτίριο Ε – Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων

Πέτρου Ράλλη & Θηβών 250

122 44 Αιγάλεω

Τηλ: 2105381110 – 2105381504

Κινητό:

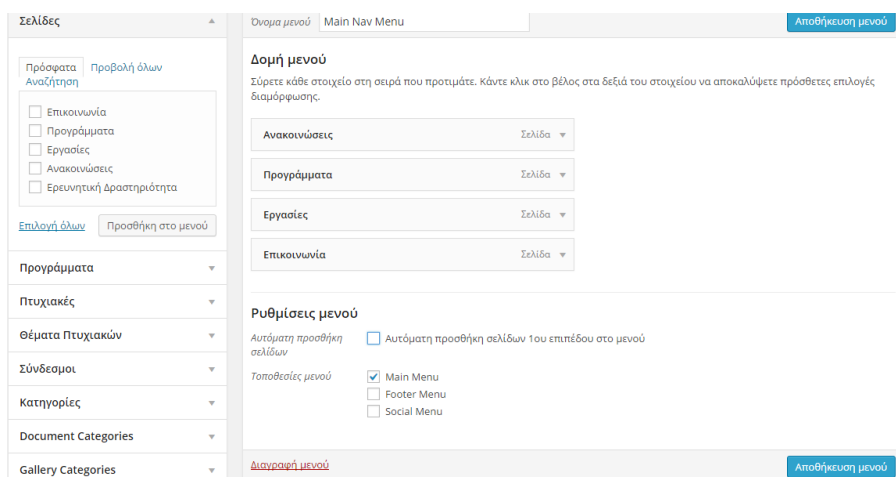
E-mail:



Εικόνα 3.46: Σελίδα επικοινωνίας.

3.3.10 ΜΕΝΟΥ ΠΛΟΗΓΗΣΗΣ

Πηγαίνουμε στο Εμφάνιση → Μενού του πίνακα ελέγχου και σέρνουμε τις σελίδες που έχουμε φτιάξει με την σειρά που θέλουμε να εμφανίζονται στο μενού μας. Την ‘Ερευνητική Δραστηριότητα’ δεν χρειάζεται να την συμπεριλάβουμε αφού την έχουμε ορίσει από πριν ως αρχική και θα εμφανίζεται πρώτη.



Εικόνα 3.47: Δημιουργία μενού πλοήγησης.

Στη συνέχεια δίνουμε στο μενού μας μια ονομασία και κάνουμε αποθήκευση.

3.3.11 WIDGETS

Σε κάποιες από τις σελίδες του ιστότοπου θα έχουμε widgets με επιπλέον πληροφορίες που θα διευκολύνουν τον επισκέπτη να βρει αυτό που αναζητά. Χρησιμοποιώντας τις επιλογές του πρόσθετου 'Dynamic Widgets' θα επιλέξουμε σε ποιες σελίδες θα εμφανίζεται το κάθε ένα από αυτά. Πηγαίνοντας λοιπόν στο Εμφάνιση → Μονάδες (Μικροεφαρμογές στις τελευταίες εκδόσεις WordPress) θα σύρουμε τα παρακάτω στο τμήμα του Main Sidebar:

List Custom Taxonomy: Το συγκεκριμένο widget θα δημιουργήσει ένα drop-down μενού με τους εισηγητές καθηγητές από το οποίο θα μπορεί ο χρήστης να βρει γρήγορα τις πτυχιακές που έχουν ολοκληρωθεί υπό την εποπτεία τους, αλλά και τυχόν διαθέσιμα θέματα τους προς εκπόνηση που μπορεί να έχουν αναρτηθεί.

List Custom Taxonomy: Εισηγητές Καθηγητές

Τίτλος:
Εισηγητές Καθηγητές

Select Taxonomy:
document-category
gallery-category
professors
category

[Hide Extended Options](#)

Show Post Counts
 Show Hierarchy
 Show Empty Terms

Order By:
Name

Ascending
 Descending

Exclude (comma-separated list of ids to exclude)

Only Show Children of (category id)

Display as Dropdown

Dynamic Widgets: Dynamic
Options set for Πτυχιακές, Θέματα Πτυχιακών, Front Page, Pages and Templates.

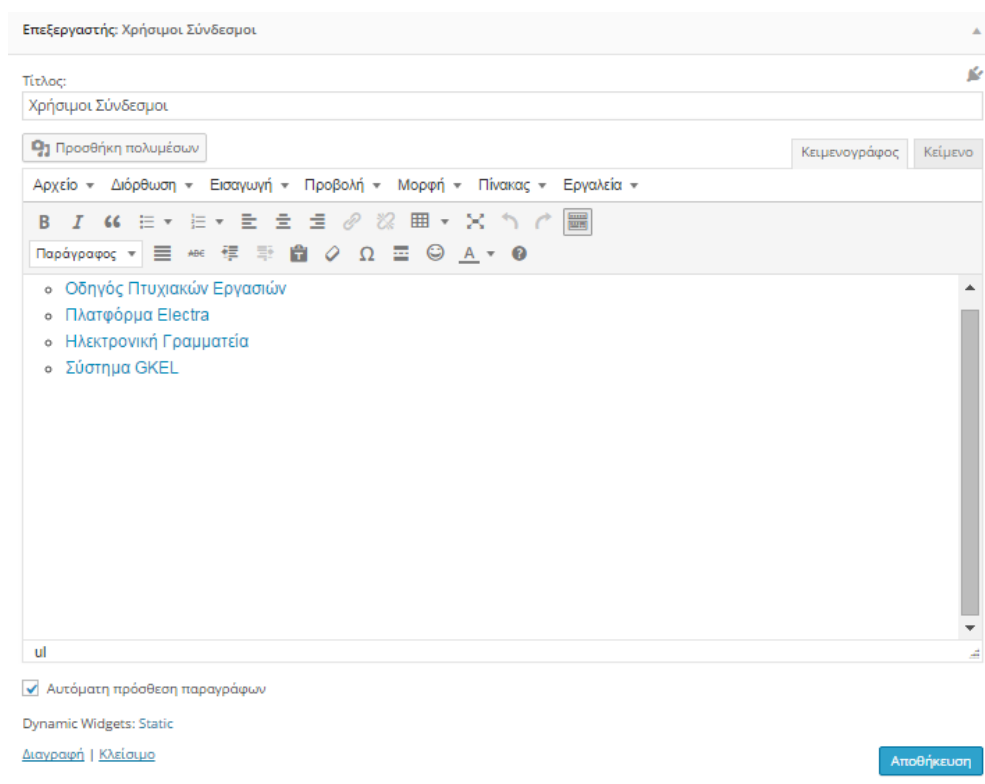
[Διαγραφή](#) | [Κλείσιμο](#) Αποθήκευση

Εικόνα 3.48: List Custom Taxonomy: Εισηγητές.

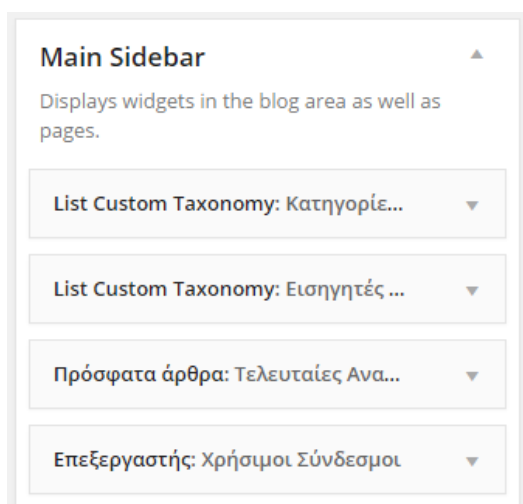
Επίσης θα το χρησιμοποιήσουμε και για να μας εμφανίζει της κατηγορίες δημοσιεύσεων στην σελίδα με τις ανακοινώσεις.

