

Α.Τ.Ε.Ι ΠΑΤΡΩΝ  
ΤΜΗΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

# Πτυχιακή Εργασία

---

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ PORTAL ΓΙΑ ΤΟ ΤΜΗΜΑ ΕΣ&ΠΣ ΒΑΣΙΣΜΕΝΟ ΣΕ  
ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ (FREE AND OPEN SOURCE)

Εισηγητής: Στάμος Κωνσταντίνος  
Χάρης Υφαντής 1350  
Εμμανουήλ Μαμαλάκης 1651  
Κωνσταντίνος Γούτας 1449

10/10/2010

## Περιεχόμενα

ΕΙΣΑΓΩΓΗ .....	4
Τι είναι τα portal;.....	4
Τύποι portal.....	4
Τι σημαίνει τεχνολογίες free και open source; .....	6
ΣΧΕΔΙΑΣΜΟΣ:.....	10
Τι χρειάζεται ένα Web Site;.....	10
ΥΛΟΠΟΙΗΣΗ: .....	12
Τι είναι CMS;.....	12
Γιατί Joomla! CMS; .....	14
Τι είναι τα AMPs;.....	15
Γιατί XAMPP; .....	17
Τι είναι ο Apache;.....	18
Τι είναι το Apache Software Foundation. ....	18
Τι είναι το Apache HTTP server Project;.....	19
Πώς έφτασε ο Apache HTTP server ως εδώ;.....	19
Ανάπτυξη του Apache. ....	21
Γιατί ο Apache είναι δωρεάν ;.....	22
Πως λειτουργεί ο Apache httpd;.....	24
Αρχιτεκτονική του διακομιστή ιστοσελίδων Apache .....	25
Τι είναι η PHP; .....	33
Πως λειτουργεί η PHP; .....	33
Η ιστορία της PHP.....	35
PHP/FI.....	35
PHP 3 .....	36
PHP 4 .....	37
PHP 5 .....	37
Χαρακτηριστικά της PHP 5 .....	37
Νέο αντικειμενοστραφές μοντέλο .....	37
Νέα αντικειμενοστραφής χαρακτηριστικά .....	39
1. Κοινό/Μυστικό/Προστατευόμενοι (public/private/protected) τροποποιητές πρόσβασης για μεθόδους και ιδιότητες. ....	39
2.  Ενιαίο όνομα κατασκευαστή (constructor) __construct().....	40
3.  Υποστήριξη καταστροφέα (destructor) αντικειμένων ορίζοντας μια μέθοδο __destructor().....	40

4.	Διεπαφές (interfaces).....	40
5.	Τελεστής instanceof .....	41
6.	Final μέθοδοι.....	41
7.	Final τάξεις.....	41
8.	Ρητή αντιγραφή (κλωνοποίηση) αντικειμένων.....	41
9.	Σταθερές αντικειμένων .....	42
10.	Στατικά Μέλη (members).....	42
11.	Στατικές μέθοδοι (methods) .....	42
12.	Αφαιρετικές (abstract) τάξεις.....	43
13.	Αφαιρετικές (abstract) μέθοδοι .....	43
14.	Συμβουλές για τύπο Αντικειμένου (Class type hints) .....	43
15.	Υποστήριξη για σπάσιμο αναφοράς (dereferencing) αντικειμένου οι οποίες επιστρέφονται από μεθόδους. ....	43
16.	Σαρωτές (Iterators).....	44
17.	__autoload() .....	44
	Άλλα νέα χαρακτηριστικά .....	44
1.	Χειρισμός εξαιρέσεων .....	44
2.	Foreach με αναφορές.....	45
3.	Προεπιλεγμένες τιμές για τις παραμέτρους με αναφορά .....	45
	Γενικές αλλαγές στην PHP .....	46
	XML και Web Services .....	46
	Τα θεμέλια.....	46
	SAX.....	46
	DOM .....	46
	XSLT .....	47
	SimpleXML.....	47
	SOAP .....	48
	Νέα επέκταση MySQLi (MySQL Improved) .....	48
	Επέκταση SQLite.....	49
	Επέκταση Tidy .....	49
	Επέκταση Perl.....	49
	Λοιπά νέα χαρακτηριστικά στην PHP 5:.....	49
	Νέος διαχειριστής μνήμης .....	49
	Κατάργηση υποστήριξης για Windows 95.....	49

Πώς το ΣΔΒΔ MySQL έφτασε ως εδώ;.....	50
MySQL και mSQL .....	50
Η ιστορία του ΣΔΒΔ mSQL .....	50
Ο David Hughes και το σύστημα Minerva .....	51
Από τον μεταφραστή PostQUEL στο Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS). .....	52
Η ιστορία του ΣΔΒΔ MySQL.....	52
Πώς λειτουργεί το ΣΔΒΔ MySQL;.....	54
Τα χαρακτηριστικά και τα πλεονεκτήματα της MySQL.....	55
Επεκτασιμότητα και Ευελιξία (Scalability and Flexibility).....	55
Υψηλή απόδοση (High Performance).....	55
Υψηλή διαθεσιμότητα (High Availability) .....	55
Υποστήριξη εύρωστων συναλλαγών (Robust Transactional Support) .....	55
Ισχυρά σημεία αποθήκης δεδομένων και ιστού (Web and Data Warehouse Strengths) ..	55
Ισχυρή προστασία δεδομένων (Strong Data Protection) .....	56
Περιεκτική ανάπτυξη εφαρμογών (Comprehensive Application Development).....	56
Ευκολία στη διαχείριση (Management Ease) .....	56
Εγκατάσταση του XAMPP.....	58
Εγκατάσταση του Joomla! CMS .....	67
Διαχείριση (Administration).....	79
Βιβλιογραφία .....	99

# ΣΤΟΧΟΙ ΠΤΥΧΙΑΚΗΣ

---

## ΕΙΣΑΓΩΓΗ

### **Τι είναι τα portal;**

Το portal είναι μια ιστοσελίδα που χρησιμοποιείται ως πύλη για άλλους προορισμούς ή δραστηριότητες στο Διαδίκτυο. Ο σκοπός του είναι να καθοδηγήσει τους επισκέπτες σε οτιδήποτε θέλουν να βρουν on-line και να παρέχει όσο το δυνατόν περισσότερες υπηρεσίες όπως δωρεάν υπηρεσίες ηλεκτρονικού ταχυδρομείου, online chat forums, εξατομικευμένες ειδήσεις, αγορές, υπηρεσίες αναζήτησης και πολλά άλλα.

Τα portal δεν είναι πανάκεια λύση («one size fits all»). Παρ' όλο που ο όρος portal συχνά χρησιμοποιείται ως συνώνυμο με τα επιχειρησιακά portal πληροφοριών, αλλά στην πραγματικότητα υπάρχουν πολλοί διαφορετικοί τύποι portal, ο καθένας προσαρμοσμένος για τις ανάγκες της κάθε επιχείρησης ή οργανισμού.

### **Τύποι portal.**

#### **Διαδικτυακές πύλες αναζήτησης (Web searching portal)**

Τα portal ξεκίνησαν στο Διαδίκτυο. Αυτά τα πρώτα portal, όπως είναι το yahoo, παρείχαν συγκεντρωμένο περιεχόμενο, συνήθως σαν σειρά προκαθορισμένων συνδέσμων μαζί με μία μηχανή αναζήτησης. Καθώς η επιβίωση τους εξαρτιόνταν από τη διαφήμιση, έπρεπε να προσελκύουν μεγάλο αριθμό ατόμων στον ιστότοπο τους.

#### **Portal ηλεκτρονικού εμπορίου (e-Commerce portal)**

Τα portal ηλεκτρονικού εμπορίου (e-Commerce) και ηλεκτρονικού επιχειρείν (e-Business) φιλοξενούσαν την ανταλλαγή πληροφοριών με εξωτερικούς συνεργάτες, πελάτες και προμηθευτές. Συνήθως έχουν ένα συστατικό (component) επεξεργασίας συναλλαγών, παρέχουν πληροφορίες πάνω σε προϊόντα και υπηρεσίες και συχνά περιέχουν χαρακτηριστικά διαχείρισης της εφοδιαστικής αλυσίδας. Τα portal e-Commerce σκοπεύουν στο να αυξήσουν την αξία της συνεργασίας (σχέσης) μειώνοντας παράλληλα το κόστος.

#### **Portal αυτοεξυπηρέτησης (Self-service portal)**

Τα portal αυτοεξυπηρέτησης επιτρέπουν τους εργαζόμενους, πελάτες και προμηθευτές να έχουν πρόσβαση σε πληροφορίες που αφορούν τους ίδιους και να διενεργούν σε συγκεκριμένες επιχειρηματικές διαδικασίες με τέτοιο τρόπο ώστε να είναι προσαρμοσμένες στις δικές τους ανάγκες. Τα portal συνήθως δικαιολογούνται

από την άποψη μείωσης του μεγάλου κόστους (hard cost) της επιχείρησης μέσω των επιλογών αυτοεξυπηρέτησης.

### **Portal επιχειρηματικής ευφυΐας (Business intelligence portal)**

Τα portal επιχειρηματικής ευφυΐας ή αλλιώς portal αποφάσεων (decision portal), ενδυναμώνουν τους χρήστες στη δική τους διαδικασία λήψης αποφάσεων. Όχι μόνο επιτρέπει στους χρήστες να ερωτούν (query) και να αναφέρουν (report) σε πολλαπλές αποθήκες δεδομένων (data stores) αλλά έχουν ενσωματωμένα εργαλεία που παρέχουν στοχοθετημένες αναφορές σε ομάδες τελικών χρηστών και ατόμων. Τα portal επιχειρηματικής ευφυΐας είναι πιθανό να καλύπτουν όπως ανάγκες των στελεχών που ψάχνουν πληροφορίες υψηλού επιπέδου που αφορούν τις επιδόσεις της εταιρίας, τις ανάγκες γενικών υπαλλήλων που χρειάζονται αναφορές ad-hoc καθώς και τις ανάγκες αυτών εντός του οργανισμού των οποίων ο ειδικός ρόλος είναι να εξορύξουν και να αναλύουν δεδομένα.

### **Portal συνεργασίας (Collaboration portal)**

Τα portal συνεργασίας καθιστούν ικανή μία γεωγραφική διασπορά εργατικού δυναμικού να αλληλεπιδρά γύρω από έργα (projects) και επιχειρηματικές λειτουργίες μέσω κοινής πρόσβασης ή μέσω συσπείρωσης σε ένα σημείο. Τα portal συνεργασίας διασφαλίζουν ότι το μη-τεχνικό προσωπικό μπορεί να διαχειρίζεται συνεδρίες συνεργασίας (collaboration sessions), ομάδες εργασίας ή τοποθεσίες, συμπεριλαμβανομένου τη δημιουργία και παραμετροποίηση χρηστών, δημιουργία σελίδων, παρουσιάσεων κ.λπ. Τα portal συνεργασίας προσφέρουν γενικά εργαλεία όπως δωμάτια συνομιλίας, πίνακες συζητήσεων (white boards) και ροές εμφωλευμένων συζητήσεων (threaded discussion streams) μαζί με τρόπους να μοιράζονται δεδομένα (π.χ. χάρτες, έγγραφα, κ.λπ.).

### **Portal επιχειρησιακών πληροφοριών (Enterprise information portal)**

Απ' όλους τους διάφορους τύπους portal, τα portal επιχειρησιακών πληροφοριών είναι τα πιο πολύπλοκα. Ο σκοπός είναι μια άκρως προσαρμοσμένη, μοναδική εμπειρία για κάθε χρήστη. Αυτό επιτυγχάνετε δένοντας (ενοποιώντας) τα απαρχαιωμένα και ανόμοια συστήματα και βάζοντας τα μαζί στην επιχειρηματική διαδικασία. Σε αυτή την περίπτωση οι διαδικασίες πολλαπλών βημάτων (multi-step processes) υποστηρίζονται από πολύπλοκες ροές εργασίας (workflows) και επιχειρηματικούς κανόνες.

### **Portal τηλεκπαίδευσης (e-Learning portal)**

Πλέον όχι μόνο στον τομέα των ακαδημαϊκών ιδρυμάτων, τα portal τηλεκπαίδευσης εστιάζονται στο να οδηγούν τους μαθητές στην ευρύτερη έννοια μέσα από δομημένη μαθησιακή εμπειρία. Τα portal αυτά δοκιμάζουν τις ικανότητες και παρέχουν ανατροφοδότηση στο μαθητή με ένα εξατομικευμένο και εμπιστευτικό τρόπο. Μπορούν επίσης να αλληλεπιδρούν με άλλα συστήματα και επιχειρηματικές διαδικασίες για να παρέχουν πλαισιωμένη εκπαίδευση και βοήθεια.

### **Portal επικοινωνίας (Communication portal)**

Τα portal επικοινωνίας συγκεντρώνουν διάφορους τρόπους επικοινωνίας σε ένα μόνο μέρος. Ενοποιώντας το ηλεκτρονικό ταχυδρομείο, τη φωνή, τη φορητότητα, διαδικτυακές ροές, κ.λπ., με τέτοιο τρόπο που επιτρέπει την πρόσβαση και τον έλεγχο από πολλαπλές διασυνδέσεις και τοποθεσίες οποιαδήποτε στιγμή. Το άτομο-χρήστης μπορεί μετά να τα διαρθρώσει, διαλέγοντας για παράδειγμα, να λαμβάνει και να διαχειρίζεται τις κρίσιμες επικοινωνίες ανεξαρτήτου που βρίσκονται ή με τι συσκευή γίνονται.

### **Τι σημαίνει τεχνολογίες free και open source;**

Το free and open source software (F/OSS ή FOSS) ή αλλιώς free/libre/open source software (FLOSS) είναι το λογισμικό που ελεύθερα αδειοθετεί τον χρήστη να χρησιμοποιεί, να αλλάζει, να μελετά και να βελτιώνει το σχεδιασμό του, μέσω της διαθεσιμότητας του ίδιου του πηγαίου κώδικα (source code). Αυτή η προσέγγιση έχει αποκτήσει δυναμικότητα και αποδοχή καθώς τα ενδεχόμενα αυτά οφέλη αναγνωρίζονται όλο και πιο πολύ από ατομικούς και εταιρικούς «παίχτες».

Στο πλαίσιο του ελεύθερου και ανοιχτού λογισμικού (free and open source software), το «free» αναφέρεται στην ελευθερία του να αντιγράψεις και να επαναχρησιμοποιήσεις το λογισμικό και όχι στην χρηματική αξία του λογισμικού. Το Ίδρυμα Ελεύθερου Λογισμικού (Free Software Foundation) είναι ένας οργανισμός που τάσσεται υπέρ του μοντέλου ελεύθερου λογισμικού (free software model), προτείνει ότι για να καταλάβει κάποιος τη γενική ιδέα του free and open source software, πρέπει να σκεφτεί τη φράση «**free as in free speech, not as in free beer**» που σημαίνει «ελεύθερο όπως η ελευθερία λόγου και όχι όπως ελεύθερη (δωρεάν) μπύρα».

Το free and open source λογισμικό είναι ένας όρος που εμπεριέχει τον όρο free software (ελεύθερο λογισμικό) και τον όρο open source software (λογισμικό ανοιχτού κώδικα). Οι οποίοι ανεξάρτητα του ότι περιγράφουν διαφορετικά αναπτυξιακά μοντέλα, έχουν διαφορετική κουλτούρα και φιλοσοφία. Ο όρος free software εστιάζει στις φιλοσοφικές ελευθερίες που δίνει στον χρήστη ενώ ο όρος open source software εστιάζει στην αντιληπτή δύναμη του p2p (peer-to-peer) μοντέλου

ανάπτυξης του. Το FOSS είναι ένας όρος ο οποίος μπορεί να χρησιμοποιηθεί χωρίς ιδιαίτερη προκατάληψη από οποιαδήποτε πολιτική πεποίθηση.

Οι άδειες χρήσης free software και open source χρησιμοποιούνται από πολλά πακέτα προγραμμάτων. Οι άδειες έχουν σημαντικές διαφορές μεταξύ τους, οι οποίες καθρεφτίζουν τις διαφορές τους με τρόπο με τον οποίο το λογισμικό μπορεί να χρησιμοποιηθεί και να διανεμηθεί. Επίσης αντικατοπτρίζουν τη διαφορετική φιλοσοφία πίσω από αυτές.

### **Η ιστορία του free and open source software.**

Τις δεκαετίες του '50, '60 και '70, ήταν φυσιολογικό για τους χρήστες των υπολογιστών να έχουν τις ελευθερίες που τους παρείχε το ελεύθερο λογισμικό. Το λογισμικό μοιράζονταν κοινώς από άτομα-χρήστες που χρησιμοποιούσαν υπολογιστές και από κατασκευαστές υλισμικού (hardware) οι οποίοι ήταν ευτυχείς που το ανθρώπινο δυναμικό έφτιαχνε λογισμικό που αξιοποιούσε τη δική τους δουλειά. Οργανισμοί χρηστών και προμηθευτών είχαν συσταθεί για να διευκολύνουν την ανταλλαγή λογισμικού. Στα τέλη της δεκαετίας του '60 η αλλαγή ήταν αναπόφευκτη. Το κόστος του λογισμικού αυξάνονταν δραματικά, η αναπτυσσόμενη βιομηχανία λογισμικού ανταγωνιζόταν με τα πακέτα προϊόντων λογισμικού (bundled software products) των κατασκευαστών υλισμικού. Τα μισθωμένα μηχανήματα που χρησιμοποιούνταν για παραγωγή λογισμικού χρειάζονταν και αυτά υποστήριξη λογισμικού, παρ' όλο που δεν παρήγαγαν έσοδα διότι διανέμονταν δωρεάν. Ακόμα μερικοί πελάτες που είχαν τη δυνατότητα να καλύψουν τις ανάγκες τους δεν ήθελαν την επιβάρυνση του κόστους του δωρεάν λογισμικού να συμπεριλαμβάνεται στο κόστος του υλισμικού που αγόραζαν. Στη δίκη των Ηνωμένων Πολιτειών ενάντια της IBM, που κατατέθηκε τον Ιανουάριο του 1969, η κυβέρνηση των Ηνωμένων Πολιτειών χρέωσε στην IBM ότι τα πακέτα λογισμικού (bundled software) ήταν αντιανταγωνιστικές. Παρ' όλο που ένα ποσοστό λογισμικών μπορεί ακόμα να είναι ελεύθερο, υπάρχει ένα αναπτυσσόμενο ποσοστό λογισμικού το οποίο ήταν μόνο προς πώληση. Στα τέλη της δεκαετίας του '70 και αρχές του '80, η βιομηχανία λογισμικού ξεκίνησε να χρησιμοποιεί τεχνικά μέτρα για να αποφύγει τη μελέτη και τη μετατροπή των λογισμικών από τους χρήστες όπως διανομή μόνο του δυαδικού (εκτελέσιμα) αντίγραφου του λογισμικού τους. Το 1980 ο νόμος της πνευματικής ιδιοκτησίας αναπτύχθηκε ώστε να συμπεριλαμβάνει και τα λογισμικά υπολογιστών.

Το 1983, ο Richard Stallman, πολυετές μέλος της κοινότητας των hacker του εργαστηρίου τεχνητής νοημοσύνης του MIT (MIT Artificial Intelligence Laboratory), ανακοίνωσε το GNU project, λέγοντας ότι ένιωθε απογοητευμένος από την διαστρέβλωση της κουλτούρας της βιομηχανίας υπολογιστών και των χρηστών της. Η ανάπτυξη λογισμικού για το λειτουργικό σύστημα του GNU ξεκίνησε τον Ιανουάριο του 1984 και το Ίδρυμα Ελεύθερου Λογισμικού (Free Software



Foundation) ιδρύθηκε το 1985. Ένα άρθρο το οποίο επισήμανε το project και τους στόχους του δημοσιεύτηκε το Μάρτιο του 1985 με τίτλο GNU Manifesto. Το manifesto αυτό εστιάστηκε στην φιλοσοφία του ελεύθερου λογισμικού. Ο Richard Stallman ανέπτυξε επίσης τον Ορισμό του Ελεύθερου Λογισμικού (Free Software Definition) και την έννοια του «copyleft», που σχεδιάστηκε για να εξασφαλίσει ελεύθερο λογισμικό για όλους.



**Εικόνα 1: Ο Richard Stallman**

Ο πυρήνας (kernel) των Linux, ξεκίνησε από τον Σουηδικής καταγωγής Linus Torvalds, κυκλοφόρησε σαν ελεύθερα τροποποιήσιμος πηγαίος κώδικας το 1991. Η άδεια χρήσης δεν ήταν ακριβώς μία άδεια χρήσης ελεύθερου λογισμικού, αλλά στην έκδοση 0.12 το Φεβρουάριο του 1992 άλλαξε την άδεια χρήσης του project του σε άδεια χρήσης GNU-GPL (GNU General Public License). Περίπου όπως των Unix, ο πυρήνας του Torvalds προσέγγισε το ενδιαφέρον εθελοντών προγραμματιστών.



**Εικόνα 2: Ο Linus Torvalds δημιουργός του Linux**

Το 1997 , ο Eric Raymond δημοσίευσε το «The Cathedral and the Bazaar», μία στοχαστική ανάλυση της κοινωνίας των hacker και των αρχών του ελεύθερου λογισμικού. Το δημοσίευμα αυτό τράβηξε το ενδιαφέρον του κόσμου στις αρχές του 1998 και ήταν ένας παράγοντας στην παρακίνηση της Netscape Communications Corporation να ανακοινώσει τη δημοφιλή Διαδικτυακή σουίτα Netscape Communicator ως ελεύθερο λογισμικό. Αυτός ο κώδικας είναι πλέον γνωστός ως Mozilla Firefox και Thunderbird.

Η ενέργεια αυτή της Netscape παρότρυνε τον Raymond και άλλους να αναζητήσουν στο πώς να φέρουν τις αρχές και τα οφέλη του ελεύθερου λογισμικού στην εμπορική βιομηχανία λογισμικού. Κατέληξαν στο συμπέρασμα ότι ο κοινωνικός ακτιβισμός του FSF δεν έγινε ελκυστικός σε εταιρίες σαν τη Netscape και έψαξαν τρόπο να επαναπροωθήσουν το κίνημα του ελεύθερου λογισμικού για να δώσουν έμφαση στις δυνατότητες των επιχειρήσεων για μοίρασμα του πηγαίου κώδικα. Η καινούργια ονομασία που επέλεξαν ήταν «open source» , και σύντομα ο Bruce Perens, ο εκδότης Tim O'Reilly και ο Linus Torvalds, συνυπέγραψαν στην επανακυκλοφόρηση του. Η Πρωτοβουλία Ανοιχτού Πηγαίου Κώδικα (Open Source Initiative) ιδρύθηκε το 1998 ενθαρρύνοντας τη χρήση του νέου όρου και ξεκίνησε τις αρχές του open source.

Ο Ορισμός του Ανοιχτού Λογισμικού που είναι γραμμένος από τον Richard Stallman και έχει δημοσιευτεί από το Ίδρυμα Ελεύθερου Λογισμικού (FSF), ορίζει το ελεύθερο λογισμικό ως θέμα ελευθερίας, και όχι τιμής. Η παλαιότερη γνωστή δημοσίευση του ορισμού αυτού ήταν στην έκδοση του Φεβρουαρίου του 1986, της πλέον μη συνεχιζόμενης GNU Bulletin έκδοσης του FSF. Η αρχική πηγή είναι στο

τμήμα φιλοσοφίας στην ιστοσελίδα του GNU project. Από τον Απρίλιο του 2008 έχει δημοσιευτεί εκεί σε 39 γλώσσες.

Ο Ορισμός του Ανοιχτού Λογισμικού χρησιμοποιείται από το Ίδρυμα Ανοιχτού Λογισμικού (OSF) για να καθορίσει πότε μία άδεια χρήσης λογισμικού μπορεί να θεωρείται ανοιχτού λογισμικού. Ο ορισμός είχε βασιστεί στο Debian Free Software Guidelines και είχε γραφτεί και υιοθετηθεί πρωτίστως από τον Bruce Perens.

## **ΣΧΕΔΙΑΣΜΟΣ:**

### **Τι χρειάζεται ένα Web Site;**

Μία βασική ιστοσελίδα δεν είναι τίποτα παραπάνω από μία συλλογή αρχείων κειμένου που μπορούν να επεξεργαστούν σε ένα πρόγραμμα τόσο απλό όσο το Textpad στα Windows ή το Simpletext στα Mac OS, μαζί με ένα αριθμό από αρχεία εικόνων, που μπορούν να εμφωλευτούν σε αυτές τις σελίδες. Οι σελίδες είναι γραμμένες σε μία γλώσσα που λέγεται HTML, η οποία περιγράφει τον τρόπο με τον οποίο το περιεχόμενο πρέπει να παρουσιάζεται στον φυλλομετρητή. Υπάρχει ένα πλήθος από εργαλεία, για παράδειγμα το Adobe Golive, το Macromedia (πλέον Adobe) Dreamweaver, το Microsoft Frontpage και άλλα, που βοηθούν στη διαδικασία δημιουργίας αυτών των σελίδων και επιτρέπουν στο χρήστη να χρησιμοποιεί μια γραφική διεπαφή για να μορφοποιεί τις σελίδες από το να πληκτρολογεί απ' ευθείας τον κώδικα HTML. Στις περισσότερες περιπτώσεις, μία σειρά από στατικές σελίδες ανεπτυγμένες κατ' αυτόν το τρόπο, επαρκούν για να παραδοθεί το μήνυμα στο δίκτυο. Παρόλα αυτά αν κάποιος είναι έτοιμος να προχωρήσει στο επόμενο βήμα και παραδώσει ένα δυναμικό περιεχόμενο στο δίκτυο το πιθανότερο είναι να χρειαστεί μία πιο εξειδικευμένη ιστοσελίδα.

### **Τα βασικά μιας ιστοσελίδας βασισμένη σε βάση δεδομένων**

Όταν κάποιος χρησιμοποιεί μία βάση δεδομένων σε συνδυασμό με ιστοσελίδες του δίνεται η δυνατότητα να παραδίδει κείμενα, εικόνες και άλλο υλικό που είναι βασισμένο σε συγκεκριμένες απαιτήσεις για πληροφορίες από ανθρώπους που επισκέπτονται την ιστοσελίδα. Οπότε αντί για στατικές σελίδες οι οποίες πάντα παρέχουν τις ίδιες πληροφορίες, για παράδειγμα μία λίστα του προσωπικού μίας επιχείρησης, μπορεί κάποιος να προσφέρει δυναμικά παραγόμενες ιστοσελίδες οι οποίες θα μπορούν να εξυπηρετήσουν καλύτερα τις ανάγκες όσων επισκέπτονται την ιστοσελίδα. Ένα παράδειγμα μίας δυναμικά παραγόμενης ιστοσελίδας θα μπορούσε να είναι μία βάση δεδομένων αναζήτησης προσωπικού. Πιθανότατα ο πιο αναγνωρισμένος τύπος ιστοσελίδας όσον αφορά βάσεις δεδομένων είναι η μηχανή αναζήτησης. Βάζοντας σε ένα πεδίο κάποιο κείμενο και πατώντας ένα κουμπί εμφανίζεται μία σειρά επιλογών που ενδεχομένως να είναι όσα ψάχνει ένας χρήστης. Αυτή η σελίδα παράγεται δυναμικά βάση του αιτήματος που έχει κάνει ο

χρήστης και κατά πάσα πιθανότητα οι πληροφορίες αποθηκεύονται σε μία βάση δεδομένων.

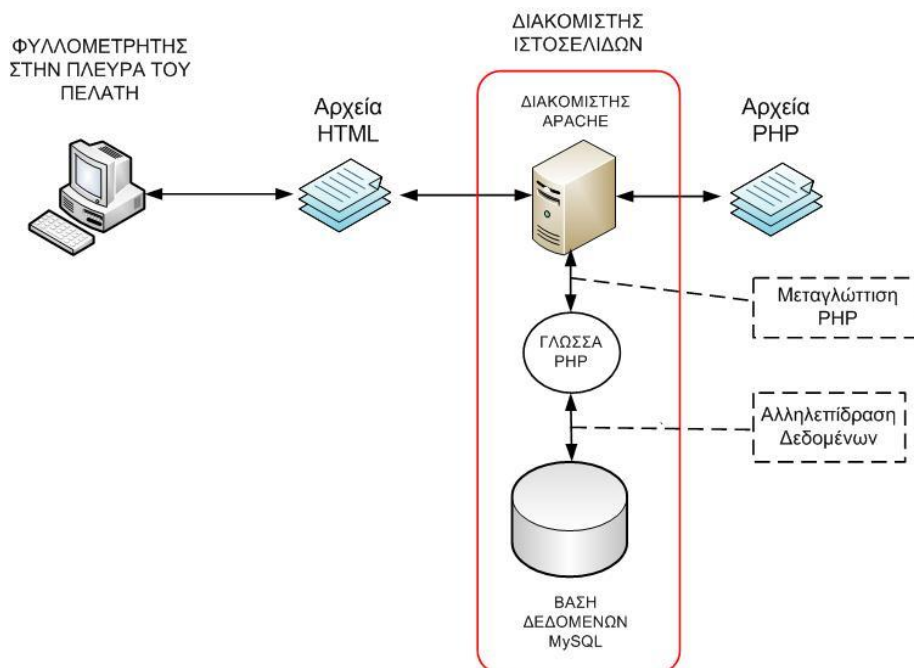
Η διαδικασία αυτή είναι στην πραγματικότητα εφάμιλλη με το να ψάχνει κανείς στη βάση δεδομένων του οργανισμού όπου εργάζεται, για άτομα σε μία συγκεκριμένη πολιτεία ή για δότες οι οποίοι έχουν δώσει παραπάνω από ένα συγκεκριμένο ποσό στο παρελθόν. Σε αυτή την περίπτωση όμως, χρησιμοποιούμε μία δικτυακή διεπαφή (Web interface) για μία βάση δεδομένων, αντί για την συνηθισμένη διεπαφή που θα χρησιμοποιούσαμε αν η βάση δεδομένων έτρεχε στον υπολογιστή μας.

Για παράδειγμα, θα ήταν εφικτό για κάποιον να τρέξει μία βάση δεδομένων στον υπολογιστή του για να παρακολουθεί τις εξαγωγές προϊόντων. Θα μπορούσε να περιέχει πληροφορίες για το υλικό που έχει εξαχθεί, σε ποιόν έχει αποσταλεί, πότε και πως θα σταλεί και σε πιο στάδιο βρίσκεται η εξαγωγή. Μπορεί κανείς να τρέξει αυτή τη βάση δεδομένων στο σύστημα διαχείρισης βάσεων δεδομένων (DBMS) Microsoft Access στον υπολογιστή του και ίσως του φάνει χρήσιμο, αλλά θα μπορούσε να σκεφτεί επίσης να κάνει τη βάση δεδομένων του προσβάσιμη και στο Διαδίκτυο προκειμένου να μπορούν οι πελάτες του να βρίσκουν πληροφορίες για τις καταστάσεις των αποστολών τους ανά πάσα στιγμή, όπως κάνουν πολλές εταιρίες μεταφορών.

Για να επεξεργαστεί κάποιος τις λέξεις που έχει βάλει σε μία μηχανή αναζήτησης ή οποιαδήποτε δεδομένα περασμένα σε μία δυναμική σελίδα (π.χ. έναν αριθμό εξαγωγής περασμένη σε ένα ιστότοπο εξαγωγών), θα πρέπει να γραφτεί ένα πρόγραμμα (συνήθως ονομάζεται script) που να μπορεί να εμφανίζει τα αποτελέσματα. Υπάρχουν πολλές διαφορετικές γλώσσες προγραμματισμού που χρησιμοποιούνται συνήθως γι' αυτό το σκοπό. Δύο από αυτές, που είναι καταλληλότερες για αλληλεπίδραση με βάσεις δεδομένων στο Διαδίκτυο, είναι η ColdFusion και η PHP, παρ' όλο που υπάρχουν και άλλες γλώσσες που λειτουργούν εξίσου καλά για διαφορετικές καταστάσεις όπως η Perl, Python, JavaScript και VBScript. Η απόφαση σχετικά με το ποια γλώσσα πρέπει να χρησιμοποιηθεί για μια συγκεκριμένη εφαρμογή, αφορά τον προγραμματιστή που αναπτύσσει την εφαρμογή. Αν ο προγραμματιστής σκέφτεται να αναπτύξει την ιστοσελίδα του με περιεχόμενο βασισμένο σε βάσεις δεδομένων θα πρέπει να λάβει υπόψη ότι οι εταιρίες που φιλοξενούν τις ιστοσελίδες του δεν θα του παρέχουν απαραίτητα όλες αυτές τις επιλογές ή μπορεί να τον χρεώσουν παραπάνω για κάποιες από αυτές.

Υπάρχει επίσης ένας αριθμός επιλογών σχετικά με τις βάσεις δεδομένων για την αποθήκευση πληροφοριών που συνθέτουν τις δυναμικές σελίδες. Μερικές εταιρίες που φιλοξενούν ιστότοπους μπορούν να επιτρέψουν σε κάποιον να χρησιμοποιήσει μια βάση δεδομένων του DBMS Microsoft Access για να τρέξει την ιστοσελίδα του ή το ανεπτυγμένο DBMS της Microsoft γνωστό ως SQL server. Ένα άλλο ευρέως

διαθέσιμο DBMS για ανάπτυξη διαδικτυακών εφαρμογών είναι το DBMS MySQL, που είναι ένα δωρεάν σύστημα διαχείρισης βάσης δεδομένων το οποίο «τρέχει» σε πολλά διαφορετικά λειτουργικά συστήματα, σε αντίθεση με τα DBMS Access και SQL server, που περιορίζονται στο να τρέχουν σε εξυπηρετητές βασισμένους σε Windows. Όπως και προηγουμένως έτσι και τώρα την απόφαση για το ποια βάση δεδομένων πρέπει να χρησιμοποιηθεί θα ήταν προτιμότερο να την πάρει ο προγραμματιστής. Η απόφαση αυτή μπορεί επίσης να αλλάξει ανάλογα με το τι είναι διαθέσιμο από την εταιρία που φιλοξενεί την ιστοσελίδα.

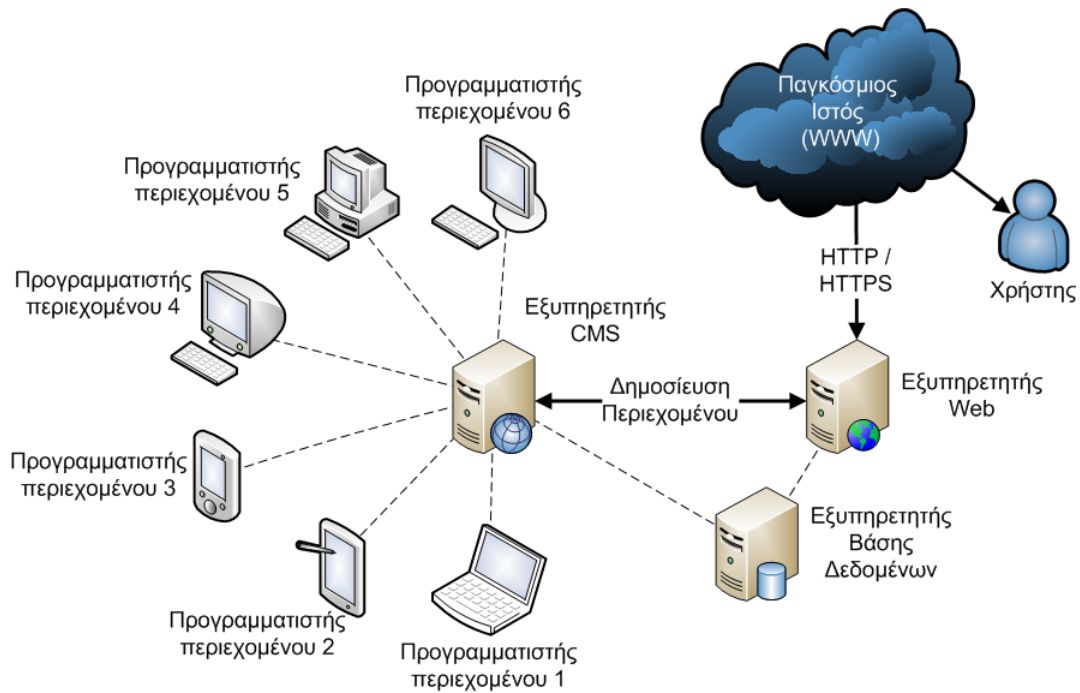


Εικόνα 3: Διακομιστής Ιστοσελίδων

## ΥΛΟΠΟΙΗΣΗ:

### Τι είναι CMS;

Ένα CMS (Σύστημα Διαχείρισης Περιεχομένου) χρησιμοποιείται για τον έλεγχο και την επεξεργασία περιεχομένου. Το περιεχόμενο αυτό περιλαμβάνει ηλεκτρονικά αρχεία, εικόνες, βίντεο, αρχεία ήχου, ηλεκτρονικά έγγραφα και κείμενα web (δικτυακά). Η κεντρική ιδέα του CMS είναι να κάνει αυτά τα αρχεία διαθέσιμα για επεξεργασία εσωτερικά ή μέσω του Διαδικτύου. Ένα CMS μπορεί να χρησιμοποιηθεί και για αρχειοθέτηση, πολλές εταιρίες χρησιμοποιούν CMS για την οργάνωση και αποθήκευση αρχείων σε κοινή μορφή (public form). Οι εταιρίες που χρησιμοποιούν CMS μπορούν να μοιράζονται υλικό με άλλους εύκολα καθώς τα περισσότερα συστήματα βασίζονται σε διακομιστές. Τα CMS είναι δημοφιλής λύση για τη δημιουργία custom portal. Στο παρακάτω σχήμα ο εξυπηρετητής CMS, ο εξυπηρετητής Βάσης Δεδομένων και ο εξυπηρετητής Web (Διακομιστής Ιστοσελίδων) μπορούν να βρίσκονται στο ίδιο φυσικό μηχάνημα (Εξυπηρετητή).

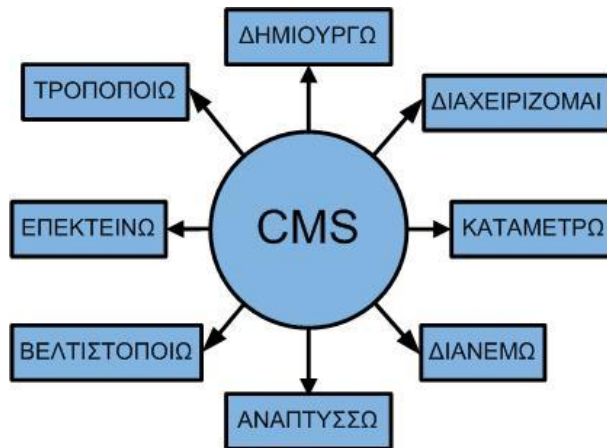


Εικόνα 4: Αλληλεπίδραση CMS και Παγκόσμιου Ιστού

Για παράδειγμα το Joomla! CMS που είναι διαθέσιμο στο Διαδίκτυο το καθιστά προσβάσιμο στους χρήστες από όλον τον κόσμο, καθώς υπάρχει πρόσβαση στο Διαδίκτυο. Το Joomla! CMS είναι ένα από τα πολλά συστήματα διαχείρισης περιεχομένου που περιέχουν μία « ροής εργασίας (workflow process) ». Το Joomla! είναι ένα web CMS και χρησιμοποιείται πρωτίστως για την έκδοση περιεχομένου σε μία συγκεκριμένη ιστοσελίδα.