Πρόσφατα Άρθρα: Ενσωματωμένο widget του WordPress που θα μας δείχνει τα τελευταία άρθρα που έχουν δημοσιευτεί.

Επεξεργαστής: Προσθήκη του 'Black Studio TinyMCE Widget' plugin που μας επιτρέπει να γράψουμε ό,τι κείμενο θέλουμε στο sidebar. Εδώ θα εκμεταλλευτούμε αυτήν την δυνατότητα για να έχουμε μερικούς χρήσιμους συνδέσμους προς άλλους ιστότοπους/πλατφόρμες της σχολής και επιπλέον πληροφορίες για την σύνταξη πτυχιακών εργασιών.



Εικόνα 3.49: Widget επεξεργαστή κειμένου.



Εικόνα 3.50: Main Sidebar με τα widgets που προσθέσαμε.

3.3.12 ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΣΘΕΤΟΥ ATTACHMENT TAXONOMIES

Για να μπορούμε να οργανώνουμε καλύτερα τα αρχεία που ανεβάζουμε στον ιστότοπο, θα δημιουργήσουμε ένα απλό plugin που θα μας δίνει την δυνατότητα. εφόσον το θέλουμε, να τα διαχωρίζουμε σε 2 είδη κατηγοριών: documents για συλλογές εγγράφων και gallery για συλλογές εικόνων.

Θα δημιουργήσουμε λοιπόν ένα νέο αρχείο 'attachment_taxonomies.php' σε έναν φάκελο με το ίδιο όνομα εντός του directory που περιέχει τα πρόσθετα, δηλαδή το `wordpress\wp-content\plugins`. Θα πρέπει να σιγουρευτούμε πως στην αρχή του αρχείου έχουμε συμπεριλάβει το 'Plugin Name: xxxx' για να εμφανιστεί στον πίνακα ελέγχου. Ο κώδικας θα έχει ως εξής:

```
<?php
function wptp_register_attachments_tax() {
    /* register the document categories taxonomy */
    register_taxonomy( 'document-category', 'attachment',
        array(
            'labels' => array(
                'name' => 'Document Categories',
                'singular_name' => 'Document Category',
                'search_items' => 'Search Document Categories',
                'all_items' => 'All Document Categories',
                'edit_item' => 'Edit Document Categories',
                'update_item' => 'Update Document Category',
                'add_new_item' => 'Add New Document Category',
                'new_item_name' => 'New Document Category Name',
                'menu_name' => 'Document Category',
            ),
            'hierarchical' => true,
            'sort' => true,
            'show_admin_column' => true
        )
    );
    register_taxonomy( 'gallery-category', 'attachment',
        array(
            'labels' => array(
                'name' => 'Gallery Categories',
                'singular_name' => 'Gallery Category',
                'search_items' => 'Search Gallery Categories',
                'all_items' => 'All Gallery Categories',
                'edit_item' => 'Edit Gallery Categories',
                'update_item' => 'Update Gallery Category',
                'add_new_item' => 'Add New Gallery Category',
                'new_item_name' => 'New Gallery Category Name',
                'menu_name' => 'Gallery Category',
            ),
            'hierarchical' => true,
            'sort' => true,
            'show_admin_column' => true
        )
    );
}
add_action( 'init', 'wptp_register_attachments_tax', 0 );
?>
```


ΚΕΦΑΛΑΙΟ 4: ΕΠΙΛΟΓΟΣ

4.1 ΤΕΛΙΚΗ ΕΜΦΑΝΙΣΗ ΙΣΤΟΤΟΠΟΥ

Έχοντας ολοκληρώσει την ανάπτυξη του απαραίτητου κώδικα και την διαμόρφωση του ιστότοπού μας, οι βασικές σελίδες του θα έχουν την παρακάτω εμφάνιση:



Εικόνα 4.1: Αρχική σελίδα.



Εικόνα 4.2: Ανακοινώσεις.



Εικόνα 4.3: Προγράμματα.

The screenshot shows a WordPress website for the Department of Computer Systems at the University of Piraeus. The header features the university's logo and the text "ΕΡΕΥΝΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ & ΠΤΥΧΙΑΚΕΣ ΕΡΓΑΣΙΕΣ". A navigation menu includes "Αρχική", "Ανακοινώσεις", "Προγράμματα", "Εργασίες", and "Επικοινωνία". The "Εργασίες" menu item is highlighted.

Θέματα προς εκπόνηση:

- > GPS - Ανάπτυξη ενός ολοκληρωμένου Global Positioning System.
- > Website - Ανάπτυξη ιστοσελίδας για το εργαστήριο Μηχανικής Λογισμικού.
- > Arduino - Δημιουργία και προγραμματισμός ενός arduino.
- > CMS - Ανάπτυξη ενός ολοκληρωμένου Content Management System στην γλώσσα C.

Αναζήτηση Εργασιών

Συμπληρώνοντας τα παρακάτω πεδία της φόρμας μπορείτε να αναζητήσετε τις πτυχιακές εργασίες που εκπονήθηκαν από φοιτητές του τμήματος. Για εμφάνιση όλων των διαθέσιμων εργασιών αφήστε τα πεδία κενά.

Εισαγωγής:

Τίτλος Εργασίας:

Όνομα Σπουδαστή:

ΕΙΣΗΓΗΤΕΣ ΚΑΘΗΓΗΤΕΣ

Αλασσαθιανός Σ. (2)

ΤΕΛΕΥΤΑΙΕΣ ΑΝΑΚΟΙΝΩΣΕΙΣ

- [Test Post 3](#) 21 Νοεμβρίου 2015
- [Test Post 2](#) 21 Νοεμβρίου 2015
- [Test Post 1](#) 21 Νοεμβρίου 2015

ΧΡΗΣΙΜΟΙ ΣΥΝΔΕΣΜΟΙ

- [Οδηγός Πτυχιακών Εργασιών](#)
- [Πλατφόρμα Electra](#)
- [Ηλεκτρονική Γραμματεία](#)
- [Σύστημα ΓΚΕΛ](#)

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων - ΑΤΕΙ Πειραιά

Εικόνα 4.4: Εργασίες.

4.2 ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΓΑΣΙΑΣ

Κατά την διάρκεια της ανάπτυξης του ιστότοπου αποκόμισα αρκετές γνώσεις σχετικά με την δομή και τις εσωτερικές λειτουργίες της πλατφόρμας WordPress, καθώς επίσης και προγραμματιστικές τεχνικές για την υλοποίηση των ζητούμενων και τρόπους για την σωστή ενσωμάτωσή τους με αυτήν.

Ο αρχικός στόχος της δημιουργίας ενός εύκολα διαχειρίσιμου ιστότοπου επιτεύχθη με τον συνδυασμό χρήσης και παραμετροποίησης έτοιμων εργαλείων και βιβλιοθηκών που υπάρχουν διαθέσιμες για το WordPress (είτε ως έχουν είτε επεξεργασμένων και προσαρμοσμένων στις ανάγκες της εργασίας) και ανάπτυξης εξ ολοκλήρου νέων τμημάτων κώδικα ειδικά για τον σκοπό αυτό. Αυτό κατέστη δυνατό λόγω της μεγάλης ευελιξίας που προσφέρει το WordPress, κάτι που πρακτικά σημαίνει πως είναι κατάλληλο τόσο για αρχάριους χωρίς καμία εμπειρία στον προγραμματισμό όσο και για πιο προχωρημένους που μπορεί να θέλουν να δημιουργήσουν και τις δικές τους επεκτάσεις και λειτουργίες.