### Γιατί χρειαζόμαστε ένα CMS;

Τα CMS είναι ένας τρόπος αποδοτικής οργάνωση εταιρικής ή μη πληροφορίας λόγω του εύκολου διαμερισμού περιεχομένου τόσο ενδοεταιρικά (intranet) όσο και στο Διαδίκτυο. Τα CMS μπορούν να θεωρηθούν ως μια ευέλικτη λύση αρχειοθέτησης όπως και μια εύκολη λύση για την δημιουργία συνεργατικών (collaborative) δικτυακών τόπων. Παρέχουν την δυνατότητα της διαχείρισης-συντήρησης ενός ιστότοπου από απλούς χειριστές χωρίς την απαίτηση για εμπλοκή ειδικού τεχνικού προσωπικού, δηλαδή παρέχει την ευκαιρία στον χειριστή του να επικεντρωθεί στο περιεχόμενο και όχι στην τεχνολογία. Τα CMS διευκολύνουν αυτοματοποιώντας εργασίες ρουτίνας (επαναλαμβανόμενες), παραδείγματος χάριν εφαρμόζουν την ίδια μορφοποίηση (layout) σε όλες τις ιστοσελίδες, τα μενού και γενικότερα η πλοήγηση αναπαράγεται επίσης αυτόματα, γενικά μας δίνουν την δυνατότητα να διαχειριζόμαστε τη δομή του ιστότοπου, τη εμφάνιση των δημοσιευμένων σελίδων καθώς και της πλοήγησης σε αυτές. Τα περισσότερα CMS παρέχουν απλά εργαλεία για επεξεργασία περιεχομένου (editors) σας το Microsoft Word.



Εικόνα 5: Γενικές λειτουργίες ενός CMS

### Γιατί Joomla! CMS;

Υπάρχουν πολλά CMS ανοικτού κώδικα. Είναι όλα σπουδαία εργαλεία, το καθένα με τις δικές του τυπικές χρήσεις και τα δικά του οφέλη.

Παρακάτω αναφέρονται μερικοί λόγοι γιατί κάποιος να θέλει να χρησιμοποιήσει το Joomla! CMS.

- Οι χρήστες τείνουν να χρησιμοποιούν το Joomla! επειδή είναι εύκολο στη χρήση. Είναι ένας ξεκάθαρος και φιλικός συνδετικός κρίκος. Κάνει εύκολη τη διαχείριση περιεχομένου, τη δημιουργία και δημοσίευση άρθρων, έτσι ώστε να διατηρείται η ιστοσελίδα ανανεωμένη, οποιαδήποτε στιγμή, οπουδήποτε, χρησιμοποιώντας το διακομιστή.
- Για να εισαχθούν νέα χαρακτηριστικά χρειάζονται μόνο μερικά κλικ. Υπάρχουν χιλιάδες επεκτάσεις διαθέσιμες, από συστήματα καταλόγου (menu) μέχρι και συστήματα σχολιασμού και forums.
- Είναι πολύ εύκολο να αλλάξει κάποιος την εμφάνιση της ιστοσελίδας, τα πρότυπα (templates) είναι άφθονα και μπορούν να εγκατασταθούν μέσα σε μερικά λεπτά, δίνοντας στην ιστοσελίδα μία νέα εμφάνιση.
- Το Joomla! είναι δραστήρια αναπτυσσόμενο και πολύ καλά υποστηριγμένο από μία τεράστια παγκόσμια κοινότητα χρηστών και προγραμματιστών. Ενημερώνεται συχνά, προσθέτοντας νέα χαρακτηριστικά, εργαλεία για την ασφάλεια και άλλες βελτιώσεις.

Εκτός από αυτά τα τυπικά θετικά στοιχεία, το Joomla!, διαθέτει όλα τα πλεονεκτήματα ενός CMS τελευταίας τεχνολογίας. Παρακάτω αναφέρονται μερικά παραδείγματα:

- Είναι πολύ εύκολο κάποιος να εισάγει ή να επεξεργαστεί το περιεχόμενο και επίσης να το έχει οργανωμένο (ακόμα και όταν είναι σε μεγάλη ποσότητα).
- Η διατήρηση ενημερωμένων υπερσυνδέσεων είναι αυτοματοποιημένη. Για παράδειγμα αν προσθέσει κάποιος μια καινούργια ιστοσελίδα σε μία κατηγορία, ένας σύνδεσμος θα εμφανιστεί αυτόματα στις σελίδες αυτές δείχνοντας εκείνη την κατηγορία.
- Δεν χρειάζεται κάποιος να διατηρήσει μόνος του την ιστοσελίδα. Άλλοι χρήστες μπορούν να προσθέσουν κείμενο, στοιχεία του μενού και πολλά άλλα.

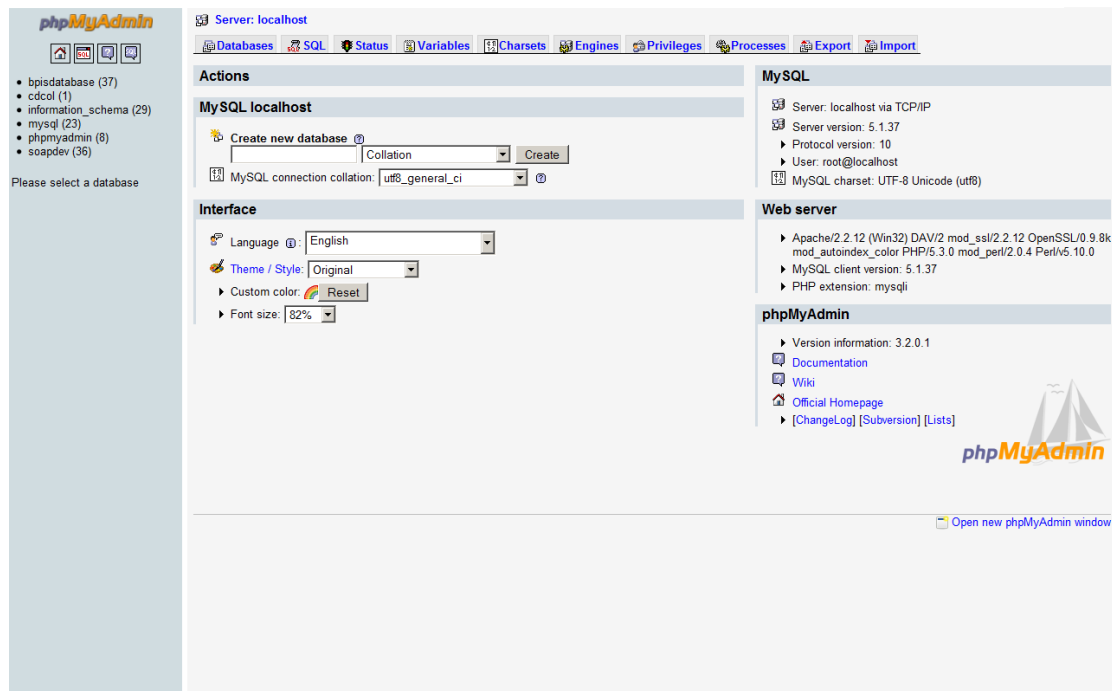
- Οι δυνατότητες αναβάθμισης και επέκτασης κάθε εφαρμογής βασισμένης σε Joomla! είναι απεριόριστες. Με μία πρόχειρη ματιά στον ιστότοπο JED (Joomla! Extension Directory) μετρήσαμε 5.804 προθέματα (extensions) που διατίθενται ήδη και καλύπτουν κάθε ανάγκη δημιουργίας εξειδικευμένων δικτυακών - διαδικτυακών εφαρμογών ή ιστότοπων

Στοιχεία δείχνουν ότι το Joomla! είναι η νούμερο ένα επιλογή για δημιουργούς ιστότοπων παγκοσμίως. Είναι κινητήριο δύναμη 20 εκατομμυρίων ιστοσελίδων παγκοσμίως και αυτό το νούμερο αυξάνεται ραγδαία διαρκώς, μέρα με τη μέρα. Το Joomla! CMS είναι ένα από τα μεγαλύτερα project ανοικτού κώδικα, που υποστηρίζεται από μία μεγάλη κοινότητα χρηστών και διαρκώς αναπτύσσεται από μια διεθνή ομάδα εθελοντών. Επίσης βασίζεται σε ανοιχτό κώδικα (open source), ελεύθερο λογισμικό (free source software) και αντικειμενοστραφή προγραμματισμό (object oriented programming), που σημαίνει ότι υπάρχει άμεση πρόσβαση στον πηγαίο και "τακτοποιημένο" κώδικα και επομένως δεν υπάρχει άμεση εξάρτηση με τον προγραμματιστή - κατασκευαστή μιας εφαρμογής βασισμένη σε αυτό.

### **Τι είναι τα AMPs;**

Τα AMPs (Apache+MySQL+PHP και πολλές φορές και Perl ή και Python) είναι πακέτα λογισμικού που σου προσφέρουν μια γρήγορη προρυθμισμένη εγκατάσταση του διακομιστή Apache, της server-side scripting γλώσσας PHP, του DBMS MySQL και πολλών προεκτάσεων (extensions) τους, τα οποία είναι βασικά συστατικά για την δημιουργία ενός βιώσιμου γενικής χρήσης διακομιστή ιστοσελίδων. Πολλές φορές συμπεριλαμβάνονται σε αυτά τα πακέτα λογισμικού και ένας διακομιστής ηλεκτρονικής αλληλογραφίας (π.χ. Mercury Mail Server) ή κάποιος διακομιστής FTP (π.χ. ProFTPD). Τα περισσότερα AMPs περιλαμβάνουν και κάποια εργαλεία που διευκολύνουν την διαχείριση του συστήματος, παραδείγματος χάριν το εργαλείο phpMyAdmin το οποίο είναι γραμμένο σε PHP και προσφέρει ένα γραφικό περιβάλλον για την διαχείριση του DBMS MySQL.





Εικόνα 6: Η σελίδα phpMyAdmin

Τα διάφορα AMPs ανάλογα με το λειτουργικό σύστημα για το οποίο προορίζονται παίρνουν και ένα πρόθεμα δηλαδή:

- LAMP = Linux AMP.
- MAMP = Macintosh AMP.
- WAMP = MS Windows AMP.
- SAMP = Solaris AMP.
- OAMP = OpenBSD AMP.
- XAMPP = AMP με υποστήριξη πολλαπλών λειτουργικών συστημάτων (X = Cross-Platform = Multi-platform).

Υπάρχουν πάρα πολλά AMPs που το όνομα τους δεν έχει αυτού του είδους το πρόθεμα παραδείγματος χάριν:

- Zend Core = Cross-Platform AMP.
- Bitnami = Linux AMP.
- EasyPHP = MS Windows AMP.
- DAMP = Macintosh AMP.

Γενικά τα AMPs έρχονται σε διάφορες ποικιλίες ανάλογα με τα έξτρα που προσφέρουν και το λειτουργικό σύστημα που υποστηρίζουν. Τα AMPs συνήθως συνιστώνται για τοπική ανάπτυξη διότι είναι ρυθμισμένα για εύκολη και γρήγορη ανάπτυξη εφαρμογών χωρίς να λαμβάνουν υπόψη την ασφάλεια.

## Γιατί XAMPP;

Την στιγμή που γραφόταν αυτή η πτυχιακή εργασία το Joomla! CMS ήταν σε έκδοση 1.5.14 και χρησιμοποιήσαμε το XAMPP έκδοσης 1.7.2 με:

- Apache 2.2.12 (IPv6 enabled)
- MySQL 5.1.37 (Community Server) with PBXT engine 1.0.08-rc
- PHP 5.3.0 + PEAR (PEAR, Mail\_Mime, MDB2, Zend)
- Perl 5.10.0 (Bundle::Apache2, Bundle::Apache::ASP, Bundle::Email, Bundle::DBD::mysql, DBD::SQLite)
- phpMyAdmin 3.2.0.1
- Mercury Mail Transport System v4.62
- FileZilla FTP Server 0.9.32

Διαλέξαμε το XAMPP για την τοπική εγκατάσταση των Apache, MySQL και PHP διότι εκείνη την στιγμή ήταν το μοναδικό AMP το οποίο ικανοποιούσε τις τεχνικές προδιαγραφές της τελευταίας έκδοσης της Joomla! CMS. Παρόλα αυτά δεν είναι αναγκαίο κάποιος να εγκαταστήσει το XAMPP για να μπορέσει να εγκαταστήσει το Joomla! CMS, μπορεί να χρησιμοποιήσει οποιοδήποτε άλλο AMP αρκεί να πληροί τις τεχνικές προδιαγραφές του Joomla! CMS.

# Apache HTTP Server

---

## **Τι είναι ο Apache;**

Ο Apache «A PAtCHy Server» HTTP Server είναι ένας διακομιστής/εξυπηρετητής του πρωτοκόλλου HTTP, είναι ένα δικτυακό λογισμικό διακομιστή το οποίο είναι αξιοσημείωτο για τον ρόλο κλειδί που έπαιξε στην αρχική ανάπτυξη του Παγκόσμιου Ιστού. Το 2009 ήταν ο πρώτος δικτυακός διακομιστής οποίος ξεπέρασε το ορόσημο των 100 εκατομμυρίων ιστοσελίδων. Ο Apache ήταν η πρώτη βιώσιμη εναλλακτική λύση έναντι του διακομιστή Netscape Communications Corporation Web Server (αυτή την στιγμή ονομάζεται Sun Java System Web Server) και έκτοτε έχει εξελιχθεί σε αντίπαλος άλλων δικτυακών εξυπηρετητών βασισμένων σε Unix από την άποψη λειτουργικότητας και επίδοσης. Η πλειοψηφία των δικτυακών διακομιστών που χρησιμοποιούν λογισμικό διακομιστή Apache τρέχουν πάνω σε κάποιο λειτουργικό σύστημα τύπου Unix.

Ο Apache αναπτύσσεται και συντηρείται από μια ανοιχτή κοινότητα προγραμματιστών υπό την αιγίδα του Apache Software Foundation. Η εφαρμογή είναι διαθέσιμη σε ευρεία γκάμα λειτουργικών συστημάτων συμπεριλαμβανομένου του Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF και eComStation. Ο Apache χαρακτηρίζεται ως λογισμικό ανοικτού κώδικα και εκδίδεται υπό την άδεια χρήσης Apache (Apache License). Από τον Απρίλιο του 1996 ο Apache φιλοξένησε πάνω από 54.46% όλων των ιστοσελίδων και πάνω από 66% των πιο απασχολημένων.

## **Τι είναι το Apache Software Foundation.**

Το Ίδρυμα Λογισμικού Apache (Apache software Foundation) παρέχει οργανωτική, νομική και οικονομική υποστήριξη για ένα ευρύ φάσμα project λογισμικού ανοικτού κώδικα. Το ASF παρέχει ένα καθιερωμένο πλαίσιο για πνευματική ιδιοκτησία και οικονομική συνεισφορά που ταυτόχρονα περιορίζει τον συνεισφέρων από ενδεχόμενη νομική έκθεση. Μέσω μιας συνεργατικής και αξιοκρατικής διαδικασίας ανάπτυξης τα Apache projects επιφέρουν εταιρικού επιπέδου και ελεύθερα διαθέσιμα προϊόντα λογισμικού που προσελκύουν μεγάλες κοινότητες χρηστών. Η πρακτική άδεια χρήσης Apache (Apache License) καθιστά εύκολο για όλους τους χρήστες, εμπορικούς και ατομικούς, να αναπτύξουν προϊόντα Apache.

Παλαιότερα ήταν γνωστό ως το γκρουπ Apache (the Apache Group), το Ίδρυμα έχει ενσωματωθεί ως μια εταιρία συνδρομητικής βάσης και μη κερδοσκοπικού χαρακτήρα προκειμένου να διασφαλιστεί ότι τα project Apache εξακολουθούν να υφίστανται πέραν της συμμετοχής μεμονωμένων εθελοντών. Κάθε άτομο που έχει επιδείξει δέσμευση για συλλογική ανάπτυξη λογισμικού ανοικτού κώδικα, μέσω

διαρκούς συμμετοχής και συνεισφοράς στα project του Ιδρύματος, μπορούν να αποκτήσουν ιδιότητα μέλους στο ASF. Κάθε άτομο γίνεται μέλος μετά από υποψηφιότητα και έγκριση από την πλειοψηφία των υπαρχόντων μελών του ASF. Έτσι το Ίδρυμα διοικείται από την κοινότητα την οποία άμεσα εξυπηρετεί – τους ανθρώπους που συνεργάζονται εντός των project του ASF.

Τα μέλη του ASF περιοδικά εκλέγουν ένα Διοικητικό Συμβούλιο για την διαχείριση των οργανωτικών υποθέσεων του Ιδρύματος, σύμφωνα με το Καταστατικό του ASF. Το Συμβούλιο με την σειρά του καθορίζει έναν αριθμό υπάλληλων για την εποπτεία των καθημερινών δραστηριοτήτων του Ιδρύματος. Ένας αριθμός δημοσίων αρχείων των δραστηριοτήτων τίθενται στη διάθεση της κοινότητας.

### **Τι είναι το Apache HTTP server Project;**

Το Apache HTTP server Project είναι μία συνεργατική προσπάθεια ανάπτυξης λογισμικού στοχεύοντας την δημιουργία ενός εύρωστου, εμπορικού επιπέδου, με πλήρης χαρακτηριστικά και ελεύθερα διαθέσιμου πηγαίου κώδικα υλοποίησης ενός διακομιστή HTTP. Το project διαχειρίζεται κοινά από μία ομάδα εθελοντών από όλων τον κόσμο, χρησιμοποιώντας το Διαδίκτυο και τον Παγκόσμιο Ιστό για να επικοινωνήσουν, σχεδιάσουν και αναπτύξουν τον διακομιστή και τα σχετικά έγγραφα τεκμηρίωσης. Αυτό το project είναι μέρος του Ιδρύματος Λογισμικού Apache. Επίσης εκατοντάδες χρήστες έχουν συνεισφέρει ιδέες, κώδικα και έγγραφα τεκμηρίωσης.

### **Πώς έφτασε ο Apache HTTP server ως εδώ;**

Τον Φεβρουάριο του 1995, το πιο διάσημο λογισμικό διακομιστή στον Παγκόσμιο Ιστό ήταν ο public domain HTTP daemon που αναπτυσσόταν από τον Rob McCool στο Εθνικό Κέντρο για Υπερυπολογιστικές Εφαρμογές του Πανεπιστημίου του Illinois, Urbana-Champaign (National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign). Ωστόσο η ανάπτυξη του httpd πάγωσε αφού έφυγε ο Rob από το NCSA στα μέσα του 1994 και πολύ webmasters είχαν αναπτύξει τις δικές τους προεκτάσεις και διορθώσεις του κώδικα τα οποία χρειάζονταν μια κοινής διανομής. Μια μικρή ομάδα αυτών των webmasters επικοινωνούσε μέσω προσωπικού ηλεκτρονικού ταχυδρομείου για τον σκοπό να συντονιστούν οι αλλαγές τους στον κώδικα (σε μορφή patch εξ' ου και το όνομα APACHE). Ο Brian Behlendorf και ο Cliff Skolnick δημιούργησαν μια ηλεκτρονική λίστα αλληλογραφίας, μοίρασαν χώρο και πρόσβαση για τους βασικούς προγραμματιστές σε ένα μηχάνημα στην περιοχή California Bay, μαζί με εύρος ζώνης δωρισμένο από την HotWired. Μέχρι το τέλος του Φεβρουαρίου, 8 βασικοί συνεισφέροντες σχημάτισαν το ίδρυμα του πρωτότυπου Apache Group. Τα ονόματα αυτών Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick,

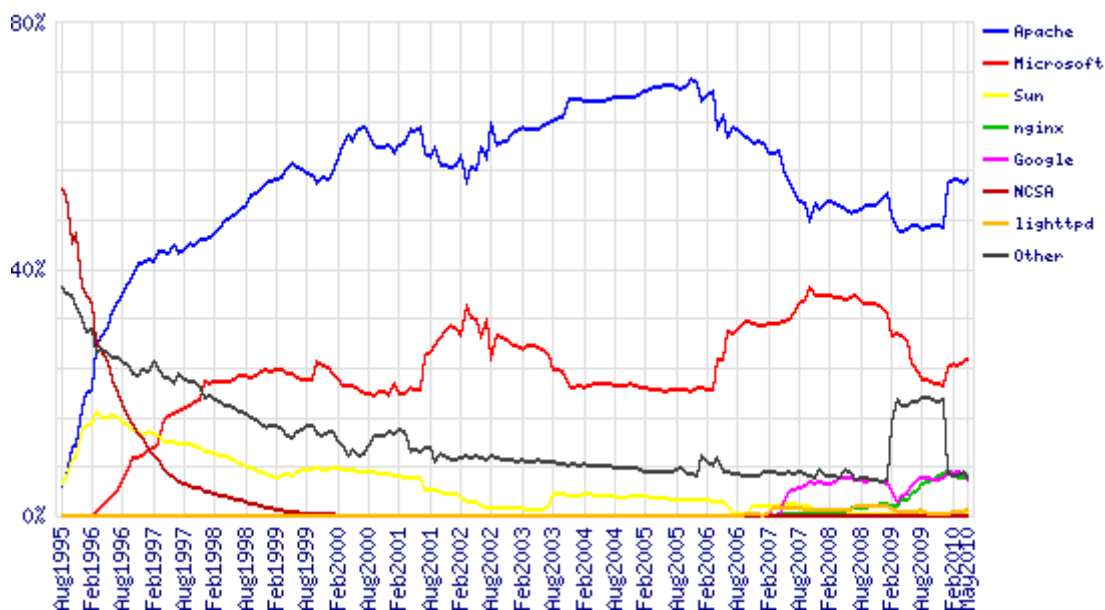
Randy Terbush, Rober S. Thau, Andrew Wilson και πρόσθετες συνεισφορές από: Eric Hagberg, Frank Peters, Nicolas Pioch.

Χρησιμοποιώντας το NCSA httpd 1.3 σαν βάση τον Απρίλιο του 1995 δημιούργησαν την πρώτη επίσημη δημόσια κυκλοφορία του διακομιστή Apache με αριθμό έκδοσης 0.6.2. Εντελώς συμπτωματικά την ίδια εποχή το NCSA ξεκίνησαν ξανά την ανάπτυξη του httpd, ο Brandon Long και ο Beth Frank της ομάδας NCSA Server Development (ανάπτυξης εξυπηρετητών) έγιναν μέλη της ηλεκτρονικής λίστας αλληλογραφίας ως επίτιμα μέλη ώστε τα δύο project να μπορούν να μοιράζονται ιδέες και διορθώσεις του κώδικα.

Ο πρώιμος διακομιστής Apache είχε μεγάλη επιτυχία, αλλά όλοι ξέρανε ότι το codebase (η βάση του κώδικα) χρειαζόταν μια γενική αναμόρφωση και επανασχεδίαση. Κατά την περίοδο Μαΐου-Ιουνίου το 1995, ενώ ο Rob Hartill και η υπόλοιπη ομάδα επικεντρώθηκαν στην εφαρμογή νέων χαρακτηριστικών για την έκδοση 0.7.x (όπως pre-forked child processes) και υποστήριξη της ταχέως αναπτυσσόμενης κοινότητας χρηστών του Apache, ο Robert Thau σχεδίασε μια νέα αρχιτεκτονική εξυπηρετητή με κωδική ονομασία Shambhala ή οποία περιείχε «αρθρωτή» (modular) δομή και API (Application programming interface) για καλύτερη επεκτασιμότητα, κατανομή μνήμης τύπου pool και προσαρμοστικό μοντέλο διεργασίας pre-forking. Η ομάδα μεταφέρθηκε σε αυτήν την βάση διακομιστή τον Ιούλιο και πρόσθεσαν τα χαρακτηριστικά από την έκδοση 0.7.x, με αποτέλεσμα τον Apache 0.8.8 τον Αύγουστο.

Μετά από εκτενής δοκιμές των beta εκδόσεων, ανάπτυξη για υποστήριξη αφανών πλατφορμών, ένα νέο σετ εγγράφων τεκμηρίωσης (από τον David Robinson) και πολλά νέα χαρακτηριστικά στην μορφή τυποποιημένων επεκτάσεων (λόγο της αρθρωτής δομής), ο Apache 1.0 εκδόθηκε την 1η Δεκεμβρίου 1995.

Σε λιγότερο από ένα χρόνο από την δημιουργία της ομάδας (Apache group), ο Apache ξεπέρασε τον httpd της NCSA ως ο νούμερο ένα διακομιστής στο Διαδίκτυο και σύμφωνα με την έρευνα της Netcraft, διατηρεί αυτήν την θέση μέχρι και σήμερα.



Εικόνα 7: Μάιος 2010. Διαδικτυακή έρευνα από την Netcraft

	Απρ. 2010	Ποσοστό	Μάιος 2010	Ποσοστό	Μεταβολή
<b>Apache</b>	110,752,854	53.93%	112,663,533	54.68%	0.75
<b>Microsoft</b>	51,284,570	24.97%	52,062,154	25.27%	0.30
<b>nginx</b>	12,977,486	6.32%	13,490,726	6.55%	0.23
<b>Google</b>	13,749,829	6.70%	12,357,212	6.00%	-0.70
<b>lighttpd</b>	1,078,403	0.53%	1,869,658	0.91%	0.38

## Ανάπτυξη του Apache.

Η ανάπτυξη του Apache γίνεται από ένα κεντρικό γκρουπ συντελεστών, που αρχικά σχηματιζόταν από τους ιδρυτές του Apache project, ανά καιρούς ενσωματώνονται και άλλοι εξαιρετικοί συντελεστές. Υπάρχουν «committers», στους οποίους έχει επιτραπεί η πρόσβαση στα αποθετήρια διαχείρισης πηγαίου κώδικα για να βοηθούν στην συντήρηση του project ή των εγγράφων και το κεντρικό γκρουπ συντελεστών το οποίο διαχειρίζεται το project, οι οποίοι ονομάζονται Επιτροπή διαχείρισης του Apache HTTP Project (Apache HTTP Project Management Committee -PMC). Γενικά κάθε project του Apache Software Foundation έχει την δική της επιτροπή διαχείρισης, για να καθορίζουν τους «committers», την πορεία του project και την συνολική διαχείριση. Οι όροι «The Apache Group» ή «Apache Core» δεν χρησιμοποιούνται πλέον.

Το Apache project είναι αξιοκρατικό – όσο πιο πολύ έργο έχεις κάνει τόσο πιο πολλά θα σου επιτρέπεται να κάνεις. Οι ιδρυτές του γκρουπ όρισαν τους αρχικούς κανόνες, αλλά μπορούν να αλλάξουν με ψήφο από τα ενεργά μέλη της επιτροπής PMC. Υπάρχει ένα γκρουπ ανθρώπων οι οποίοι έχουν πρόσβαση στον διακομιστή και στα αποθετήρια πηγαίου κώδικα. Όλοι έχουν πρόσβαση μόνο για ανάγνωση στα

αποθετήρια. Αλλαγές στον πηγαίο κώδικα προτείνονται στις λίστες ηλεκτρονικής αλληλογραφίας και συνήθως ψηφίζονται από τα ενεργά μέλη – τρεις θετικές ψήφοι ή αρνητικές ψήφοι χρειάζονται για να γίνει μία αλλαγή στον πηγαίο κώδικα κατά την περίοδο ανάπτυξης και έκδοσης, οι αλλαγές γίνονται συνήθως πρώτα και μετά αλλάζουν αν χρειαστεί, με τις διαφωνίες να λύνονται από την πλειοψηφία των ψήφων.

Η κύρια μέθοδος επικοινωνίας είναι η λίστα ηλεκτρονικής αλληλογραφίας. Περίπου 40 μηνύματα ρέουν στην λίστα καθημερινά, χαρακτηριστικά έχουν τόνο συνομιλίας. Συζητούνται νέα χαρακτηριστικά για να προστεθούν, διορθώσεις σφαλμάτων, προβλήματα χρηστών, εξελίξεις στην κοινότητα των εξυπερευνητών δικτύου, ημερομηνίες κυκλοφορίες, κ.τ.λ.. Η πραγματική ανάπτυξη κώδικα γίνεται στους τοπικούς υπολογιστές των προγραμματιστών, με προτεινόμενες αλλαγές να ανακοινώνονται χρησιμοποιώντας patch (το αποτέλεσμα μιας ενοποιημένης «diff – u παλαιό\_αρχείο νέο\_αρχείο» εντολής) και μετά το περνάνε στο αποθετήριο πηγαίου κώδικα ένας από τους συνεισφέροντες (committers). Οποιοσδήποτε που παρακολουθεί την λίστα ηλεκτρονικής αλληλογραφίας μπορεί να ψηφίσει ένα συγκεκριμένο θέμα, αλλά μόνο οι ψήφοι από τα ενεργά μέλη ή από ανθρώπους γνώστες του συγκεκριμένου θέματος προσμετρούνται ώστε να ικανοποιηθεί η κατάλληλη συνθήκη (να γίνει ή όχι). Οι αρνητικές ψήφοι πρέπει να συνοδεύονται από μια πειστική τεχνική αιτιολόγηση.

Τα νέα μέλη της επιτροπής διαχείρισης του Apache HTTP Project (Apache HTTP PMC) προστίθενται όταν ένας συντελεστής προτείνεται από ένα μέλος και ομόφωνα εγκρίνεται από τα μέλη με δικαίωμα ψήφου. Στις περισσότερες περιπτώσεις αυτά τα νέα μέλη έχουν ενεργά συνεισφέρει στο έργο του γκρουπ για περισσότερο από έξη μήνες, έτσι η επιλογή είναι συνήθως εύκολη.

Οι κατευθυντήριες γραμμές του project συνεχώς εξελίσσονται υπό την εποπτεία της PMC, καθώς τα μέλη του γκρουπ αλλάζουν και τα αναπτυξιακά/συντονιστικά εργαλεία βελτιώνονται.

## **Γιατί ο Apache είναι δωρεάν ;**

Το Ίδρυμα Λογισμικού Apache (ASF) αναφέρει στη ιστοσελίδα της (σε ελεύθερη μετάφραση στα Ελληνικά):

«Λογισμικό Apache υφίσταται για να παρέχει μία εύρωστη και εταιρικού επιπέδου εφαρμογή αναφοράς πολλών τύπων λογισμικού. Θα πρέπει να παραμείνει μια πλατφόρμα πάνω στην οποία τα άτομα και τα ιδρύματα μπορούν να δημιουργήσουν αξιόπιστα συστήματα, τόσο για πειραματικούς σκοπούς όσο και για κρίσιμους για κάποια αποστολή σκοπούς. Πιστεύουμε ότι τα εργαλεία online publishing θα πρέπει να είναι στα χέρια του καθενός και οι εταιρίες λογισμικού θα

πρέπει να βγάζουν τα χρήματά τους με την παροχή υπηρεσιών προστιθέμενης αξίας όπως ενθέματα (module) και υποστήριξη, μεταξύ άλλων. Αντιλαμβανόμαστε ότι συχνά θεωρείται ως οικονομικό πλεονέκτημα για μία εταιρία να της «ανήκει» μία αγορά – στην βιομηχανία λογισμικού αυτό σημαίνει να ελέγχει σφικτά ένα συγκεκριμένο αγωγό έτσι ώστε όλοι οι άλλοι θα πρέπει να πληρώσουν για να των χρησιμοποιήσουν. Αυτό γίνεται συνήθως με την «ιδιοκτησία» του πρωτοκόλλου μέσω του οποίου οι εταιρείες επιχειρούν, σε βάρος όλων αυτών των άλλων εταιρειών. Στο βαθμό που τα πρωτόκολλα του Παγκόσμιου Ιστού παραμένουν «μη-ιδιόκτητα» από μία μεμονωμένη εταιρία, το Διαδίκτυο θα παραμείνει ισότιμος όρος ανταγωνισμού για τις μεγάλες και μικρές εταιρείες. Έτσι, «ιδιοκτησία» των πρωτοκόλλων πρέπει να εμποδιστεί. Για το σκοπό αυτό, είναι πάρα πολύ καλό η ύπαρξη ισχυρών εφαρμογών αναφοράς των διαφόρων πρωτοκόλλων και διεπαφών προγραμματισμού εφαρμογών (Application Programming Interfaces-API), να διατίθεται δωρεάν σε όλες τις εταιρείες και τους ιδιώτες.

Επιπλέον, το Ίδρυμα Λογισμού Apache (ASF) είναι ένας οργανικός φορέας: εκείνοι που επωφελούνται από αυτό το λογισμικό, χρησιμοποιώντας το, συχνά συνεισφέρουν πίσω σε αυτό παρέχουν βελτιώσεις χαρακτηριστικών, διορθώσεις σφαλμάτων καθώς και υποστήριξη για τους άλλους σε δημόσιους καταλόγους και ομάδες συζητήσεων. Η προσπάθεια που καταβάλλεται από οποιοδήποτε μεμονωμένο άτομο είναι συνήθως αρκετά ελαφριά, αλλά το απορρέον προϊόν γίνεται πολύ ισχυρό. Αυτού του είδους κοινότητες μπορούν να δημιουργηθούν μόνο με ελεύθερα διαθέσιμο λογισμικό – όταν κάποιος έχει πληρώσει για το λογισμικό, συνήθως δεν είναι πρόθυμος να διορθώσει τα σφάλματα δωρεάν. Μπορεί κανείς να υποστηρίξει, τότε, ότι η δύναμη του Apache προέρχεται από το γεγονός ότι είναι δωρεάν και ότι αν γινόταν «μη-δωρεάν» θα υπέφερε τρομερά, ακόμη και αν αυτά τα χρήματα δαπανηθούν σε μία πραγματική ομάδα ανάπτυξης.

Θέλουμε να δούμε το λογισμικό Apache να χρησιμοποιείται ποικίλα ευρέως – από μεγάλες εταιρίες, μικρές εταιρίες, ερευνητικά ινστιτούτα, σχολεία, ιδιώτες, στο περιβάλλον intranet, παντού –έστω και αν αυτό σημαίνει ότι εταιρίες έχουν την δυνατότητα να έχουν εμπορικά λογισμικά και θα πλήρωναν για αυτά χωρίς δισταγμό, θα μπορούν να έχουν ένα «δωρεάν βόλτα» χρησιμοποιώντας Apache. Είμαστε ευτυχής ακόμα όταν μερικές εμπορικές εταιρίες λογισμικού εγκαταλείπουν εντελώς τα δικά τους σχέδια ανάπτυξης ενός διακομιστή HTTP και χρησιμοποιούν τον Apache ως βάση, με ορθή απόδοση όπως περιγράφετε στην Άδεια Χρήσης (LICENSE). Εν κατακλείδι ο Apache HTTP Server προέρχεται μόνο από το Ίδρυμα Λογισμικού Apache (ASF) αλλά πολλοί πωλητές προμηθεύουν το δικό τους προϊόν βασισμένο στο Apache {project}. Δεν υπάρχει {πωλητής}Apache{Προϊόν}, αυτό είναι μια κατάχρηση των θεμελίων του Ιδρύματος Λογισμικού Apache.»



## Πως λειτουργεί ο Apache httpd;

Ο Apache είναι ένα πρόγραμμα το οποίο τρέχει υπό κατάλληλο λειτουργικό σύστημα που υποστηρίζει multitasking. Λειτουργικά συστήματα όπως Unix και Windows 95/98/2000/Me/NT/ κτλ. τα οποία για χάρη συντομίας τα ονομάζουμε Win32, επίσης διάφορες παραλλαγές των Unix όπως της IBM το OS/2 και της Novell το Netware καθώς και της Macintosh το OS X που έχει βάση το FreeBSD και έρχεται μαζί με τον Apache.

Το δυαδικό αρχείο του Apache ονομάζεται httpd σε λειτουργικό σύστημα Unix και apache.exe υπό λειτουργικό σύστημα Win32 και συνήθως τρέχει στο παρασκήνιο. Κάθε αντίγραφο του httpd/apache το οποίο ξεκινά έχει την προσοχή του στραμμένη σε έναν δικτυακό τόπο, το οποίο συνήθως είναι ένας κατάλογος αρχείων και φακέλων. Ανεξαρτήτου του λειτουργικού συστήματος ο κατάλογος αρχείων και φακέλων ενός δικτυακού τόπου τυπικά εμπεριέχει τέσσερις υποκαταλόγους.

- **conf**  
Ο κατάλογος conf εμπεριέχει το αρχείο/-ία ρυθμίσεων, των οποίων το πιο σημαντικό είναι το httpd.conf. Πολλές φορές αναφέρετε ως το αρχείο Config. Σε αυτό το αρχείο καθορίζονται τα URL τα οποία θα διακομιστούν.
- **htdocs**  
Ο κατάλογος htdocs εμπεριέχει τα αρχεία HTML τα οποία θα διακομιστούν του πελάτες του δικτυακού τόπου. Αυτός ο κατάλογος και ότι εμπεριέχει ονομάζεται συνήθως Web Space (δικτυακός χώρος) και είναι προσπελάσιμος από οποιοδήποτε στο Διαδίκτυο και ως εκ τούτου παρουσιάζετε σοβαρός κίνδυνος για την ασφάλεια, εάν χρησιμοποιηθεί για οτιδήποτε άλλο εκτός από δημόσια δεδομένα.
- **logs**  
Στον κατάλογο logs αποθηκεύονται τα δεδομένα καταγραφής των προσβάσεων και των σφαλμάτων.
- **cgi-bin**  
Στον κατάλογο cgi-bin εμπεριέχονται τα script CGI. Αυτά είναι προγράμματα ή shell scripts γραμμένα από ή για τον webmaster τα οποία μπορούν να εκτελεστούν από τον Apache για λογαριασμό των πελατών του. Είναι πάρα πολύ σημαντικό για λόγους ασφαλείας αυτός ο φάκελος να μην βρίσκεται στο Web Space (δικτυακό χώρο) δηλαδή μέσα στον κατάλογο /htdocs ή και παρακάτω.

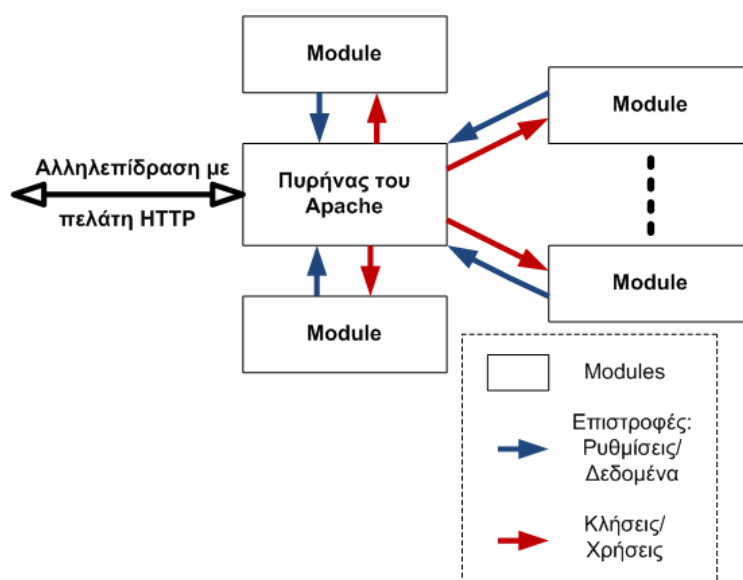
Σε κατάσταση αναμονής ο Apache δεν κάνει τίποτα άλλο εκτός από το να ακούει στις διευθύνσεις IP που έχουν ορισθεί μέσα στο αρχείο Config (httpd.conf). Όταν μία αίτηση παρουσιαστεί ο Apache την λαμβάνει και αναλύει τις κεφαλίδες (headers). Μετά εφαρμόζει τους κανόνες που βρίσκει στο αρχείο Config και λαμβάνει τα κατάλληλα μέτρα. Ο κύριος έλεγχος του Apache από τον Webmaster

γίνεται μέσω του αρχείου Config. Ο Webmaster έχει περίπου 200 οδηγίες στην διάθεση του. Επίσης ο Webmaster έχει μια δωδεκάδα flags τα οποία μπορεί χρησιμοποιήσει όταν ξεκινά ο Apache.