4.3 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Αν και ο ιστότοπος που αναπτύχθηκε λειτουργεί όπως περιμέναμε και ικανοποιεί όλες τις αρχικές μας απαιτήσεις στο πλαίσιο της πτυχιακής εργασίας, εντούτοις ο χώρος των διαδικτυακών εφαρμογών εξελίσσεται συνεχώς και με αμείωτους ρυθμούς. Αυτό σημαίνει πως πάντοτε θα μπορούσαν να προστεθούν λειτουργίες και να βελτιωθούν οι υπάρχουσες ώστε ο ιστότοπος να προσφέρει περισσότερες υπηρεσίες που ίσως χρειαστούν στο μέλλον, να βελτιωθεί η ασφάλεια και η απόδοσή του ή ακόμα και να χρησιμοποιηθούν πιο βέλτιστες τεχνικές για την επίτευξη παρόμοιων αποτελεσμάτων. Πέρα των πιθανών εξελίξεων στο χώρο, είναι σημαντικό να αναφέρουμε πως η χρήση και δημόσια έκθεση μιας ιστοσελίδας χωρίς τους βασικούς ελέγχους σε πραγματικές συνθήκες με περισσότερους χρήστες, είναι πιθανό να δημιουργήσει προβλήματα και κενά ασφαλείας που θα επηρεάσουν τον ιστότοπο, τον server αλλά και τα δεδομένα της βάσης. Οπότε πιθανές επεκτάσεις θα μπορούσαν επίσης να περιλαμβάνουν το θέμα της ασφάλειας και της απόδοσης του ιστότοπου.

ΚΕΦΑΛΑΙΟ 5: ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

Σε αυτό το κεφάλαιο παρουσιάζονται αναλυτικά τα αρχεία με τον νέο κώδικα που συγγράψαμε αλλά και τα αρχεία του αρχικού γονικού θέματος (WP-Forge) που τροποποιήθηκαν για το νέο μας child theme.

5.1 ΣΥΝΑΡΤΗΣΕΙΣ

5.1.1 functions.php

```
ini_set('display_errors', 'On');
error_reporting(E_ALL);
*/

// Exit if accessed directly
if ( !defined('ABSPATH')) exit;

// BEGIN ENQUEUE PARENT ACTION
// END ENQUEUE PARENT ACTION

//Post Format Support
function add_post_formats() {
    add_theme_support( 'post-formats', array( 'aside' ) );
}
add_action( 'after_setup_theme', 'add_post_formats', 20 );

// Ορισμός default post format για post type d_topic (Θέματα Πτυχιακών)
function slimline_default_post_format( $format ) {
    global $post_type;
    return ( 'd_topic' === $post_type ? 'aside' : $format );
}
add_filter( 'option_default_post_format', 'slimline_default_post_format' );

//Συνάρτηση που ορίζει τα post types που θέλουμε να εμφανίζονται στην
σελίδα των άρθρων και στο feed.
function my_get_posts( $query ) {
    if ( ( is_home() && $query->is_main_query() ) || is_feed() )
        $query->set( 'post_type', array( 'post', 'd_topic' ) );

    return $query;
}
add_filter( 'pre_get_posts', 'my_get_posts' );

/*
//Συνάρτηση για να εμφανίζονται και custom post types στις αναζητήσεις
μέσω categories και tags.
function namespace_add_custom_types( $query ) {
    if( is_category() || is_tag() && empty( $query->query_vars['suppress_filters'] ) ) {
        $query->set( 'post_type', array(
            'post', 'nav_menu_item', 'dissertation'
        ));
        return $query;
    }
}
```

```

}
add_filter( 'pre_get_posts', 'namespace_add_custom_types' );
*/

//Συνάρτηση για να μην έχουν default link οι εικόνες προς το ίδιο το
αρχείο.
function wpb_imagelink_setup() {
    $image_set = get_option( 'image_default_link_type' );

    if ( $image_set !== 'none' ) {
        update_option( 'image_default_link_type', 'none' );
    }
}
add_action( 'admin_init', 'wpb_imagelink_setup', 10 );

//-----
-----

function register_custom_jquery() {
    $url =
'https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js';
    $test_url = @fopen( $url, 'r' ); //Έλεγχος για το αν υπάρχει το URL
    if ( $test_url !== false ) {
        wp_deregister_script( 'jquery' );
        wp_register_script(
            'jquery',
            'https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js',
            array(),
            '2.1.4'
        );
    }
}
add_action( 'init', 'register_custom_jquery' );

function load_all_jquery() {
    //Load jQuery
    wp_enqueue_script( "jquery" );
    $jquery_ui = array(
        "jquery-ui-core",
        "jquery-ui-widget",
        "jquery-ui-mouse",
        "jquery-ui-accordion",
        "jquery-ui-autocomplete",
        "jquery-ui-slider",
        "jquery-ui-tabs",
        "jquery-ui-sortable",
        "jquery-ui-draggable",
        "jquery-ui-droppable",
        "jquery-ui-selectable",
        "jquery-ui-position",
        "jquery-ui-datepicker",
        "jquery-ui-resizable",
        "jquery-ui-dialog",
        "jquery-ui-button"
    );
    foreach( $jquery_ui as $script ) {
        //Load each jQuery UI function in the array above

```

```

        wp_enqueue_script($script);
    }
}
add_action('wp_enqueue_scripts', 'load_all_jquery');

//Register Scripts
function theme_register_scripts () {
    wp_register_script( 'dissertations_search',
get_stylesheet_directory_uri() . '/js/dissertations_search.js', array
('jquery'), '', true );
    wp_localize_script( 'dissertations_search', 'wp_localize_data',
array ('ajax_url' => admin_url('admin-ajax.php')) );
}
add_action('wp_enqueue_scripts', 'theme_register_scripts', 1);

//Enqueue Scripts
function theme_enqueue_scripts () {
    wp_enqueue_script( 'jquery' );
    wp_enqueue_script( 'dissertations_search',
get_stylesheet_directory_uri() . '/js/dissertations_search.js', array
('jquery') );
}
add_action('wp_enqueue_scripts', 'theme_enqueue_scripts');

//Dissertation Search AJAX callback function
add_action ( 'wp_ajax_dissertation_search',
'dissertation_search_callback' );
add_action ( 'wp_ajax_nopriv_dissertation_search',
'dissertation_search_callback' );

function dissertation_search_callback(){

    //Get input data sent by AJAX call
    $query_data = $_POST;
    //Initialize query variables
    $dissertation_title = null;
    $students = null;
    $professors = null;
    //Arguments array for WP_Query
    $args = array(
        'post_type' => 'dissertation',
        'posts_per_page' => -1,
        'orderby' => 'meta_value',
        'meta_key'=>'students',
        'order' => 'ASC'
    );
    //Check if any of the query variables is not empty (null,false,""
etc.)
    if ( !empty($query_data['dissertation']) ||
!empty($query_data['student']) || !empty($query_data['professor']) )
    {
        //Sanitize the values that exist and add meta_query data
accordingly
        if ( !empty($query_data['dissertation']) ) {
            $dissertation_title = sanitize_text_field(
$query_data['dissertation'] );