## Αρχιτεκτονική του διακομιστή Ιστοσελίδων Apache

Αντίθετα με την μονολιθική (monolithic) αρχιτεκτονική διακομιστή στην οποία όλες οι δραστηριότητες γίνονται από μια μονάδα, ο Apache χρησιμοποιεί «αρθρωτή» (modular) προσέγγιση.

- **Πυρήνας (Core)**  
Ο πυρήνας είναι υπεύθυνος για τον ορισμό και εκτέλεση των βημάτων για την παροχή ενός αιτήματος.
- **Ενθέματα (Module)**  
Τα module εφαρμόζουν τις διαφορετικές φάσεις διεκπεραίωσης του αιτήματος.



Εικόνα 8: Υψηλού επιπέδου "εννοιολογική" αρχιτεκτονική

Η ιδέα είναι να κρατήσουν τον κώδικα του βασικού διακομιστή καθαρό ενώ επιτρέποντας σε τρίτους να παρακάμπτουν ή να επεκτείνουν ακόμη και βασικά χαρακτηριστικά.

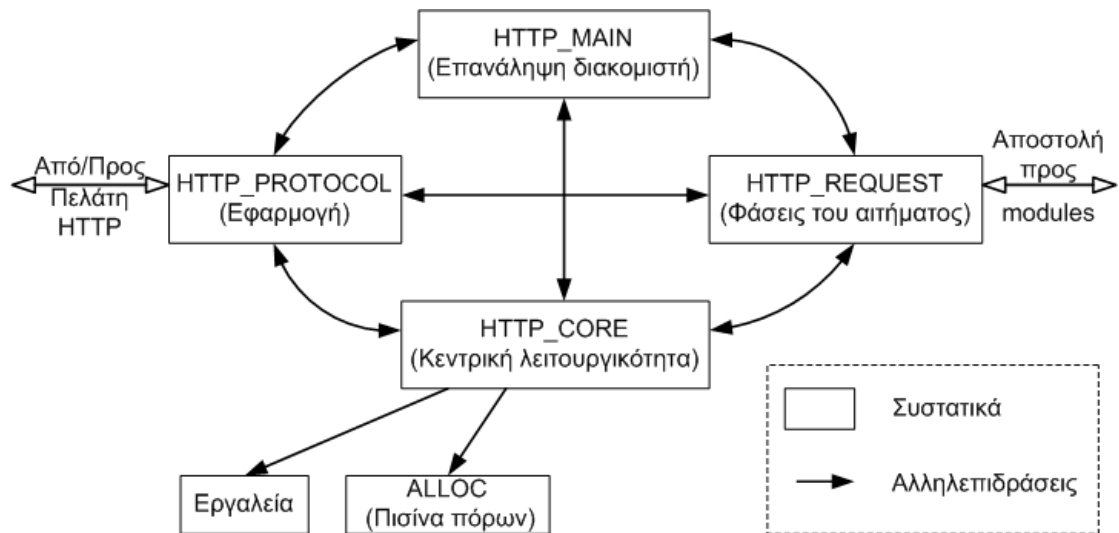
### Ο πυρήνας

Ο πυρήνας (core) υλοποιεί την βασική λειτουργικότητα του διακομιστή. Επιπρόσθετα υλοποιεί ορισμένες βοηθητικές λειτουργίες, π.χ. ένα εργαλείο το οποίο παρέχει διαμερισμό πόρων σε μία ανά αίτηση πύσινα. Αυτή η ευκολία παρέχεται όχι μόνο στον πυρήνα του Apache αλλά και στα ενθέματα (module) του.

## Συστατικά του πυρήνα (core):

- **http\_protocol.c:** Εμπεριέχει ρουτίνες η οποίες επικοινωνούν απευθείας με τον πελάτη (client) (μέσω του socket σύνδεσης), ακλουθώντας το πρωτόκολλο HTTP. Όλες οι μεταφορές δεδομένων χρησιμοποιούν αυτό το συστατικό.
- **http\_main.c:** Αυτό το συστατικό ξεκινάει τον διακομιστή και εμπεριέχει την κεντρική επανάληψη του διακομιστή που περιμένει για και αποδέχεται τις συνδέσεις. Επίσης είναι υπεύθυνο για την διαχείριση των χρονικών ορίων σύνδεσης.
- **http\_request.c:** Το συστατικό αυτό χειρίζεται την ροή τις επεξεργασίας των αιτήσεων, αποστέλλει τον έλεγχο στα module στην κατάλληλη σειρά. Είναι επίσης υπεύθυνο με την διαχείριση των σφαλμάτων.
- **http\_core.c:** Αυτό το συστατικό εφαρμόζει την πιο βασική λειτουργικότητα, η οποία περιγράφεται σε ένα σχόλιο από ένα αρχείο πηγαίου κώδικα ως «μετά βίας αρκετά λειτουργικό να διακομίσει έγγραφα, αν και όχι τόσο καλά». Ένα άλλο ενδιαφέρον σχόλιο από ένα αρχείο πηγαίου κώδικα απεικονίζει πάρα πολύ καλά την λειτουργικότητα αυτού του συστατικού «αυτό το αρχείο θα μπορούσε σχεδόν να είναι mod\_core.c». Το οποίο σημαίνει ότι αυτό το συστατικό συμπεριφέρεται σαν ένα module άλλα πρέπει να έχει πρόσβαση σε μερικές παγκόσμιες (globals) απευθείας (το οποίο δεν είναι χαρακτηριστικό για ένα module).
- **alloc.c:** Το συστατικό αυτό φροντίζει την διαμοίραση των πόρων από πισίνες και την παρακολούθηση τους.
- **http\_config.c:** Αυτό το συστατικό εκτελεί διάφορες άλλες λειτουργίες, συμπεριλαμβανομένου το διάβασμα αρχείων διαμόρφωσης και διαχείριση των πληροφοριών που μάζεψε από αυτά τα αρχεία, όπως και υποστήριξη εικονικών hosts. Μια σημαντική λειτουργία του http\_config.c είναι η διαμόρφωση της λίστας των module τα οποία θα κλιθούν να υπηρετήσουν διαφορετικές φάσεις των αιτήσεων.

*(Υποσημείωση: τα συστατικά του πυρήνα δεν είναι module)*



Εικόνα 9: Ο πυρήνας του Apache.

Η εικόνα παραπάνω απεικονίζει την αλληλεπίδραση μεταξύ διαφορετικών συστατικών του πυρήνα. Επίσης όλα τα συστατικά χρησιμοποιούν τα διαφορετικά εργαλεία, η σύνδεση στα Εργαλεία και ALLOC δεν έχει απεικονιστεί σαφώς. Ο όρος αλληλεπίδραση χρησιμοποιείται ως μια ευρύτερη έννοια, δηλαδή από την κλίση μίας λειτουργίας που είναι υπηρεσία ενός συστατικού μέχρι «εννοιολογικά» παραίτηση του ελέγχου του εν λόγω συστατικού.

### Οι φάσεις ενός αιτήματος

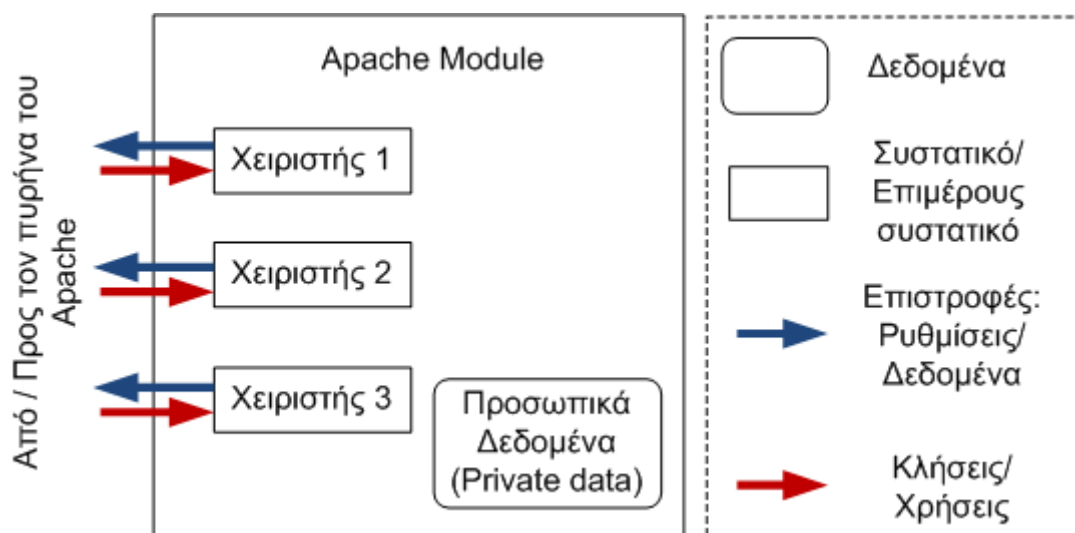
Τα module δεν γνωρίζονται μεταξύ τους. Ο έλεγχος μεταφέρετε από και προς τον πυρήνα και μεταξύ διάφορων module ως εξής:

1. Μετάφραση URI σε όνομα αρχείου.
2. Έλεγχος πρόσβασης βασισμένος στην διεύθυνση του host και άλλες διαθέσιμες πληροφορίες.
3. Παίρνουν ένα κωδικό χρήστη από την αίτηση HTTP και την επικυρώνει.
4. Εξουσιοδοτεί τον χρήστη.
5. Καθορίζει τον τύπο MIME του αντικειμένου της αίτησης (τον τύπο του περιεχομένου, την κωδικοποίηση και την γλώσσα).
6. Διορθώσεις (για παράδειγμα αντικατάσταση των aliases (ψευδώνυμα) με την πραγματική διαδρομή (διεύθυνση)).
7. Αποστολή των πραγματικών δεδομένων πίσω στον client.
8. Καταγραφή των αιτήσεων.

Οι φάσεις «ελέγχονται» από το συστατικό http\_request του πυρήνα (βλ. εικόνα 9).

## Τα ενθέματα (module)

Όλα τα module έχουν την ίδια διεπαφή για τον πυρήνα του διακομιστή. Τα module δεν αλληλεπιδρούν κατευθείαν το ένα με το άλλο. Αλληλεπιδρούν πάντα μέσω του πυρήνα του Apache.



Εικόνα 10: Η αρχιτεκτονική ενός module του Apache

Ο Apache επιτρέπει την φόρτωση των module όταν αυτά χρειάζονται (είναι συνδεδεμένα δυναμικά με τον διακομιστή) και συνεπώς οι μέθοδοι εκκίνησης και ρύθμισης μπορεί να κλιθούν όταν φορτώνετε το module σε αντίθεση με όταν γίνεται η εκκίνηση του διακομιστή.

## Χειριστές (Handlers)

Οι χειριστές χρησιμοποιούνται για μια δράση του Apache που πρέπει να γίνει σε κάποια φάση της εξυπηρέτησης μιας αίτησης. Για παράδειγμα όταν το αντικείμενο της αίτησης είναι ένα αρχείο τότε ο χειριστής που επιστρέφει το αρχείο θα πρέπει να ανοίξει το αρχείο, να διαβάσει το περιεχόμενό του και να το παραδώσει στον client (μέσω του πυρήνα του Apache). Οι χειριστές ορίζονται από τα module, ένα module μπορεί να προσδιορίσει χειριστές για μια, πολλές ή και καμία φάση μιας αίτησης.

## Τυπικά module (standard)

Ο Apache έρχεται με ένα σετ από τυπικά module για να παρέχει πλήρη λειτουργικότητα ενός διακομιστή. Τα πιο αντιπροσωπευτικά/σχετικά ανάμεσα στα standard module είναι τα εξής:

- Για την φάση μετάφρασης του URI σε όνομα αρχείου:
  - **mod\_userdir:** Μεταφράζει τους καταλόγους home του χρήστη σε πραγματικές διαδρομές (διευθύνσεις).
  - **mod\_rewrite:** Ξαναγράφει τα URLs βασισμένα σε κανονικές εκφράσεις (regular expressions), έχει επιπρόσθετους χειριστές για διορθώσεις και για τον καθορισμό του τύπου MIME.
- Για την φάση πιστοποίησης / εξουσιοδότησης:
  - **mod\_auth, mod\_auto\_anon, mod\_auth\_db, mod\_auth\_dbm:** Πιστοποιεί τον χρήστη χρησιμοποιώντας αρχεία κειμένων, ανώνυμους σε συλ FTP, αρχεία Berkeley DB και αρχεία DBM.
  - **mod\_access:** Διαχειρίζεται τις προσβάσεις βασισμένες στον host.
- Για την φάση καθορισμού του τύπου MIME του αιτηθέντος αντικειμένου (τον τύπο του περιεχομένου, την κωδικοποίηση και την γλώσσα):
  - **mod\_mime:** Καθορίζει τον τύπο του αρχείου βάση των επεκτάσεων τους.
  - **mod\_mime\_magic:** Καθορίζει τον τύπο του αρχείου χρησιμοποιώντας τους «μαγικούς αριθμούς» (π.χ. όλα τα αρχεία gif ξεκινάνε με έναν συγκεκριμένο κωδικό).
- Για την φάση διορθώσεων:
  - **mod\_alias:** Αντικαθιστά τα ψευδώνυμα (aliases) με τις πραγματικές διαδρομές.
  - **mod\_env:** Διορθώνει το περιβάλλον (βάση στις πληροφορίες που βρίσκονται στα αρχεία ρυθμίσεων).
  - **mod\_speling:** Αυτόματα διορθώνει ελάσσων τυπογραφικά λάθη στα URL.
- Για την φάση αποστολής πραγματικών δεδομένων πίσω στον client:
 

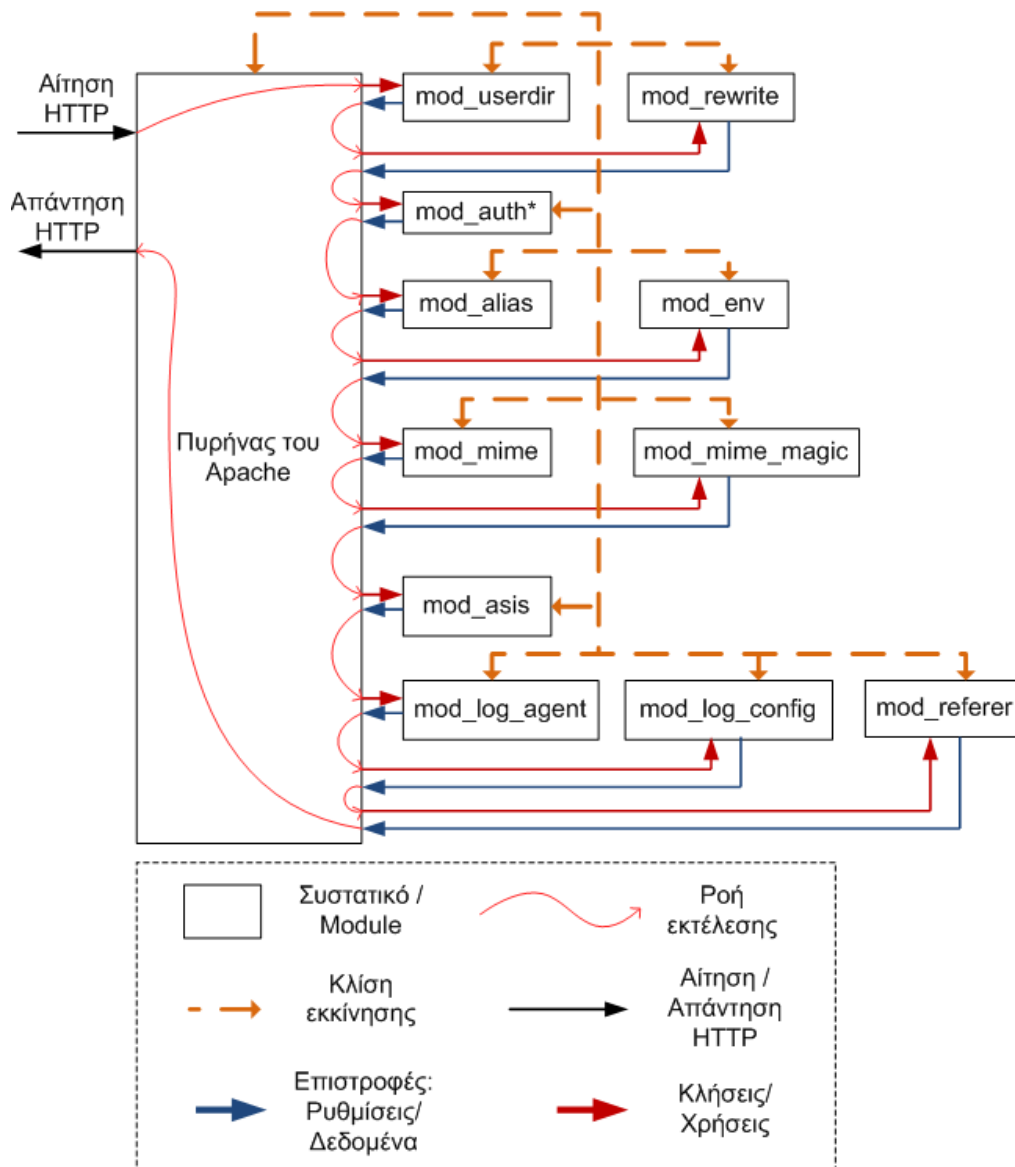
Για την επιλογή του κατάλληλου module σε αυτήν την φάση ο τύπος MIME ή ο τύπος ψευδό-MIME (π.χ. για ένα script CGI) χρησιμοποιείται.

  - **mod\_actions:** Εκτελεί script βασισμένα σε μεθόδους/τύπους αρχείου.
  - **mod\_asis:** Στέλνει το αρχείο όπως είναι.
  - **mod\_autoindex:** Στέλνει μια αυτόματα παραγόμενη αναπαράσταση μιας λίστας καταλόγων.
  - **mod\_cgi:** Επικαλείται script CGI και επιστρέφει το αποτέλεσμα τους.
  - **mod\_include:** Χειρίζεται τα server side includes (SSI) (ανάλυση εγγράφων από τον διακομιστή τα οποία εμπεριέχουν ορισμένα επιπρόσθετα δεδομένα πριν δοθεί το έγγραφο στον client).
  - **mod\_dir:** Χειρίζεται σε βασικό επίπεδο καταλόγους.
  - **mod\_imap:** Χειρίζεται αρχεία image-map.
- Για την φάση καταγραφής (logging) της αίτησης:
  - **mod\_log\_\*:** Module διαφόρων τύπων για καταγραφή.

## Εννοιολογική αρχιτεκτονική

Η «εννοιολογική» αρχιτεκτονική του διακομιστή Apache έχει «αρθρωτή» (modular) δομή ώστε να επιτρέπει στους χρήστες να προσθέτουν/τροποποιούν την λειτουργικότητα με ευκολία. Η επεκτάσεις του Apache μπορούν να υλοποιηθούν σε C ή σε ένα ευρύ φάσμα άλλων γλωσσών προγραμματισμού με την χρήση κατάλληλων module.