```



```

        $args['meta_query'][] = array(
            'key' => 'dissertation_title',
            'value' => $dissertation_title,
            'compare' => 'LIKE'
        );
    }
    if ( !empty($query_data['student']) ) {
        $students = sanitize_text_field( $query_data['student'] );
        $args['meta_query'][] = array(
            'key' => 'students',
            'value' => $students,
            'compare' => 'LIKE'
        );
    }
    if ( !empty($query_data['professor']) ) {
        $professors = sanitize_text_field( $query_data['professor']
);
        $args ['tax_query'][]= array(
            'taxonomy' => 'professors',
            'field' => 'slug',
            'terms' => $professors,
            'operator' => 'IN'
        );
    }
}

//New WP_Query with input arguments
$dissertations = new WP_Query( $args );
if( $dissertations->have_posts() ):
    while( $dissertations->have_posts() ): $dissertations->
>the_post();
        get_template_part( 'content', 'dissertation' );
        endwhile;
        wp_reset_postdata();
        wp_die();

    else:
        get_template_part('content-noresults');
    endif;

    wp_reset_postdata();
    wp_die();
}
//If variables are not set (null) get all posts from custom post
type 'dissertation'
else {
    $dissertations = new WP_Query( $args );
    while( $dissertations->have_posts() ): $dissertations->
>the_post();
        get_template_part( 'content', 'dissertation' );

        endwhile;

        wp_reset_postdata();
        wp_die();

    }
}

```

5.2 CSS

5.2.1 style.css

```
/*
Theme Name: WP-Forge Child
Template: wp-forge
Description: WP-Forge child theme.
Author: Adamidis
Version: 0.5
*/

@charset "UTF-8";
@import url('../wp-forge/style.css');
@import url('css/jquery-ui.min.css');

body {
    text-align: justify;
    font-family: 'Arimo','Helvetica Neue', Helvetica, Arial, sans-serif;
}

.float_r {
    float: right;
}

.float_l {
    float: left;
}

.center_text {
    text-align: center;
}

.center_img {
    display: block;
    margin-left: auto;
    margin-right: auto;
}

.display_inline {
    display: inline-block;
}

.f_size85 {
    font-size: 85%;
}

.email_item {
    margin-right: 12px;
}

div#ajax-response {
    margin-bottom: 10%;
}

#no-results {
    background-color: #EAF4FF;
    padding: 15px;
}
```

```
}

/* Post Meta*/
/*
span.entry-date.updated { /* Post date
    display: none;
}
*/
span.byline { /* Post author */
    display: none;
}
h1.entry-title {
    display: ; /* Post title */
    font-size: 25px;
}
/*

/* Home page */

#home_content div {
    margin: 20px;
}

#home_content p{
    font-size: 15px;
}
/*

/* Dissertations Template */
ul#topics-list li {
    list-style-type: none;
    padding-left      : 20px;
    background-image  : url('../wp-forge-child/images/arrow.png');
    background-size   : 35px 35px;
    background-position : -17px -5px;
    background-repeat  : no-repeat;
}

.dissertation_post {
    margin-bottom: 15px;
    overflow: auto;
}

.hidden {
    display: none;
}

#content_panel {
    padding: 8px 8px;
}

table.details {
    width: 100%;
    margin-bottom: 0px;
}

}
```

```
table.details tr th, tr td {
    padding: 3px 3px 7px 6px;
    background-color: white;
}

#diss_thumbnail {
    text-align: center;
}

img.d_thumb {
    max-width: none;
}

img.pdf_icon {
    height: 25px;
    width: 25px;
}

span.file {
    float: right;
    font-size: 12px;
}
/*

/* Reasearch Programs Template */
table.program {
    float: left;
    width: 95%;

    padding: 5px 20px 20px 20px;
    margin-left: 40px;
}

.program th {
    width: 5%;
}

.image img {
    height: 80px;
    width: 220px;
    margin-left: auto;
    margin-right: auto;
    display: block;
    padding: 0px 0px;
    border-style: none;
}

h1#p_title {
    font-size: 25px;
    text-align: center;
    margin: 0px 0px;
    padding: 0px 0px;
}

h2#full_title {
    font-size: 18px;
    text-align: center;
}
}
```

```
div#description {  
  
}  
/*  
  
/* Aside (Θέματα Πτυχιακών) */  
article.post-aside {  
    font-size: 120%;  
    padding: 20px;  
    border-radius: 5px;  
    box-shadow: 3px 3px 3px rgba(0, 0, 0, .1);  
    background-color: #EAF4FF;  
}  
  
a.title {  
    font-size: 20px;  
    font-weight: italic;  
}  
  
.mini-meta {  
    font-size: 65%;  
    margin: 0;  
    color: #777;  
}  
  
.post-aside h2 {  
    font-size: 95%;  
}  
  
.post-aside p {  
    margin-top: 2%;  
    margin-left: 3%;  
}  
}  
/*  
  
/* Style for smaller screens*/  
@media  
only screen and (max-width: 480px),  
(min-device-width: 480px) and (max-device-width: 640px) {  
  
/* Force table to not be like tables anymore */  
table, thead, tbody, th, td, tr {  
    display: block;  
    border: none;  
}  
table.program {  
    margin-left: 0px;  
}  
  
/* Hide table headers*/  
thead tr {  
    position: absolute;  
    top: -9999px;  
    left: -9999px;  
}  
}
```

```
td th{
  /* Behave like a row */
  border: none;
  position: relative;
  padding-left: 50%;
  background: white;
}

td:before {
  position: absolute;
  top: 6px;
  left: 6px;
  width: 45%;
  padding-right: 10px;
  white-space: nowrap;
}

img.d_thumb {
  display: none;
}

table tr.even, table tr.alt, table tr:nth-of-type(even){
  background: none;
}
/*
```

5.3 JAVASCRIPT

5.3.1 dissertations_search.js

```
jQuery(document).ready( function($) {
  var searchForm = $("#dissertation-search");

  searchForm.submit(function(e) {
    e.preventDefault();
    search_posts();
  });

  function search_posts() {

    var dissertation = searchForm.find("#dissertation").val();
    var student = searchForm.find("#student").val();
    var professor =
searchForm.find('option:selected').attr("value");

    var data = {
      action: "dissertation_search",
      dissertation: dissertation,
      student: student,
      professor: professor,
    };
    console.log("Input Search Parameters: ",data);

    $.ajax({
      cache: false,
      method: "POST",
```

```

        url : wp_localize_data.ajax_url,
        data : data,
        //dataType: '',
        beforeSend: function() {
            //Display Ajax Loader image
            $("#loading").show();
        },
        complete: function( jqXHR, textStatus ){
            //Hide Ajax Loader image
            $("#loading").hide();
            console.log("complete");
        },
        success: function(response, textStatus, jqXHR ) {
            console.log("success");
            $("#results").show();
            var $ajax_response = $( response );
            //Output response to html element with id='ajax-
            response' within template file
            $( '#ajax-response' ).fadeOut(1).html( $ajax_response
            ).fadeIn();
            //Add jQuery-ui effects on response data (dissertation
            posts)
            $( '.dissertation_post' ).accordion({
                active: false,
                collapsible: true,
                heightStyle: 'panel'
            });
        },
        error: function( jqXHR, textStatus, errorThrown ){
            console.log( 'Error Message: ' + textStatus,
            errorThrown );
        }
    });
};
});

```

5.4 ΠΡΟΣΘΕΤΟ ΣΥΝΗΜΜΕΝΩΝ

5.4.1 attachment_taxonomies.php

```

<?php
/*
Plugin Name: Attachment Taxonomies
Description: Plugin that creates taxonomies for attachments.
Version: 0.1
Author: Stylianos Adamidis
*/
?>

<?php
function wptp_register_attachments_tax() {

    /* register the document categories taxonomy */
    register_taxonomy( 'document-category', 'attachment',
        array(
            'labels' => array(

```

```

        'name' => 'Document Categories',
        'singular_name' => 'Document Category',
        'search_items' => 'Search Document Categories',
        'all_items' => 'All Document Categories',
        'edit_item' => 'Edit Document Categories',
        'update_item' => 'Update Document Category',
        'add_new_item' => 'Add New Document Category',
        'new_item_name' => 'New Document Category Name',
        'menu_name' => 'Document Category',
    ),
    'hierarchical' => true,
    'sort' => true,
    'show_admin_column' => true
)
);

register_taxonomy( 'gallery-category', 'attachment',
    array(
        'labels' => array(
            'name' => 'Gallery Categories',
            'singular_name' => 'Gallery Category',
            'search_items' => 'Search Gallery Categories',
            'all_items' => 'All Gallery Categories',
            'edit_item' => 'Edit Gallery Categories',
            'update_item' => 'Update Gallery Category',
            'add_new_item' => 'Add New Gallery Category',
            'new_item_name' => 'New Gallery Category Name',
            'menu_name' => 'Gallery Category',
        ),
        'hierarchical' => true,
        'sort' => true,
        'show_admin_column' => true
    )
);
}
add_action( 'init', 'wptp_register_attachments_tax', 0 );
?>

```

5.5 ΣΕΛΙΔΕΣ & ΠΡΟΤΥΠΑ

5.5.1 404.php

```

<?php
/**
 * The template for displaying 404 pages (Not Found).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns 404"
role="main">

```



```

<?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

<article id="post-0" class="post error404 no-results not-
found">
    <header class="entry-header">
        <h1 class="entry-title"><?php _e( 'Error 404!',
'wpforge' ); ?></h1>
    </header>

    <div class="entry-content">
        <p><?php _e( 'Η σελίδα που ψάχνετε φαίνεται πως δεν
υπάρχει πλέον. Επιστροφή στην', 'wpforge' ); ?> <a href="<?php echo
esc_url( home_url( '/' ) ); ?>">Αρχική</a></p>
        
        <br />
        <p><?php _e( 'Εναλλακτικά, αν θέλετε να έρθετε σε
επικοινωνία με κάποιον υπεύθυνο, μπορείτε να μεταβείτε στην', 'wpforge'
); ?> <a href="<?php echo esc_url( home_url( '/' ) ); ?>contact">σελίδα
επικοινωνίας</a>.</p>

    </div><!-- .entry-content -->
</article><!-- #post-0 -->

</div><!-- #content -->
<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.2 archive.php

```

<?php

/**
 * The template for displaying Archive pages.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

    <?php if ( have_posts() ) : ?>
        <header class="archive-header">
            <h4><?php
                if ( is_day() ) :
                    printf( __( 'Daily Archives: %s', 'wpforge' ),
'<span>' . get_the_date() . '</span>' );
                elseif ( is_month() ) :

```

```

                printf( __( 'Monthly Archives: %s', 'wpforge' ),
'<span>' . get_the_date( _x( 'F Y', 'monthly archives date format',
'wpforge' ) ) . '</span>' );
                elseif ( is_year() ) :
                printf( __( 'Yearly Archives: %s', 'wpforge' ),
'<span>' . get_the_date( _x( 'Y', 'yearly archives date format',
'wpforge' ) ) . '</span>' );
                elseif ( is_tax( 'post_format', 'post-format-aside'
) ) :
                        _e( 'Asides', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-audio'
) ) :
                        _e( 'Audio', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-chat' )
) :
                        _e( 'Chats', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-
gallery' ) ) :
                        _e( 'Galleries', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-image'
) ) :
                        _e( 'Images', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-link' )
) :
                        _e( 'Links', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-quote'
) ) :
                        _e( 'Quotes', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-status'
) ) :
                        _e( 'Status', 'wpforge' );

                elseif ( is_tax( 'post_format', 'post-format-video'
) ) :
                        _e( 'Videos', 'wpforge' );

                else :
                        _e( 'Άρχειο', 'wpforge' );
                endif;
        ?></h4>
</header><!-- .archive-header -->

<?php
/* Start the Loop */
while ( have_posts() ) : the_post();

        /* Include the post format-specific template for the
content. If you want to
        * this in a child theme then include a file called
content-____.php
        * (where ____ is the post format) and that will be used
instead.
        */

```

```

        get_template_part( 'content', get_post_format() );

    endwhile;

    wpforge_content_nav( 'nav-below' );
?>

<?php else : ?>
    <?php get_template_part( 'content', 'none' ); ?>
<?php endif; ?>

</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.3 category.php

```

<?php
/**
 * The template for displaying Category pages.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

    <?php if ( have_posts() ) : ?>
        <header class="archive-header">
            <h4><?php printf( __( 'Αρχείο: %s', 'wpforge' ),
'<span>' . single_cat_title( '', false ) . '</span>' ); ?></h4>

            <?php if ( category_description() ) : // Show an optional
category description ?>
                <div class="archive-meta"><?php echo
category_description(); ?></div>
            <?php endif; ?>
        </header><!-- .archive-header -->

        <?php
        /* Start the Loop */
        while ( have_posts() ) : the_post();

            /* Include the post format-specific template for the
content. If you want to
            * this in a child theme then include a file called
called content-____.php
            * (where ____ is the post format) and that will be used
instead.
            */
            get_template_part( 'content', get_post_format() );

        endwhile;

```

```

        wpforge_content_nav( 'nav-below' );
    }
}

<?php else : ?>
    <?php get_template_part( 'content', 'none' ); ?>
<?php endif; ?>

</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.4 content.php

```

<?php
/**
 * The default template for displaying content. Used for both single and
 * index/archive/search.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */
?>

<article id="post-<?php the_ID(); ?>" <?php post_class(); ?>>
    <header class="entry-header">
        <?php wpforge_entry_meta_categories(); ?>

        <?php if ( is_single() ) : ?>
            <h1 class="entry-title"><?php the_title(); ?></h1>
            <?php if ( get_theme_mod( 'wpforge_single_thumb_display'
) == 'yes' ) : // Show thumbnail in single post view if theme customizer
option is set to yes ?>
                <?php the_post_thumbnail(); ?>
            <?php endif; // end if ?>
            <?php else : ?>
                <h1 class="entry-title">
                    <a href="<?php the_permalink(); ?>" title="<?php
echo esc_attr( sprintf( __( 'Permalink to %s', 'wpforge' ),
the_title_attribute( 'echo=0' ) ) ); ?>" rel="bookmark"><?php
the_title(); ?></a>
                </h1>
            <?php endif; // is_single() ?>

        <div class="entry-meta-header">
            <?php wpforge_entry_meta_header(); ?>