Μερικές φάσεις όπως η εξουσιοδότηση, η πιστοποίηση, η επιστροφή αντικειμένων στους client επιτρέπει μόνο ένα module (χειριστή ενός module) να κλιθεί. Άλλες φάσεις εξυπηρέτησης μίας αίτησης μπορεί να έχουν καλέσει παραπάνω από ένα χειριστή (π.χ. μπορούμε να έχουμε παραπάνω από ένα module το οποίο καλείται να υλοποιήσει το κομμάτι της καταγραφής της αίτησης). Επίσης άλλες φάσεις μπορεί να απαιτήσουν όλους τους χειριστές (εντός των δηλωμένων/ενεργοποιημένα module) να κλιθούν μέχρι ένα από αυτά να επιστρέψει έναν ειδικό κωδικό.

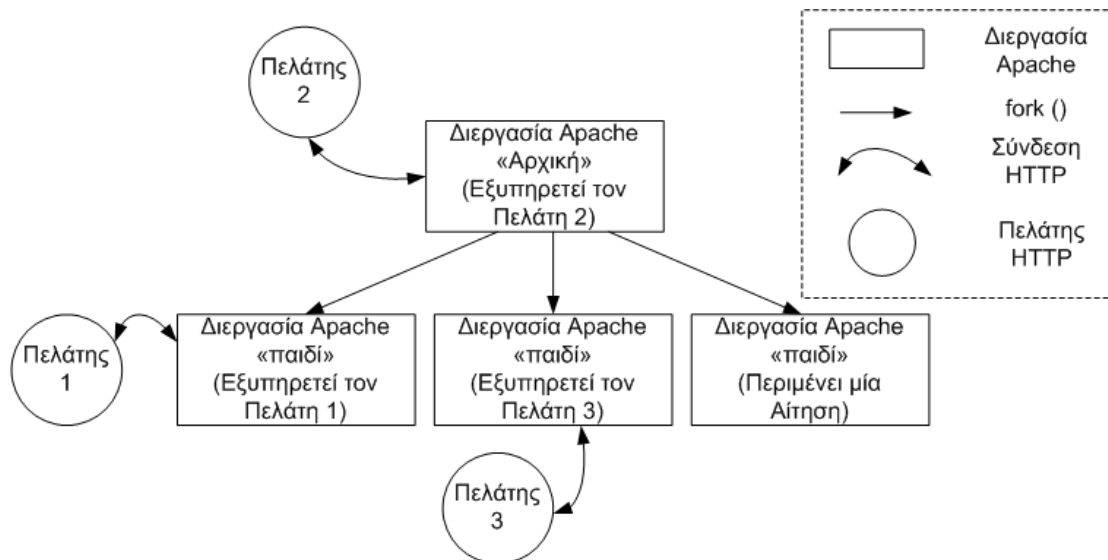


Εικόνα 11: Εννοιολογική Αρχιτεκτονική του Apache

### Συγχρονισμός στον Apache.

Ορισμένοι ιστότοποι έχουν μεγάλο φόρτο (πολλές αιτήσεις ανά λεπτό ή ακόμα ανά δευτερόλεπτο). Παραδοσιακά οι εξυπηρετητές TCP/IP αναπαράγουν (fork) ένα νέο «παιδί» για να διαχειριστεί νέες εισερχόμενες αιτήσεις από τους πελάτες. Ωστόσο, στην περίπτωση ενός πολυάσχολου ιστότοπου η παραπάνω αναπαραγωγή ενός μεγάλου αριθμού «παιδιών» απλά θα πνίξει το μηχάνημα. Ως συνέπεια, ο Apache χρησιμοποιεί μια διαφορετική τεχνική που ονομάζεται persistent server processes (επίμονες διαδικασίες διακομιστή). Αναπαράγει ένα σταθερό αριθμό «παιδιών» από την αρχή. Τα «παιδιά» εξυπηρετούν εισερχόμενες αιτήσεις ανεξάρτητα (διαφορετικοί χώροι διευθύνσεων). Ο συγχρονισμός του διακομιστή Apache φαίνεται στην εικόνα 12.





Εικόνα 12: Συγχρονισμός στον Apache (Unix)

Εναλλακτικά, όταν ο Apache μεταγλωττίζεται σε λειτουργικό σύστημα Microsoft Windows (σε αντίθεση με UNIX), ξεκινάει από την αρχή με ένα σταθερό αριθμό διεργασιών (threads) για την εξυπηρέτηση των εισερχομένων αιτήσεων (πιθανών λόγο ιδιαιτεροτήτων του λειτουργικού συστήματος).

Είναι ενδιαφέρον που ο διακομιστής Apache μπορεί δυναμικά να ελέγχει τον αριθμό των «παιδιών» που αναπαράγει (αυξάνοντας ή μειώνοντάς τα), με βάση τον τρέχον φόρτο.

Από μια άλλη οπτική γωνία κάποιος θα μπορούσε να αναρωτηθεί αν τα module είναι μια διαχωρισμένη διεργασία ή μπορεί να υλοποιηθεί ως μία διαχωρισμένη διεργασία. Στον Apache το module δεν είναι μια διαχωρισμένη διεργασία. Ωστόσο μερικά module μπορεί να αναπαράγουν νέα «παιδιά» προκειμένου να κάνουν την δουλειά τους. Ένα παράδειγμα είναι το module `mod_cgi`, το οποίο χειρίζεται τα `cgi script`. Πρέπει να αναπαράγει ένα νέο «παιδί» για να εκτελέσει το πραγματικό `cgi script` (μετά από κατάλληλη ανακατεύθυνση της τυποποιημένης εισόδου και εξόδου για τη διαδικασία «παιδί») και να περιμένει μέχρι να τελειώσει. Αλλά αυτό είναι χαρακτηριστικό του `mod_cgi`, πολλά άλλα module δεν χρειάζεται να αναπαράγουν «παιδιά».

Ένα διαφορετικό είδος module είναι αυτό που, αν και δεν είναι διαχωρισμένη διεργασία και δεν αναπαράγει «παιδιά», επικοινωνεί μέσω μηχανισμούς IPC (Inter-Process Communication) ή sockets μέσα με μία διαφορετική διεργασία (που θα μπορούσε, για παράδειγμα, να βρίσκεται σε διαφορετικό μηχάνημα). Παράδειγμα ενός τέτοιου module είναι ένα module εξουσιοδότησης το οποίο επικοινωνεί με έναν εξυπηρετητή ο οποίος διαχειρίζεται πληροφορίες χρηστών και κωδικούς. Μέχρι και το CGI module (`mod_cgi`) μπορεί να υλοποιηθεί με αυτόν τον τρόπο (δηλαδή το πραγματικό `cgi script` να εκτελεστεί ως μία εντελώς διαφορετική

διεργασία και όχι ως «παιδί») το οποίο θα έχει ως αποτέλεσμα βελτιωμένη ασφάλεια, αλλά το πλεόνασμα (overhead) της επικοινωνίας θα λειτουργήσει ως αντίβαρο.

## Γλώσσα PHP

---

### **Τι είναι η PHP;**

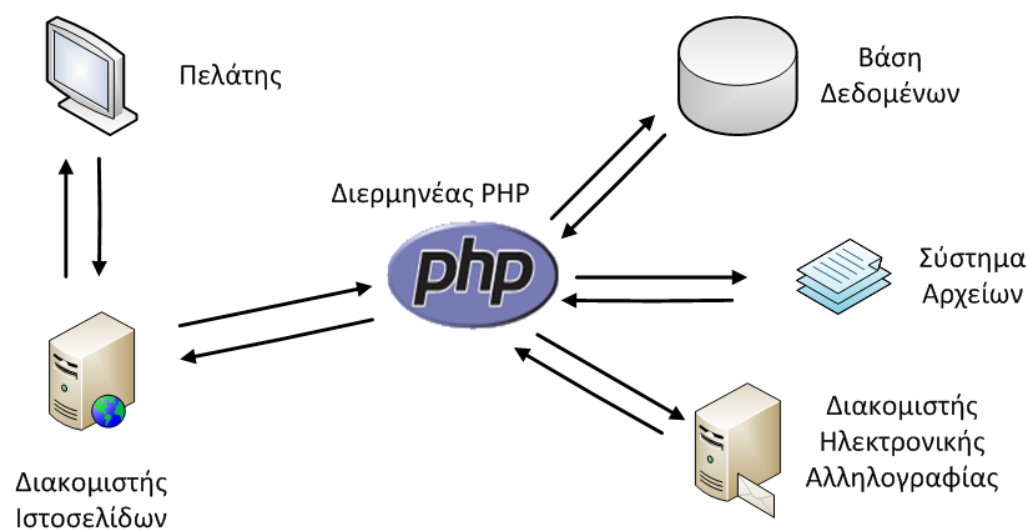
Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι γνωστή και ως: HyperText preprocessor. Είναι μια server-side (εκτελείτε στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντί αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές κτλ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες. Η PHP χρησιμοποιείται σε παραπάνω από 13 εκατομμύρια τομείς (domains) (σύμφωνα με έρευνα της Netcraft). Το συντακτικό της είναι παρόμοιο με αυτό της C, η PHP όμως είναι απλούστερη της C διότι δεν χρησιμοποιεί μερικές από τις πιο δύσκολες έννοιες της C. Επίσης η PHP δεν εμπεριέχει τις χαμηλού επιπέδου δυνατότητες προγραμματισμού της C διότι είναι σχεδιασμένη να προγραμματίζει ιστοσελίδες όπου δεν απαιτούνται αυτές οι δυνατότητες. Το δυνατό σημείο της PHP είναι η ικανότητα της να αλληλεπιδρά με βάσεις δεδομένων, η PHP υποστηρίζει σχεδόν όλες τις γνωστές (και μερικές άγνωστες) βάσεις δεδομένων.

### **Πως λειτουργεί η PHP;**

Η PHP είναι μία ενσωματωμένη γλώσσα scripting και χρησιμοποιείται σε ιστοσελίδες. Αυτό σημαίνει ότι ο κώδικας PHP είναι ενσωματωμένος στον κώδικα HTML. Χρησιμοποιούμε τις λέξεις κλειδιά (tags) της HTML για να επισυνάψουν τη γλώσσα PHP που έχουμε ενσωματώσει στο αρχείο HTML, με τον ίδιο τρόπο που θα χρησιμοποιούσαμε άλλες λέξεις κλειδιά της HTML. Δημιουργούμε και επεξεργαζόμαστε ιστοσελίδες που περιέχουν τη γλώσσα PHP, με τον ίδιο τρόπο που δημιουργούμε και επεξεργαζόμαστε ιστοσελίδες που περιέχουν τη γλώσσα HTML. Η PHP δουλεύει σε συνεργασία με το διακομιστή ιστοσελίδων (web server). Ο διακομιστής ιστοσελίδων είναι το λογισμικό που παραδίδει ιστοσελίδες στο κοινό. Όταν πληκτρολογούμε ένα URL στο φυλλομετρητή, στέλνουμε μήνυμα στο διακομιστή στο ίδιο URL, ζητώντας να μας σταλθεί ένα αρχείο HTML. Ο διακομιστής

ανταποκρίνεται, στέλνοντας το προαπαιτούμενο αρχείο. Ο φυλλομετρητής διαβάζει το HTML αρχείο και εμφανίζει την ιστοσελίδα. Μπορούμε επίσης να ζητήσουμε από τον διακομιστή να μας στείλει ένα αρχείο, όταν πατήσουμε πάνω σε ένα σύνδεσμο μέσα σε μία ιστοσελίδα.

Όταν η PHP εγκατασταθεί, ο διακομιστής έχει ρυθμιστεί να αποδέχεται συγκεκριμένες προεκτάσεις (extensions) αρχείων, τα οποία περιέχουν αναφορές της PHP. Συχνά αυτές οι προεκτάσεις είναι της μορφής .php ή .html, αλλά οποιαδήποτε προέκταση μπορεί να χρησιμοποιηθεί. Όταν ο web server λαμβάνει μία αίτηση για ένα αρχείο με την ορισμένη προέκταση, στέλνει τις αναφορές HTML ως είναι, αλλά οι αναφορές της PHP επεξεργάζονται από το λογισμικό της PHP πριν σταλούν πίσω στον αιτούντα. Όταν γίνεται επεξεργασία των αναφορών της PHP, μόνο το αποτέλεσμα στέλνεται από το διακομιστή στο φυλλομετρητή (web browser). Οι αναφορές της PHP δεν συμπεριλαμβάνονται στο αποτέλεσμα που έχει αποσταλεί στο φυλλομετρητή, οπότε ο κώδικας PHP είναι ασφαλής και διαφανής προς το χρήστη.



Εικόνα 13: Διερμηνέας PHP

Για παράδειγμα, σε αυτή την απλή δήλωση:

```
<?php echo "<p> Χαιρετίσματα από τον διακομιστή";?>
```

Η `<?php` είναι η λέξη κλειδί (tag) ανοίγματος της PHP και η `?>` είναι tag κλεισίματος. Η `echo` είναι μια οδηγία της PHP που αναφέρει το επερχόμενο κείμενο. Το λογισμικό της PHP επεξεργάζεται τις αναφορές της PHP και βγάζει το εξής αποτέλεσμα:

```
<p> Χαιρετίσματα από τον διακομιστή
```

Η οποία είναι μία κανονική αναφορά HTML. Αυτή η αναφορά HTML έχει τώρα παραδοθεί στο φυλλομετρητή. Ο φυλλομετρητής ερμηνεύει την αναφορά ως

κώδικα HTML και απεικονίζει το κείμενο «Χαιρετίσματα από τον διακομιστή» σε μία παράγραφο μέσα σε μια ιστοσελίδα. Η αναφορά της PHP δεν παραδίδεται στο φυλλομετρητή, οπότε ο χρήστης δεν τη βλέπει ποτέ. Η συνεργασία PHP και διακομιστή πρέπει να είναι στενή. Η PHP δεν εμπεριέχεται σε όλους τους διακομιστές, αλλά μπορεί να λειτουργήσει με τους πιο γνωστούς. Η PHP έχει αναπτυχθεί σαν project από το Apache software Foundation, συνεπώς δουλεύει καλύτερα με τον Apache. Η PHP δουλεύει επίσης με τον Microsoft IIS/PWS, iPlanet (πρώην Netscape Enterprise Server) και άλλους. Αναφέρθηκε και πιο πάνω ότι η PHP δουλεύει καλύτερα με τον Apache, οπότε αν έχουμε τη δυνατότητα να επιλέξουμε διακομιστή ο Apache θα ήταν η καλύτερη επιλογή, λόγω του ότι είναι δωρεάν, ανοιχτού κώδικα, σταθερός και δημοφιλής.

## Η ιστορία της PHP

### PHP/FI

Η γλώσσα PHP είναι διάδοχος ενός παλαιότερου προϊόντος που ονομαζόταν PHP/FI, Η PHP/FI δημιουργήθηκε από τον Rasmus Lerdorf το 1995, αρχικά σαν ένα απλό σετ από scripts της γλώσσας Perl για να παρακολουθεί την επισκεψιμότητα του διαδικτυακού βιογραφικού του. Αυτό το σετ των scripts τα ονόμασε «Personal Home Page Tools». Καθώς περισσότερη λειτουργικότητα απαιτούνταν, ο Rasmus έγραψε μία μεγαλύτερη εφαρμογή του σε κώδικα C η οποία ήταν ικανή να επικοινωνεί με βάσεις δεδομένων και επέτρεπε στους χρήστες να αναπτύσσουν δυναμικές διαδικτυακές εφαρμογές. Ο Rasmus επέλεξε να κοινοποιήσει τον πηγαίο κώδικα της PHP/FI έτσι ώστε οποιοσδήποτε να μπορεί να τον δει, το χρησιμοποιήσει, να διορθώσει σφάλματα (bugs) και να τον εξελίξει.

Η PHP/FI (Personal Home Page/ Forms Interpreter) περιείχε μερική από την βασική λειτουργικότητα της σημερινής PHP. Είχε μεταβλητές όμοιες με αυτές της γλώσσας Perl, αυτόματη μεταγλώττιση των μεταβλητών Form και ενσωματωμένο HTML συντακτικό. Το συντακτικό της καθεαυτού έμοιαζε με αυτό της Perl, μολονότι ήταν πολύ περισσότερο περιορισμένη, απλή, και κάπως ασυνεπής.

Το 1997, η PHP/FI 2.0, η δεύτερη έκδοση της εφαρμογής σε C, είχε οπαδούς μερικές χιλιάδες χρήστες στον κόσμο (εκτίμηση), με περίπου 50.000 domains να δηλώνουν ότι την έχουν εγκαταστημένη, που υπολογίζεται στο 1% των domains στον Διαδίκτυο. Ενώ υπήρχαν μερικοί άνθρωποι που συνέβαλαν κομμάτια κώδικα στο project, ήταν όμως ακόμα ένα project του ενός ατόμου.

Η PHP/FI 2.0 ανακοινώθηκε επίσημα τον Νοέμβρη του 1997, αφού ξόδεψε την περισσότερη ζωή της σε δοκιμαστικές (Beta) εκδόσεις. Σύντομα διαδέχθηκε από τις πρώτες αρχικές (Alpha) εκδόσεις της PHP 3.0.

## Παράδειγμα κώδικα PHP/FI

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

## PHP 3

Η PHP 3.0 ήταν η πρώτη έκδοση που μοιάζει στην PHP έτσι όπως την γνωρίζουμε σήμερα. Δημιουργήθηκε από τον Andi Gutmans και τον Zeev Suraski το 1997 όπου την ξανά έγραψαν από την αρχή διότι βρήκαν την PHP/FI 2.0 να υστερεί σοβαρά για ανάπτυξη εφαρμογών e-Επιχειρηματικότητας, όταν δούλευαν σε ένα πανεπιστημιακό project. Σε μια προσπάθεια συνεργασίας και εκκίνησης της ανάπτυξης πάνω στην υπάρχον βάση χρηστών (user-base) της PHP/FI , ο Andi, Rasmus και ο Zeev αποφάσισαν να συνεργαστούν και να ανακοινώσουν την PHP 3.0 σαν την επίσημη διάδοχο της PHP/FI 2.0, και η ανάπτυξη της PHP/FI ως επί τον πλείστον σταμάτησε.

Ένα από τα πιο δυνατά σημεία της PHP 3.0 ήταν τα δυνατά χαρακτηριστικά επεκτασιμότητας. Επιπλέον για να παρέχουν στους τελικούς χρήστες (end users) μία συμπαγής υποδομή για πολλές διαφορετικές βάσης δεδομένων, πρωτόκολλα και διασυνδέσεις προγραμματισμού εφαρμογών (APIs), τα χαρακτηριστικά επεκτασιμότητας της PHP 3.0 προσέλκυσαν δεκάδες προγραμματιστές να συμμετέχουν και να υποβάλουν νέα ενθέματα (module). Αναμφισβήτητα αυτό ήταν το κλειδί της τεράστιας επιτυχίας της PHP 3.0. Άλλα χαρακτηριστικά κλειδί που εισήχθησαν στην PHP 3.0 ήταν η υποστήριξη αντικειμενοστραφούς συντακτικού και το κατά πολύ πιο δυνατό και συνεπές συντακτικό της γλώσσας.

Η ολοκληρωτικά νέα γλώσσα κυκλοφόρησε κάτω από νέο όνομα, έτσι καταργήθηκε ο υπαινιγμός της περιορισμένης προσωπικής χρήσης που υπονοούσε το όνομα PHP/FI. Ονομάστηκε απλά «PHP», με το όνομα να είναι ένα αναδρομικό ακρωνύμιο - PHP: Hypertext Preprocessor.

Στα τέλη του 1998, η PHP μεγάλωσε βάση εγκατάστασης σε δεκάδες χιλιάδες χρήστες (εκτίμηση) και εκατοντάδες χιλιάδες δικτυακοί τόποι δήλωναν να την έχουν εγκατεστημένη. Στην κορύφωση της η PHP 3.0 ήταν εγκατεστημένη σε περίπου το 10% των διακομιστών ιστοσελίδων (web servers) στο Διαδίκτυο.

Η PHP 3.0 κυκλοφόρησε επίσημα το Ιούνιο του 1998, αφού πέρασε 9 μήνες ελεύθερης δοκιμασίας.

#### **PHP 4**

Το χειμώνα του 1998, λίγο μετά την επίσημη κυκλοφορία του PHP 3.0, ο Andi Gutmans και ο Zeev Suraski ξεκίνησαν να ξαναγράψουν τον πυρήνα της PHP. Ο σχεδιασμός είχε στόχο να βελτιώσει την απόδοση πολύπλοκων εφαρμογών και την «αρθρωτή» δομή της code base της PHP. Τέτοιες εφαρμογές μπορούσαν να γίνουν δυνατές λόγω των νέων χαρακτηριστικών και την ευρεία ποικιλία υποστήριξης από άλλους κατασκευαστές (third party) βάσεων δεδομένων και API, αλλά η PHP 3.0 δεν ήταν σχεδιασμένη να χειρίζεται τέτοιες πολύπλοκες εφαρμογές αποτελεσματικά. Η νέα μηχανή, ονομάστηκε «Zend Engine» (αποτελείται από τα αρχικά τους ονόματα, Zeev και Andi), η οποία επίτευξε αυτούς τους σχεδιαστικούς στόχους και πρώτο παρουσιάστηκε στα μέσα του 1999.

Η PHP 4.0, βασίστηκε σε αυτή την μηχανή, και ήρθε με μια ευρεία γκάμα από επιπλέον νέα χαρακτηριστικά, επίσημα κυκλοφόρησε τον Μάιο 2000, σχεδόν δύο χρόνια μετά από την κυκλοφορία του προκάτοχού του, PHP 3.0. Εκτός από την άκρως βελτιωμένη επίδοση του, η PHP 4.0 συμπεριλαμβάνει άλλα σημαντικά χαρακτηριστικά όπως υποστήριξη για ακόμα περισσότερους διαδικτυακούς εξυπηρετητές, συνεδρίες HTTP, ενταμίευση εξόδου (output buffering), περισσότερο ασφαλής τρόπος χειρισμού της παρέμβασης του χρήστη και μερικές νέες δομές της γλώσσας.

Σήμερα, η PHP χρησιμοποιείται από εκατοντάδες χιλιάδες προγραμματιστές (εκτιμώμενο) και έχουν εγκατεστημένο μερικά εκατομμύρια ιστοσελίδες, το οποίο προϋπολογίζεται για περισσότερο από 20% των domains στο διαδίκτυο.

Η ομάδα προγραμματιστών της PHP περιλαμβάνει δεκάδες προγραμματιστές, επίσης δεκάδες άλλοι δουλεύουν σε project σχετικά με την PHP όπως το PEAR και το project τεκμηρίωσης.

#### **PHP 5**

Η PHP 5 κυκλοφόρησε τον Ιούλιο του 2004 μετά από μακροχρόνια εξέλιξη και αρκετές πρώιμες εκδόσεις. Κυρίως καθοδηγείται από τον πυρήνα της, την Zend Engine 2.0 με ένα νέο μοντέλο αντικειμένων και δεκάδες άλλα νέα χαρακτηριστικά.

### **Χαρακτηριστικά της PHP 5**

#### **Νέο αντικειμενοστραφές μοντέλο**

Όταν ο Zeev Suraski πρόσθεσε το αντικειμενοστραφές συντακτικό επί των ημερών της PHP 3, προστέθηκε ως επιπλέον συντακτικό για πρόσβαση σε συλλογές ("syntactic sugar for accessing collections"). Το αντικειμενοστραφές μοντέλο επίσης

υποστήριζε κληρονομικότητα και επέτρεπε μία τάξη (και αντικείμενο) να συναθροίζει μεθόδους και ιδιότητες, αλλά τίποτα παραπάνω. Όταν ο Zeev και ο Andi ξαναέγραψαν την μηχανή script για την PHP 4, ήταν μια εντελώς νέα μηχανή, πολύ πιο γρήγορη, πολύ πιο σταθερή και με πολλά χαρακτηριστικά. Ωστόσο το αντικειμενοστραφές μοντέλο που πρωτοεισήχθη στην PHP 3, δεν πειράχθηκε.

Παρόλο που το μοντέλο αντικειμένων είχε σοβαρούς περιορισμούς χρησιμοποιούνταν ευρέως σε όλον τον κόσμο, συχνά σε πολύ μεγάλες εφαρμογές PHP. Αυτή η εντυπωσιακή χρήση του αντικειμενοστραφούς προγραμματισμού (Object-Oriented Programming paradigm) με την PHP 4 παρόλο τις αδυναμίες του έγινε το επίκεντρο για την επόμενη έκδοση της PHP.

Ο μεγαλύτερος περιορισμός (που οδήγησε σε περαιτέρω περιορισμούς) είναι το γεγονός ότι η σημασιολογία αντιγραφής των αντικειμένων ήταν η ίδια με αυτή των μητρικών (native) τύπων. Πως αυτό πραγματικά επηρεάζει των προγραμματιστή PHP; Όταν θα όριζες μία μεταβλητή (η οποία δείχνει προς ένα αντικείμενο) σε μια άλλη μεταβλητή, ένα αντίγραφο του αντικειμένου να δημιουργηθεί. Αυτό δεν επηρεάζει μόνο την επίδοση αλλά συνήθως οδηγεί σε δυσνόητη συμπεριφορά και σφάλματα σε εφαρμογές που είναι γραμμένες σε PHP 4, γιατί πολύ προγραμματιστές πίστευαν ότι και οι δύο μεταβλητές θα έδειχναν στο ίδιο αντικείμενο. Οι μεταβλητές δείχνουν σε ξεχωριστά αντίγραφα του ίδιου αντικειμένου, αλλάζοντας το ένα δεν θα γινόταν η αλλαγή και στο άλλο.

Για παράδειγμα:

```
class Person {
    var $name;
    function get_name() {
        return $this->name;
    }
    function set_name($name) {
        $this->name = $name;
    }
    function Person($name) {
        $this->set_name($name);
    }
}

function changename($Person, $name) {
    $Person ->set_name($name);
}

$Person = new Person("Χάρης");
changename ($Person, "Μάνος");
print $Person->get_name();
```

Ο παραπάνω κώδικας στην PHP 4 δεν θα εκτύπωνε το όνομα Μάνος αλλά το όνομα Χάρης. Αυτό γίνεται διότι περνάμε το αντικείμενο Person που το δείχνει η

μεταβλητή \$Person ως τιμή (by-value) έτσι το αντικείμενο Person αντιγράφετε, συνεπώς η λειτουργία (function) changename δουλεύει πάνω στο νέο (αντίγραφο) αντικείμενο Person το οποίο κρατάει την μεταβλητή \$name με τιμή Χάρης έτσι όπως του ορίσαμε όταν δημιουργήσαμε το πρώτο αντικείμενο και μετά αντιγράφηκε.

Αυτή η συμπεριφορά δεν είναι πολύ εννοητική, όπως πολλοί προγραμματιστές θα περίμεναν συμπεριφορά σαν της Java. Στην Java οι μεταβλητές πραγματικά κρατάνε μια λαβή (ή δείκτες) στο αντικείμενο και ως εκ τούτου όταν αντιγράφετε, μόνο η λαβή αντιγράφεται και όχι ολόκληρο το αντικείμενο.

Υπήρχαν δύο είδη χρηστών της PHP 4, αυτοί που γνώριζαν αυτό το πρόβλημα και αυτοί που δεν είχαν ιδέα. Οι τελευταίοι συνήθως δεν παρατηρούσαν το πρόβλημα και ο κώδικας τους ήταν γραμμένος έτσι ώστε να μην επηρεάζετε πραγματικά από την ύπαρξη του προβλήματος. Σίγουρα κάποιοι από αυτούς περάσανε νύχτες άγρυπνοι προσπαθώντας να επιλύσουν περίεργα σφάλματα τα οποία δεν μπορούσαν να εντοπίσουν. Αυτοί που γνώριζαν το πρόβλημα το παρέκαμπταν περνώντας και ορίζοντας πάντα αντικείμενα με αναφορές (by-reference). Έτσι εμπόδιζαν την μηχανή να αντιγράψει το αντικείμενο αλλά αυτό σαν τακτική δημιουργούσε κώδικα γεμάτο σύμβολα & το οποίο έκανε τον κώδικα να είναι δυσανάγνωστος.

Το παλιό μοντέλο αντικειμένων όχι μόνο οδηγούσε στο παραπάνω πρόβλημα αλλά επίσης οδηγούσε σε θεμελιώδη προβλήματα τα οποία εμπόδιζαν την εφαρμογή μερικών επιπρόσθετων χαρακτηριστικών πάνω στο υπάρχον μοντέλο αντικειμένων.

Στην PHP 5, η υποδομή του μοντέλου αντικειμένων ξαναγράφηκε να λειτουργεί με λαβές αντικειμένων. Εκτός και αν ρητά αντιγράψεις ένα αντικείμενο χρησιμοποιώντας την λέξη κλειδί clone δεν θα δημιουργήσεις ποτέ ένα αντίγραφο του αντικειμένου στο παρασκήνιο. Στην PHP 5 δεν υπάρχει λόγος να περάσεις ή ορίσεις ένα αντικείμενο με αναφορά.

Υποσημείωση: Ορισμός και πέρασμα με αναφορά υποστηρίζεται ακόμα, στην περίπτωση που θες πραγματικά να αλλάξεις το περιεχόμενο μιας μεταβλητής (είτε αντικείμενο ή άλλου τύπου).

## **Νέα αντικειμενοστραφής χαρακτηριστικά**

Τα νέα αντικειμενοστραφής χαρακτηριστικά είναι πάρα πολλά για να αναλυθούν. Έτσι θα σας παρουσιάσουμε τα κύρια από αυτά.

### **1. Κοινόι/Μυστικοί/Προστατευόμενοι (public/private/protected) τροποποιητές πρόσβασης για μεθόδους και ιδιότητες.**

Επιτρέπουν την χρήση κοινών αντικειμενοστραφών τροποποιητών πρόσβασης για να ελέγχουν την πρόσβαση των μεθόδων και ιδιοτήτων.

Για παράδειγμα:



```

class MyClass {
    private $id = 13;

    public function get_id() {
        return $this->id;
    }
}

```

## 2. Ενιαίο όνομα κατασκευαστή (constructor) `__construct()`

Αντί το όνομα του κατασκευαστή να είναι το όνομα της τάξης, θα πρέπει να δηλώνεται ως `__construct()`, κάνοντας το ευκολότερο να μετατοπιστούν τάξεις μέσα στις κληρονομικότητες των τάξεων.

Για παράδειγμα:

```

class MyClass {
    function __construct() {
        print "inside constructor";
    }
}

```

## 3. Υποστήριξη καταστροφεία (destructor) αντικειμένων ορίζοντας μια μέθοδο `__destruct()`

Επιτρέπει τον ορισμό μιας μεθόδου καταστροφής αντικειμένου που εκτελείτε όταν ένα αντικείμενο καταστρέφεται.

Για παράδειγμα:

```

class MyClass {
    function __destruct() {
        print "Destroying object";
    }
}

```

## 4. Διεπαφές (interfaces)

Δίνει την δυνατότητα για μία τάξη να εκπληρώσει παραπάνω από μία σχέση is-a. Μία τάξη μπορεί να κληρονομήσει από μία μόνο τάξη αλλά μπορεί να υλοποιήσει όσες διεπαφές θέλει.

Για παράδειγμα:

```

interface Display {
    function display();
}

class circle implements Display {
    function display() {
        print "Displaying circle";
    }
}

```

## 5. Τελεστής instanceof

Υποστήριξη σε επίπεδο γλώσσας για έλεγχο σχέσεων is-a. Η λειτουργία is\_a() της PHP 4 είναι πλέον μη συνιστώμενη (deprecated).

Για παράδειγμα:

```
if ($obj instanceof circle) {
    print '$obj is a circle';
}
```

## 6. Final μέθοδοι

Η λέξη κλειδί final μας επιτρέπει να σημαδέψουμε μεθόδους έτσι ώστε μια τάξη που κληρονομεί να μην μπορεί να την υπερφορτώσει (overload).

Για παράδειγμα:

```
class MyClass {
    final function getBaseClassName() {
        return __CLASS__;
    }
}
```

## 7. Final τάξεις

Άμα ορίσουμε μία τάξη ως final τότε δεν μπορεί να κληρονομηθεί. Για παράδειγμα ο παρακάτω κώδικας θα μας έδινε σφάλμα κατά την εκτέλεση του:

```
final class FinalClass{
}

class BogusClass extends FinalClass {
}
```

## 8. Ρητή αντιγραφή (κλωνοποίηση) αντικειμένων.

Για να αντιγράψεις πιστά ένα αντικείμενο θα πρέπει να χρησιμοποιήσεις την λέξη κλειδί clone. Μπορούμε επίσης να δηλώσουμε μία μέθοδο \_\_clone() η οποία καλείται κατά την διάρκεια της αντιγραφής (αφού οι ιδιότητες έχουν αντιγραφεί από το πρωτότυπο αντικείμενο).

Για παράδειγμα:

```
class MyClass{
    function __clone() {
        print "Object is being cloned";
    }
}

$obj = new MyClass();
clone $obj;
```

## 9. Σταθερές αντικειμένων

Οι ορισμοί τάξεων πλέον μπορούν να εμπεριέχουν σταθερές τιμές και αναφέρονται χρησιμοποιώντας την τάξη.

```
class MyClass {
    const SUCCESS = "success";
    const FAILURE = "failure";
}
print MyClass::SUCCESS;

print MyClass::FAILURE;
```

## 10. Στατικά Μέλη (members)

Οι δηλώσεις τάξεων πλέον μπορούν να περιέχουν στατικά μέλη (ιδιότητες), με πρόσβαση μέσω της τάξης. Συνήθης χρησιμότητα των στατικών μελών είναι στο πρότυπο Σόλο ή πρότυπο μονοσύνολο (Singleton pattern).

Για παράδειγμα:

```
class Singleton{
    static private $instance = NULL;

    private function __construct() {
    }

    static public function get_Instance() {
        if (self::$instance == NULL) {
            self::$instance = new Singleton();
        }
        return self::$instance;
    }
}
```

## 11. Στατικές μέθοδοι (methods)

Πλέον μπορούμε να ορίσουμε μεθόδους ως στατικές επιτρέποντας τες να κληθούν από μη-αντικείμενα. Οι στατικές μέθοδοι δεν ορίζουν την μεταβλητή \$this λόγω του ότι δεν είναι συνυφασμένες με κάποιο συγκεκριμένο αντικείμενο.

Για παράδειγμα:

```
class MyClass{
    static function helloworld() {
        print "Hello, world";
    }
}
MyClass::helloworld();
```

## 12. Αφαιρετικές (abstract) τάξεις.

Μία τάξη μπορεί να δηλωθεί ως αφηρημένη έτσι ώστε να είναι αδύνατη η δημιουργία αντικειμένου από αυτή. Ωστόσο μπορεί μια τάξη (ή τάξεις) να κληρονομήσουν από την αφηρημένη.

Για παράδειγμα:

```
abstract class MyBaseClass{
    function display() {
        print "Default display routine being called";
    }
}
```

## 13. Αφαιρετικές (abstract) μέθοδοι.

Μια μέθοδος μπορεί να δηλωθεί ως αφηρημένη, έτσι αναβάλετε ο ορισμός της σε μία τάξη κληρονόμο. Μια τάξη που εμπεριέχει αφηρημένες μεθόδους θα πρέπει να δηλωθεί ως αφηρημένη και αυτή.

Για παράδειγμα:

```
abstract class MyBaseClass {
    abstract function display();
}
```

## 14. Συμβουλές για τύπο Αντικειμένου (Class type hints)

Οι δηλώσεις των λειτουργιών μπορούν να περιέχουν συμβουλές για τις παραμέτρους των αντικειμένων. Αν μια λειτουργία κληθεί με λάθος τύπο τάξης τότε έχουμε σφάλμα.

Για παράδειγμα:

```
function expectsMyClass(MyClass $obj) {
}
```

## 15. Υποστήριξη για σπάσιμο αναφοράς (dereferencing) αντικειμένου οι οποίες επιστρέφονται από μεθόδους.

Στην PHP 4, δεν μπορούσες απευθείας να σπάσεις μια αναφορά σε αντικείμενο όπως αυτή επιστρέφεται από μία μέθοδο. Πρώτα θα έπρεπε να την ορίσεις σε μία νέα «ψευδό» (dummy) μεταβλητή και μετά να σπάσεις την αναφορά.

Για παράδειγμα στην PHP 4:

```
$dummy = $obj->method();
$dummy->method2();
```

Για παράδειγμα στην PHP 5:

```
$obj->method()->method2();
```

## 16. Σαρωτές (Iterators)

Η PHP 5 επιτρέπει τάξεις PHP και τάξεις προεκτάσεις PHP για την υλοποίηση μιας διεπαφής σαρωτή. Μόλις υλοποιηθεί αυτή η διεπαφή τότε θα μπορούσαμε να σαρώσουμε μια περίπτωση μιας τάξης χρησιμοποιώντας την λειτουργία `foreach()`.

Για παράδειγμα:

```
$obj = new MyIteratorImplementation();  
foreach ($obj as $value) {  
    print "value : ".$value;  
}
```

## 17. `__autoload()`

Πολύ προγραμματιστές που γράφουν αντικειμενοστραφές εφαρμογές δημιουργούν ένα αρχείο πηγαίου PHP κώδικα ανά ορισμό τάξης. Μια από τις μεγαλύτερες ενοχλήσεις είναι να πρέπει να γράψεις μια μεγάλη λίστα των αρχείων που πρέπει να εμπεριέχονται με τους ορισμούς των τάξεων στην αρχή του κάθε script (ένα για κάθε τάξη). Στην PHP 5 αυτό πλέον δεν είναι αναγκαίο. Μπορείς να ορίσεις μία λειτουργία `__autoload()` η οποία καλείται αυτόματα στην περίπτωση που προσπαθείς να χρησιμοποιήσεις μία τάξη η οποία δεν έχει οριστεί ακόμα. Καλώντας αυτή τη λειτουργία η μηχανή script δίνει μια τελευταία ευκαιρία να φορτώσει την τάξη πριν η PHP μας δώσει κάποιο σφάλμα.

Για παράδειγμα:

```
function __autoload($class_name) {  
    include_once($class_name . ".php");  
}  
  
$obj = new MyClass1();  
$obj2 = new MyClass2();
```

## Άλλα νέα χαρακτηριστικά

### 1. Χειρισμός εξαιρέσεων

Η PHP 5 προσθέτει την ικανότητα να γίνεται δομημένος χειρισμός των εξαιρέσεων με το πολύ γνωστό παράδειγμα (paradigm) `try/throw/catch` (προσπάθησε/πέτα/άρπαξε). Επιτρέπεται να `throw` (πετάς) αντικείμενα τα οποία κληρονομούν από την τάξη `Exception`.

Για παράδειγμα:

```

class SQLException extends Exception {
    public $problem;
    function __construct($problem) {
        $this->problem = $problem;
    }
}

try {
    ...
    throw new SQLException ("Couldn't connect to database ");
    ...
} catch (SQLException $e) {
    print "caught an SQLException with problem $obj->problem";
} catch (Exception $e) {
    print "Caught unrecognized exception";
}

```

Αυτή την στιγμή για λόγους αντίστροφης συμβατότητας οι περισσότερες εσωτερικές λειτουργίες δεν πετάνε εξαιρέσεις. Ωστόσο νέες προεκτάσεις χρησιμοποιούν αυτήν την ικανότητα. Επίσης παρόμοια με την ήδη υπάρχουσα λειτουργία `set_error_handler()` πλέον μπορεί να χρησιμοποιηθεί η `set_exception_handler()` για να `catch` (πιάσουμε) τις μη χειριζόμενες εξαιρέσεις πριν τερματιστεί το script.