            <?php if ( comments_open() ) : ?>
                <span class="genericon genericon-comment"></span>
                <?php comments_popup_link( '<span class="leave-reply">' . __( 'Comment',
'wpforge' ) . '</span>', __( '1 Comment', 'wpforge' ), __( '% Comments',
'wpforge' ) ); ?>
            <?php endif; // comments_open() ?>
            <?php edit_post_link( __( 'Edit', 'wpforge' ), '<span
class="edit-link"><span class="genericon genericon-edit"></span> ',
'</span>' ); ?>

```

```

        </div><!-- end .entry-meta-header -->
        <?php if ( is_home() && get_theme_mod(
'wpforge_thumb_display' ) == 'yes' ) : // Display thumbnail on home page
if theme customizer option is set to yes ?>
            <?php the_post_thumbnail(); ?>
        <?php endif; // end if ?>
    </header><!-- .entry-header -->

    <?php if ( is_home() && get_theme_mod( 'wpforge_post_display' )
== 'excerpt' ) : // Display Excerpts if theme customizer option is set
to excerpt ?>
        <div class="entry-summary">
            <?php the_excerpt(); ?>
        </div><!-- .entry-summary -->
    <?php else : ?>
        <div class="entry-content">
            <?php the_content( __( 'Continue reading <span
class="meta-nav">&raquo;</span>', 'wpforge' ) ); ?>
            <?php wp_link_pages( array( 'before' => '<div
class="page-links">' . __( 'Pages:', 'wpforge' ), 'after' => '</div>' )
); ?>
        </div><!-- .entry-content -->

    <?php endif; // end if ?>

    <footer class="entry-meta">
        <div class="entry-meta-footer">
            <?php wpforge_entry_meta_footer(); ?>
        </div><!-- end .entry-meta-footer -->
        <?php get_template_part( 'content', 'author' ); ?>
    </footer><!-- .entry-meta -->
</article><!-- #post -->

```

5.5.5 content-aside.php

```

<?php
/**
 * The template for displaying posts in the Aside post format on index
and archive pages ( Aside == Θέμα Πτυχιακής ).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */
?>

<article class="post post-aside">
    <h2
        class="entry-title">Θέμα Πτυχιακής:
        <a class="title" href="<?php the_permalink(); ?>"><?php
the_title(); ?></a>
        <?php // Αν υπάρχει αρχείο να εμφανίζεται εικονίδιο για
download

        if( get_field('topic_file')!= "" ) {
            $file_id= get_field('topic_file');
            $url = wp_get_attachment_url( $file_id );
            $title = get_the_title( $file_id );
            $ext = pathinfo($url, PATHINFO_EXTENSION);

```



```

        <?php wpforge_entry_meta_header(); ?>

        <?php if ( comments_open() ) : ?>
            <span class="genericon genericon-comment"></span>
<?php comments_popup_link( '<span class="leave-reply">' . __( 'Comment',
'wpforge' ) . '</span>', __( '1 Comment', 'wpforge' ), __( '% Comments',
'wpforge' ) ); ?>
        <?php endif; // comments_open() ?>
        <?php edit_post_link( __( 'Edit', 'wpforge' ), '<span
class="edit-link"><span class="genericon genericon-edit"></span> ',
'</span>' ); ?>
    </div><!-- end .entry-meta-header -->
    <?php if ( is_home() && get_theme_mod(
'wpforge_thumb_display' ) == 'yes' ) : // Display thumbnail on home page
if theme customizer option is set to yes ?>
        <?php the_post_thumbnail(); ?>
    <?php endif; // end if ?>
</header><!-- .entry-header -->

    <?php if ( is_home() && get_theme_mod( 'wpforge_post_display' )
== 'excerpt' ) : // Display Excerpts if theme customizer option is set
to excerpt ?>
        <div class="entry-summary">
            <?php the_excerpt(); ?>
        </div><!-- .entry-summary -->
    <?php else : ?>
        <div class="entry-content">
            <?php the_content( __( 'Continue reading <span
class="meta-nav">&raquo;</span>', 'wpforge' ) ); ?>
            <?php wp_link_pages( array( 'before' => '<div
class="page-links">' . __( 'Pages:', 'wpforge' ), 'after' => '</div>' )
); ?>
        </div><!-- .entry-content -->

    <?php endif; // end if ?>

    <footer class="entry-meta">
        <div class="entry-meta-footer">
            <?php wpforge_entry_meta_footer(); ?>
        </div><!-- end .entry-meta-footer -->
        <?php get_template_part( 'content', 'author' ); ?>
    </footer><!-- .entry-meta -->
</article><!-- #post -->

```

5.5.7 content-dissertation.php

```

<?php
/**
 * The Template for displaying dissertation post type
content (Ολοκληρωμένη πτυχιακή).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */
?>
<div class="dissertation_post">
    <h3><?php the_field('students');?> - <?php the_title(); ?></h3>
    <div id="content_panel">
        <?php $file_id= get_field('file');

```



```

$url = wp_get_attachment_url( $file_id );
$title = get_the_title( $file_id );
$ext = pathinfo($url, PATHINFO_EXTENSION);
$filesize = filesize( get_attached_file( $file_id ) );
$filesize = size_format($filesize, 2);
?>
<table class='details'>
  <tr>
    <th id="diss_thumbnail" rowspan=4>
      <?php $image = get_field('thumb');
      // Αν υπάρχει εικόνα την χρησιμοποιούμε στο
      μέγεθος που θέλουμε. Αν όχι χρησιμοποιούμε την default ως thumbnail
      if (!empty($image)) {
        $url = $image['url'];
        $alt = $image['alt'];
      }
      " width="150" height="150" />
      <?php
    }else {?>
      
      <?php } ?>
    </th>
    <th>Τίτλος</th>
    <td>
      <a href="<?php the_permalink(); ?>"><?php
      the_title(); ?>&nbsp;</a>
      <a href="<?php echo $url; ?>"><img
      class="pdf_icon"src='
      <?php
      if ($ext=='pdf') {
        bloginfo('stylesheet_directory');
        echo "/images/pdf.png"; }
      elseif ($ext=='zip'){
        bloginfo('stylesheet_directory');
        echo "/images/zip.png"; }
      else {
        bloginfo('stylesheet_directory');
        echo "/images/download.png";
      }
      ?>
      alt="download icon"/>
      </a>
      <span id="file_size"><?php echo
      $filesize;?></span>
    </td>
  </tr>
  <tr>
    <th>Σπουδαστές</th>
    <td>
      <?php the_field('students'); ?>
    </td>
  </tr>
  <tr>
    <th>Σύντομη Περιγραφή</th>
    <td>
      <?php the_field('description'); ?>
    </td>
  </tr>

```

```

        </tr>
        <tr>
            <th>Ημερομηνία</th>
            <td><?php $date = DateTime::createFromFormat('Ymd',
get_field('presentation_date'));
            echo $date->format('d/m/Y');?>
        </td>
        </tr>
        <tr>
            <td class="center_text" colspan=3><?php $taxonomies =
the_taxonomies( array( 'template' => '%s: %1' )); ?></td>
        </tr>
    </table><!-- end.details -->
</div>
</div> <!-- end.dissertation_post -->

```

5.5.8 content-noresults.php

```

<?php
/**
 * The template for no entries found in dissertation search results.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */
?>
<div id="no-results">
    <h5 class="center_text">Δεν βρέθηκαν εγγραφές</h4>
</div>

```

5.5.9 content-programs.php

```

<?php
/**
 * The Template for displaying program post type content(Ερευνητικό
Πρόγραμμα).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */
?>
<article id="post-<?php the_ID(); ?>" <?php post_class(); ?>>
    <header class="entry-header">
        <?php wpforge_entry_meta_categories(); ?>
        <div class="entry-meta-header">
            <?php edit_post_link( __( 'Edit', 'wpforge' ), '<span
class="edit-link"><span class="genericon genericon-edit"></span> ',
'</span>' ); ?>
        </div><!-- end .entry-meta-header -->
        <?php if ( is_home() && get_theme_mod( 'wpforge_thumb_display' )
== 'yes' ) : // Display thumbnail on home page if theme customizer
option is set to yes ?>
            <?php the_post_thumbnail(); ?>
        <?php endif; // end if ?>
    </header><!-- .entry-header -->

```

```

<?php if ( is_home() && get_theme_mod( 'wpforge_post_display' ) ==
'excerpt' ) : // Display Excerpts if theme customizer option is set to
excerpt ?>
    <div class="entry-summary">
        <?php the_excerpt(); ?>
    </div><!-- .entry-summary -->
<?php else : ?>
    <div class="entry-content">
        <?php the_content( __( 'Continue reading <span class="meta-
nav">&raquo;</span>', 'wpforge' ) ); ?>
        <?php wp_link_pages( array( 'before' => '<div class="page-
links">' . __( 'Pages:', 'wpforge' ), 'after' => '</div>' ) ); ?>
    </div><!-- .entry-content -->

<?php endif; // end if ?>

<footer class="entry-meta">
    <div class="entry-meta-footer">
        <?php wpforge_entry_meta_footer(); ?>
    </div><!-- end .entry-meta-footer -->
    <?php get_template_part( 'content', 'author' ); ?>
</footer><!-- .entry-meta -->
</article><!-- #post -->

```

5.5.10 dissertations-template.php

```

<?php
/**
 * Template Name: Dissertations
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <h5>Θέματα προς εκπόνηση:</h5>
    <ul id="topics-list">
        <?php $topics = new WP_Query( 'post_type=d_topic' );
        if ( $topics->have_posts() ) {
            while ( $topics->have_posts() ) : $topics->the_post();
                ?>
                <li>
                    <h5 class="display_inline"><a href="<?php
the_permalink();?>"> <?php the_title();?></a> - </h5>
                    <p class="display_inline f_size85"><?php
the_field('topic_description')?></p>
                </li>
                <?php
            endwhile;?>
        }
    </ul>
    <?php
    }
    else
        echo "<h6>Δεν υπάρχουν διαθέσιμα θέματα αυτή τη στιγμή.
Επικοινωνήστε με κάποιον υπεύθυνο καθηγητή για ανάθεση.</h6>";?>

```

```

</br>

<div id="dissertation-search">
  <h5>Αναζήτηση Εργασιών</h5>
  <p>
    Συμπληρώνοντας τα παρακάτω πεδία της φόρμας μπορείτε να
    αναζητήσετε τις πτυχιακές εργασίες που εκπονήθηκαν από φοιτητές του
    τμήματος. Για εμφάνιση όλων των διαθέσιμων εργασιών αφήστε τα πεδία
    κενά.
  </p>
  <form role="search" id="dissertation-search"
class="searchform" method="POST" action="">
  <!-- <input type="hidden" name="post_type"
value="dissertation" />-->
  <label for="professor">Εισηγητής:</label>

  <select name="professor" form="dissertation-search"
id="prof_list">
    <option value=""></option>
    <?php
      $args = array(
        'orderby' =>'name',
        'hide_empty' => 0,
      );
      $professors= get_terms('professors',$args);
      $counter = 0;
      foreach ($professors as $professors ) {
        $counter++;
        echo '<option value="' . $professors->
slug.'"' . $professors->name.'</option>';
      }
    <?>
  </select>
  <label for="dissertation">Τίτλος Εργασίας:</label>
  <input type="text" value="" id="dissertation"
name="dissertation" class="text-search" />
  <label for="student">Όνομα Σπουδαστή:</label>
  <input type="text" value="" id="student" name="student"
class="text-search" />

  <input type="submit" id="searchsubmit"
name="searchsubmit" value="Αναζήτηση" />
  
  </form>

</div><!-- #dissertation-search -->

<!-- AJAX Search Response Content -->
<h6 id="results" class="hidden">Αποτελέσματα Αναζήτησης:</h6>
<div id="ajax-response" class="hidden"> </div>
<div>
  <?php
    if ( isset( $_POST['searchsubmit'] ) ) {
      dissertation_search_callback();
    }
  <?>
</div>

```

```
</div><!-- #content -->
```

```
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

5.5.11 index.php

```
<?php
/**
 * Main template file.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">
    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>
    <?php if ( have_posts() ) : ?>
        <?php while ( have_posts() ) : the_post(); ?>

            <?php get_template_part( 'content', get_post_format() );
?>

        <?php endwhile; ?>

        <?php wpforge_content_nav( 'nav-below' ); ?>

    <?php else : ?>
        <?php get_template_part( 'content', 'none' ); ?>
    <?php endif; // end have_posts() check ?>
</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

5.5.12 programs-template.php

```
<?php
/**
 * Template Name: Research Programs
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <h3 class="center_text">Ερευνητικά Προγράμματα τμήματος
Μηχανικών Η/Υ Συστημάτων:</h3><br/>

    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>
```

```

        <?php $programs=new WP_Query(array('post_type' =>
'programs')); ?>

        <?php while($programs->have_posts() : $programs->
the_post(); ?>
<table class="program">
    <tr>
        <td class="image" colspan="4">
            <?php $image = get_field('image');
                $url = $image['url'];
                $alt = $image['alt']; ?>
            "
width="180" height="180" />
        </td>
    </tr>
    <tr>
        <td colspan="4">
            <h2 id='full_title'><a href="<?php the_field('ext_url');?>">
<?php the_field('full_title'); ?> </a></h2>
        </td>
    </tr>
    <tr>
        <th>Προϋπολογισμός:</th>
        <td><?php the_field('budget'); ?> €</td>
        <th>Διάρκεια:</th>
        <td><?php the_field('duration'); ?></td>
    </tr>
    <tr>
        <td colspan="4" id="description">
            <p><?php the_field('description'); ?></p>
            <!-- Αν υπάρχει περιεχόμενο στον post editor προσίθεται
link προς αυτό μετά την περιγραφή -->
            <?php if( $post->post_content != "" ) { ?>
                <a class="float_r" href="<?php the_permalink();
?>">Περισσότερα...</a>
            <?php } ?>
        </td>
    </tr>
</table>

        <?php endwhile; ?>
</div><!-- #content -->
<?php get_sidebar(); ?>

<?php get_footer(); ?>

```

5.5.13 search.php

```

<?php
/**
 * The template for displaying Search Results pages.
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

```

```

<?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

<?php if ( have_posts() ) : ?>

    <header class="page-header">
        <h3><?php printf( __( 'Αποτελέσματα Αναζήτησης για: %s',
'wpforge' ), '<span>' . get_search_query() . '</span>' ); ?></h3>
    </header>

    <?php /* Start the Loop */ ?>
    <?php while ( have_posts() ) : the_post(); ?>
        <?php get_template_part( 'content', 'loop' ); ?>
    <?php endwhile; ?>

    <?php wpforge_content_nav( 'nav-below' ); ?>

<?php else : ?>
    <?php get_template_part( 'content', 'none' ); ?>
<?php endif; ?>

</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.14 single-d_topic.php

```

<?php
/**
 * The Template for displaying Dissertation Topics Posts (Θέμα
Πτυχιακής).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

    <?php while ( have_posts() ) : the_post(); ?>

        <?php get_template_part( 'content-d_topic',
get_post_format() ); ?>

        <nav class="nav-single">
            <span class="nav-previous"><?php previous_post_link(
'%link', '<span class="meta-nav">' . _x( '&laquo;', 'Previous post
link', 'wpforge' ) . '</span> %title' ); ?></span>
            <span class="nav-next"><?php next_post_link(
'%link', '%title <span class="meta-nav">' . _x( '&raquo;', 'Next post
link', 'wpforge' ) . '</span>' ); ?></span>
        </nav><!-- .nav-single -->

```

```

        <?php comments_template( '', true ); ?>

    <?php endwhile; // end of the loop. ?>

</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.15 single-dissertation.php

```

<?php
/**
 * The Template for displaying all Dissertation posts (Ολοκληρωμένη
 Πτυχιακή).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child
 */

get_header(); ?>

<div id="content" class="medium-8 large-8 columns" role="main">

    <?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>

    <?php while ( have_posts() ) : the_post(); ?>

        <article id="post-<?php the_ID(); ?>" <?php
post_class(); ?>>
            <header class="entry-header">
                <?php wpforge_entry_meta_categories(); ?>

                <?php if ( is_single() ) : ?>
                    <h1 class="entry-title">Πτυχιακή Εργασία:
<?php the_title(); ?></h1>
                    <?php if ( get_theme_mod(
'wpforge_single_thumb_display' ) == 'yes' ) : // Show thumbnail in
single post view if theme customizer option is set to yes ?>
                        <?php the_post_thumbnail(); ?>
                    <?php endif; // end if ?>
                    <?php else : ?>
                        <h1 class="entry-title">
                            <a href="<?php the_permalink(); ?>"
title="<?php echo esc_attr( sprintf( __( 'Permalink to %s', 'wpforge' ),
the_title_attribute( 'echo=0' ) ) ); ?>" rel="bookmark"><?php
the_title(); ?></a>

                        </h1>
                    <?php endif; // is_single() ?>

                <p>
                    <?php the_field('description'); ?>
                    <?php // Αν υπάρχει αρχείο να εμφανίζεται
εικονίδιο για download

```



```

    . __ ( 'Comment', 'wpforge' ) . '</span>', __ ( '1 Comment', 'wpforge' ),
    __ ( '% Comments', 'wpforge' ) ); ?>
        <?php endif; // comments_open() ?>
        <?php edit_post_link( __ ( 'Edit', 'wpforge'
), '<span class="edit-link"><span class="genericon genericon-
edit"></span> ', '</span>' ); ?>
        </div><!-- end .entry-meta-header -->
        <?php if ( is_home() && get_theme_mod(
'wpforge_thumb_display' ) == 'yes' ) : // Display thumbnail on home page
if theme customizer option is set to yes ?>
            <?php the_post_thumbnail(); ?>
        <?php endif; // end if ?>
    </header><!-- .entry-header -->

    <?php if ( is_home() && get_theme_mod(
'wpforge_post_display' ) == 'excerpt' ) : // Display Excerpts if theme
customizer option is set to excerpt ?>
        <div class="entry-summary">
            <?php the_excerpt(); ?>
        </div><!-- .entry-summary -->
    <?php else : ?>
        <div class="entry-content">
            <?php the_content( __ ( 'Continue reading
<span class="meta-nav">&raquo;</span>', 'wpforge' ) ); ?>
            <?php wp_link_pages( array( 'before' =>
'<div class="page-links">' . __ ( 'Pages:', 'wpforge' ), 'after' =>
'</div>' ) ); ?>
        </div><!-- .entry-content -->

    <?php endif; // end if ?>

    <footer class="entry-meta">
        <div class="entry-meta-footer">
            <?php wpforge_entry_meta_footer(); ?>
        </div><!-- end .entry-meta-footer -->
        <?php get_template_part( 'content', 'author'
); ?>

    </footer><!-- .entry-meta -->
</article><!-- #post -->

    <?php comments_template( '', true ); ?>

<?php endwhile; // end of the loop. ?>

</div><!-- #content -->

<?php get_sidebar(); ?>
<?php get_footer(); ?>

```

5.5.16 single-programs.php

```

<?php
/**
 * The Template for displaying all Research Program Posts (Ερευνητικό
Πρόγραμμα).
 *
 * @package WordPress
 * @subpackage WP_Forge-Child

```

```
*/  
get_header(); ?>  
  
<div id="content" class="medium-8 large-8 columns" role="main">  
  
<?php if ( function_exists('yoast_breadcrumb') ) {  
yoast_breadcrumb('<p class="breadcrumbs">', '</p>'); } ?>  
  
<?php while ( have_posts() ) : the_post(); ?>  
  
<?php get_template_part( 'content-programs',  
get_post_format() ); ?>  
  
<?php comments_template( '', true ); ?>  
  
<?php endwhile; // end of the loop. ?>  
  
</div><!-- #content -->  
  
<?php get_sidebar(); ?>  
<?php get_footer(); ?>
```

ΚΕΦΑΛΑΙΟ 6: ΒΙΒΛΙΟΓΡΑΦΙΑ

6.1 ΣΥΓΓΡΑΜΜΑΤΑ

1. Ανάπτυξη Διαδικτυακών Εφαρμογών - Θεωρία και Πράξη (2009), Παναγιώτης Δ. Κεντερλής, Π.Δ. Κεντερλής
2. Δίκτυα και Διαδίκτυα Υπολογιστών και Εφαρμογές τους στο Internet - 4^η Αμερικάνικη Έκδοση (2007), Douglas E. Comer, Κλειδάριθμος

6.2 ΔΙΑΔΥΚΤΙΑΚΕΣ ΠΗΓΕΣ

1. <http://codex.wordpress.org/>
2. <http://el.wordpress.org>
3. <http://api.jquery.com/>
4. <http://jqueryui.com/>
5. <http://www.w3schools.com/>
6. <http://wordpress.org/support/>
7. <http://wordpress.stackexchange.com/>
8. <http://en.wikipedia.org/wiki/Internet>
9. <http://el.wikipedia.org/wiki/Διαδίκτυο>
10. http://en.wikipedia.org/wiki/World_Wide_Web
11. http://el.wikipedia.org/wiki/Παγκόσμιος_Ιστός
12. http://en.wikipedia.org/wiki/Web_page
13. <http://el.wikipedia.org/wiki/Ιστοσελίδα>
14. <http://el.wikipedia.org/wiki/JavaScript>
15. Είδη Συστημάτων Διαχείρισης Περιεχομένου:
http://pacific.jour.auth.gr/content_management_systems/eidi.htm
16. Συστήματα Διαχείρισης Περιεχομένου:
<http://www.kepa.gov.cy/diktiouthite/Portal/PortalDocuments.aspx?DocumentId=a5d27b95-5b46-4a56-a535-0c2324141d42>
17. Understanding WordPress Website Architecture:
<https://www.optimizesmart.com/wordpress-ninja-15-minutes/>

18. WordPress Custom Post Types. Ανάλυση και tutorial:

http://webapptester.com/custom-post-types-to-wordpress/#_o_8220post_type8221_wordpress

19. The Complete Guide To Custom Post Types:

<http://www.smashingmagazine.com/2012/11/complete-guide-custom-post-types/>

20. A Tutorial On Using AJAX in WordPress:

<http://www.makeuseof.com/tag/tutorial-ajax-wordpress/>