## 2. Foreach με αναφορές

Στην PHP 4, δεν μπορούσες να σαρώσεις έναν πίνακα και να τροποποιήσεις τις τιμές του. Η PHP 5 το υποστηρίζει αυτό επιτρέποντας να σαρώσεις με αναφορές (&), έτσι οι που τιμές αλλάζουν με την σάρωση αλλάζουν στον πίνακα που σαρώνει.

```

foreach ($array as &$value) {
    if ($value === "NULL") {
        $value = NULL;
    }
}

```

## 3. Προεπιλεγμένες τιμές για τις παραμέτρους με αναφορά

Στην PHP 4, προεπιλεγμένες τιμές μπορούσαν να δοθούν μόνο σε παραμέτρους που περνιούνται με τιμή (by-value). Πλέον υποστηρίζεται να δοθούν προεπιλεγμένες τιμές και σε παραμέτρους που περνιούνται με αναφορά (by-value).

Για παράδειγμα:

```

function my_func(&$arg = null) {
    if ($arg === NULL) {
        print '$arg is empty';
    }
}

```

```
}  
my_func();
```

## Γενικές αλλαγές στην PHP

### XML και Web Services

Ακλουθώντας τις αλλαγές της γλώσσας, οι ενημερώσεις στην XML μέσα στην PHP 5 είναι πιθανών οι πιο σημαντικές και συναρπαστικές. Η ενισχυμένη λειτουργικότητα XML στην PHP 5 την τοποθετεί στο ίδιο επίπεδο με άλλες τεχνολογίες του διαδικτύου σε μερικούς τομείς και σε άλλους τομείς υπερνικά.

### Τα θεμέλια

Η υποστήριξη της XML στην PHP 4 υλοποιήθηκε χρησιμοποιώντας μια ποικιλία από υποκείμενες βιβλιοθήκες XML. Υποστήριξη για SAX υλοποιήθηκε χρησιμοποιώντας την παλιά βιβλιοθήκη Expat, η XSLT υλοποιήθηκε χρησιμοποιώντας την βιβλιοθήκη Sablotron (ή χρησιμοποιώντας την βιβλιοθήκη libxml2 μέσω της επέκτασης DOM) και η DOM υλοποιήθηκε χρησιμοποιώντας την πιο ισχυρή βιβλιοθήκη libxml2 από το project GNOME.

Χρησιμοποιώντας μια ποικιλία από βιβλιοθήκες δεν έκανε την PHP 4 να διαπρέψει στο θέμα υποστήριξης της XML. Η συντήρηση ήταν φτωχή, τα νέα στάνταρ στην XML δεν υποστηρίζονταν πάντα, η επίδοση δεν ήταν τόσο καλή όσο θα μπορούσε να ήταν και η διαλειτουργικότητα μεταξύ των διαφόρων επεκτάσεων XML δεν υπήρχε.

Στην PHP 5, όλες οι επεκτάσεις XML ξανά γράφτηκαν να χρησιμοποιούν την εξάισια εργαλειοθήκη XML libxml2. Είναι μια πολύ πλούσια, με υψηλή συντήρηση και αποδοτική υλοποίηση των στάνταρ XML φέρνοντας την αιχμή της τεχνολογίας XML στην PHP. Όλες οι προαναφερμένες επεκτάσεις (SAX, DOM και XSLT) πλέον χρησιμοποιούν την libxml2 συμπεριλαμβανομένων των νέων επεκτάσεων SimpleXML και SOAP.

### SAX

Όπως αναφέρθηκε, η νέα υλοποίηση της SAX μεταπήδησε από τη χρήση της βιβλιοθήκης Expat στην libxml2. Παρά το γεγονός ότι η νέα επέκταση θα πρέπει να είναι συμβατή μπορεί να υπάρχουν κάποιες μικρές διαφορές. Οι προγραμματιστές οι οποίοι εξακολουθούν να θέλουν να εργαστούν με την βιβλιοθήκη Expat μπορούν να το κάνουν ρυθμίζοντας και κτίζοντας (building) την PHP αναλόγως (μη συνιστώμενο).

### DOM

Παρόλο που η υποστήριξη της DOM στην PHP 4 βασιζόταν ήδη στην βιβλιοθήκη libxml2, είχε αρκετά σφάλματα (bugs), είχε σφάλματα διαρροής μνήμης (memory leaks) και το API σε πολλές περιπτώσεις δεν ήταν συμμορφούμενο με την W3C

(W3C compliant). Η επέκταση DOM πέρασε από ένα διεξοδικό «λίφτινγκ» για την PHP 5. Όχι μόνο η επέκταση ως επί τον πλείστον ξαναγράφηκε αλλά πλέον είναι και συμμορφούμενο με την W3C. Για παράδειγμα, τα ονόματα των λειτουργιών πλέον χρησιμοποιούν studyCaps όπως περιγράφετε από τα στάνταρ της W3C καθιστώντας ευκολότερο για τον χρήστη να διαβάσει γενική τεκμηρίωση από την W3C και να υλοποιήσει αυτό που έμαθε, απευθείας στην PHP. Εξάλλου, η επέκταση DOM πλέον υποστηρίζει τρία είδη σχημάτων (schemas) για την επικύρωση της XML, DTD, XML Schema και RelaxNG.

Ως αποτέλεσμα αυτών των αλλαγών, κώδικας γραμμένος σε PHP 4 χρησιμοποιώντας DOM δεν θα τρέχει πάντα στην PHP 5. Ωστόσο, στις περισσότερες περιπτώσεις προσαρμόζοντας τα ονόματα των λειτουργιών στο νέο στάνταρντ πιθανός να λύσει το πρόβλημα.

## **XSLT**

Στην PHP 4, υπήρχαν δύο επεκτάσεις που υποστήριζαν μετασχηματισμούς XSL (XSL Transformations). Η πρώτη χρησιμοποιούσε την επέκταση Sablotron και η δεύτερη χρησιμοποιούσε την υποστήριξη XSLT στην επέκταση DOM. Στην PHP 5, μια νέα επέκταση XSL γράφηκε και, όπως αναφέρθηκε, είναι βασισμένη στην επέκταση libxml2. Από την PHP 5, η XSLT δεν λαμβάνει σαν παράμετρο το φύλλο στυλ της XSLT αλλά εξαρτάται από την επέκταση DOM να το φορτώσει, το φύλλο στυλ μπορεί να γραφεί στην κρυφή μνήμη (cached in memory), έτσι μπορεί να εφαρμοστεί σε πολλά έγγραφα εξοικονομώντας χρόνο εκτέλεσης.

## **SimpleXML**

Πιθανότατα κοιτώντας πίσω ένα με δύο χρόνια θα ήταν σαφές ότι η SimpleXML άλλαξε τον τρόπο με τον οποίο οι προγραμματιστές PHP δούλευαν με τα αρχεία XML. Η SimpleXML θα μπορούσε πραγματικά να λέγεται «XML για χαζούς (XML for Dummies)». Αντί να πρέπει να δουλέψει με τη DOM ή ακόμα χειρότερα με την SAX, ή SimpleXML παρουσιάζει τα αρχεία XML ως ένα αντικείμενο (native) PHP. Μπορείς να διαβάσεις, γράψεις ή σαρώσεις το αρχείο XML με ευκολία έχοντας πρόσβαση σε στοιχεία και χαρακτηριστικά.

Θεωρήστε το ακόλουθο αρχείο XML:

```
<clients>
<client>
  <name>Κώστας Παραδειγματίδης</name>
  <account_number>87234838</account_number>
</client>
<client>
  <name>Μαρία Εξεμελοπούλου</name>
  <account_number>72384329</account_number>
</client>
</clients>
```



Το παρακάτω κομμάτι κώδικα εκτυπώνει το όνομα (<name>) και τον αριθμό λογαριασμού (<account\_number>) του κάθε πελάτη (<client>):

```
$clients = simplexml_load_file('clients.xml');
foreach ($clients->client as $client)

{
    print "Ο/Η $client->name έχει αριθμό λογαριασμού:

        $client->account_number". "<br />";
}

```

Είναι προφανές το πόσο απλή είναι πραγματικά η SimpleXML.

Σε περίπτωση που χρειάζεται να γίνει κάτι πιο εξειδικευμένο στο αντικείμενο SimpleXML το οποίο δεν υποστηρίζεται από αυτή την ελαφριά επέκταση, μπορούμε να το μετατρέψουμε σε ένα δέντρο DOM καλώντας την λειτουργία `dom_import_simplexml()`, να το χειριστούμε με την DOM και να το μετατρέψουμε πάλι σε SimpleXML χρησιμοποιώντας τη λειτουργία `simplexml_import_dom()`. Χάρη στις δύο επεκτάσεις που χρησιμοποιούν την ίδια υποκείμενη βιβλιοθήκη XML η εναλλαγή μεταξύ αυτών των δύο έχει γίνει πραγματικότητα.

## SOAP

Επίσημη (native) υποστήριξη του SOAP στην PHP 4 έλειπε. Η συνηθέστερη υλοποίηση του SOAP ήταν του PEAR αλλά επειδή ήταν υλοποιημένη εξολοκλήρου σε PHP δεν μπορούσε να αποδώσει σαν μια εμφωλευμένη επέκταση γραμμένη σε C. Άλλες διαθέσιμες επεκτάσεις γραμμένες σε C δεν έφτασαν ποτέ στη σταθερότητα και ευρεία αποδοχή και υιοθέτηση, έτσι δεν συμπεριλήφθηκαν στην κεντρική διανομή της PHP 5.

Η υποστήριξη του SOAP στη PHP 5 ξαναγράφηκε εξολοκλήρου ως μία επέκταση σε C και αν και μόλις είχε ολοκληρωθεί σε πολύ προχωρημένο στάδιο της δοκιμαστικής φάσης (beta stage process), συμπεριλήφθηκε στην προεπιλεγμένη διανομή λόγω τις εις βάθος υλοποίησης του μεγαλύτερου μέρους του στάνταρ του SOAP.

Το παρακάτω κομμάτι κώδικα καλεί την λειτουργία `SomeFunction()` που ορίζεται σε ένα αρχείο WSDL:

```
$client = new SoapClient("some.wsdl");
$client->SomeFunction($a, $b, $c);

```

## Νέα επέκταση MySQLi (MySQL Improved)

Για την PHP 5, η εταιρία MySQL AB έγραψε μια νέα επέκταση MySQL η οποία επιτρέπει να επωφεληθούν οι χρήστες πλήρως από την νέα λειτουργικότητα στην MySQL 4.1 και μετά. Σε αντίθεση με την παλιά επέκταση MySQL, η νέα δίνει μία λειτουργική και μία αντικειμενοστραφές διεπαφή έτσι ώστε να μπορούν να επιλέξουν αυτό που προτιμάνε. Νέα χαρακτηριστικά που υποστηρίζονται από την

επέκταση αυτή περιλαμβάνουν προετοιμασμένες δηλώσεις (prepared statements) και ένωση μεταβλητών (variable binding), SSL και συμπιεσμένες συνδέσεις (compressed connections), έλεγχος συναλλαγής (transaction control), υποστήριξη αναπαραγωγής (replication support) και άλλα...

### **Επέκταση SQLite**

Η υποστήριξη για την SQLite πρωτοεμφανίστηκε στην σειρά PHP 4.3.x. Είναι μια εμφωλευμένη βιβλιοθήκη SQL η οποία δεν χρειάζεται εξυπηρετητή SQL και είναι ιδιαίτερα κατάλληλη για εφαρμογές που δεν χρειάζονται την επεκτασιμότητα ενός εξυπηρετητή SQL ή αν αναπτύσσετε η εφαρμογή σε κάποιον ISP που δεν δίνει πρόσβαση σε έναν εξυπηρετητή SQL. Σε αντίθεση με ότι υποδηλώνει το όνομα της η SQLite είναι πλούσια σε χαρακτηριστικά και υποστηρίζει συναλλαγές (transactions), μερική επιλογή (sub-selects), όψεις και μεγάλα αρχεία DB (Βάσεις Δεδομένων). Αναφέρετε ως χαρακτηριστικό της PHP 5 διότι πρωτοεμφανίστηκε στις τελευταίες εκδόσεις της σειράς PHP 4 και δεδομένου ότι εκμεταλλεύεται τις δυνατότητες της PHP 5 παρέχοντας μια αντικειμενοστραφές διεπαφή και υποστηρίζει σαρωτές.

### **Επέκταση Tidy**

Η PHP 5 συμπεριλαμβάνει υποστήριξη για την πολύ χρήσιμη βιβλιοθήκη Tidy. Επιτρέπει στους προγραμματιστές PHP να αναλύουν, διαγνώσουν, καθαρίσουν και επισκευάσουν αρχεία HTML. Η επέκταση Tidy υποστηρίζει λειτουργικές (functional) διεπαφές αλλά και αντικειμενοστραφές, επίσης το API της χρησιμοποιεί τον μηχανισμό εξαιρέσεων (exception mechanism) της PHP 5.

### **Επέκταση Perl**

Παρόλο που δεν συμπεριλαμβάνετε στο προεπιλεγμένο πακέτο PHP 5, η επέκταση Perl επιτρέπει στον χρήστη να καλέσει script γραμμένα σε Perl, να χρησιμοποιήσει αντικείμενα Perl και να χρησιμοποιήσει άλλες λειτουργίες της Perl από την PHP (natively). Αυτή η νέα επέκταση βρίσκεται στην αποθήκη (repository) PECL (PHP Extension Community Library).

## **Λοιπά νέα χαρακτηριστικά στην PHP 5:**

### **Νέος διαχειριστής μνήμης**

Η μηχανή Zend διαθέτει ένα νέο διαχειριστή μνήμης. Τα δύο βασικά προτερήματα είναι η καλύτερη υποστήριξη για πολυ-νηματικά (multi-threaded) περιβάλλοντα (Οι κατανομές δεν χρειάζεται να κάνουν κλειδώματα αμοιβαίου αποκλεισμού) και μετά από κάθε αίτηση που ελευθερώνει τα καταναμημένα μπλοκ μνήμης είναι πλέον πολύ πιο αποδοτικό. Δεδομένου ότι αυτό αποτελεί υποκείμενη υποδομή οι αλλαγές δεν μπορούν να παρατηρηθούν απευθείας από τον τελικό χρήστη.

### **Κατάργηση υποστήριξης για Windows 95**

Η PHP πλέον δεν υποστηρίζει την πλατφόρμα Windows 95 λόγω του ότι δεν υποστηρίζει την λειτουργικότητα που χρησιμοποιεί η PHP. Όπως η Microsoft έχει

επίσημα σταματήσει να υποστηρίζει την πλατφόρμα Windows 95 πάνω από ένα χρόνο τώρα, έτσι και η αναπτυξιακή κοινότητα της PHP αποφάσισε ότι αυτή ήταν μια σοφή απόφαση.

# ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ MySQL

---

## Πως το ΣΔΒΔ MySQL έφτασε ως εδώ;

### MySQL και mSQL

Το DBMS MySQL και mSQL είναι πολύ όμοια, φτηνά, ελαφριά και γρήγορα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων (RDBMS). Ωστόσο είναι διαφορετικά με πολλούς σημαντικούς τρόπους. Τα δύο συστήματα υποστηρίζουν προγραμματισμό σε C, Perl, Java (μέσω του Java DataBase Connectivity API [JDBC]) και Perl. Με τα εργαλεία που προσφέρουν το ΣΔΒΔ MySQL και mSQL σε αυτές τις γλώσσες είναι δυνατόν να δημιουργήσουμε μια πλήρης εφαρμογή πελάτη/διακομιστή και ιστότοπους βασισμένους σε βάσεις δεδομένων χωρίς να ξοδέψουμε μια περιουσία.

Η ανέξοδη και σε ορισμένες περιπτώσεις δωρεάν φύση του ΣΔΒΔ MySQL και mSQL δεν έρχονται χωρίς «κόστος». Κανένα σύστημα διαχείρισης βάσεων δεδομένων δεν υποστηρίζει το πλήρες φάσμα της SQL. Αυτές οι μηχανές στερούνται μερικά χαρακτηριστικά τα οποία μπορεί να χρειάζονται από πιο σύνθετες εφαρμογές. Για μερικές εφαρμογές θα πρέπει να δουλέψουμε λίγο πιο σκληρά από πλευράς πελάτη για την κάλυψη των αναγκών.

### Η ιστορία του ΣΔΒΔ mSQL

Πριν το 1994, θα ήσουν άτυχος αν ήθελες ένα σύστημα διαχείρισης βάσεων δεδομένων βασισμένο στην SQL χωρίς να πληρώσεις μεγάλα χρηματικά ποσά. Οι κύριες εμπορικές λύσεις για SQL ήταν η Oracle, Sybase και Informix. Αυτές οι μηχανές βάσεων δεδομένων είχαν σχεδιαστή να χειρίζονται τεράστιες ποσότητες δεδομένων με πολύ σύνθετες σχέσεις. Ήταν ισχυρές, πλούσιες σε χαρακτηριστικά, πολύ απαιτητικές σε πόρους και ακριβές. Εκείνες τις ημέρες μπορούσες να αγοράσεις έναν διακομιστή Pentium στα 200MHz για \$2.000. Οι πόροι που χρειάζονταν από αυτές τις μηχανές βάσεων δεδομένων κόστιζαν δεκάδες χιλιάδες δολάρια.

Οι μεγάλες εταιρίες και τα μεγαλύτερα Πανεπιστήμια δεν είχαν πρόβλημα να ξοδέψουν εκατομμύρια δολάρια το χρόνο για τέτοιους μεγάλους συνδυασμούς DBMS/διακομιστή (server). Ως μια μικρή επιχείρηση ή ως ατομικός χρήστης θα

έπρεπε να συμβιβαστής με μία αδύνατη εφαρμογή βάσεων δεδομένων. Μερικές φτηνές μηχανές βάσεων δεδομένων πελάτη/διακομιστή υπήρχαν εκείνη την εποχή, αλλά καμία από αυτές δεν χρησιμοποιούσε την SQL ως γλώσσα για ερωτήματα. Η πιο αξιοσημείωτη μηχανή βάσεων δεδομένων ήταν το ΣΔΒΔ Postgres το οποίο ήταν ένα ΣΔΒΔ που προήλθε από τις ίδιες ρίζες με το εμπορικό RDBMS Ingres . Το ΣΔΒΔ Postgres δίστιχος ερχόταν με παρόμοιες απαιτήσεις σε πόρους με τα αντίστοιχα εμπορικά ΣΔΒΔ χωρίς το πλεονέκτημα της SQL για γλώσσα ερωτημάτων. Εκείνη την εποχή το ΣΔΒΔ Postgres χρησιμοποιούσε μια παραλλαγή της γλώσσας QUEL που ονομαζόταν PostQUEL.

### **Ο David Hughes και το σύστημα Minerva**

Ο David Hughes (a.k.a Bambi) για ένα μέρος της διδακτορικής του διατριβής στο Πανεπιστήμιο Bond στην Αυστραλία ανέπτυξε ένα σύστημα για παρακολούθηση και διαχείριση μιας πληθώρας συστημάτων από μία ή περισσότερες τοποθεσίες. Το project αυτό ονομαζόταν Minerva Network Management System. Ένα βασικό κομμάτι του συστήματος αυτού ήταν ένα σύστημα διαχείρισης βάσεων δεδομένων για την αποθήκευση πληροφοριών για τα μηχανήματα που ήταν στο δίκτυο. Ως μαθητής Πανεπιστημίου χωρίς άμεση πρόσβαση σε κάποιον διακομιστή που να τρέχει κάποιες από τις εμπορικές σχεσιακές μηχανές βάσεων δεδομένων, ο Hughes στράφηκε στο ΣΔΒΔ Postgres ως την προφανές λύση στις ανάγκες του για μια βάση δεδομένων.

Οι συνάδελφοι του Hughes αρχικά του πρότειναν την SQL για στάνταρ γλώσσας ερωτημάτων για το σύστημα Minerva. Παρόλα αυτά η SQL ήταν και εξακολουθεί να είναι το πιο συντριπτικά αποδεκτό στάνταρ για γλώσσα ερωτημάτων. Τυποποιώντας στη SQL το σύστημα Minerva θα μπορούσε να εξυπηρετήσει τις ανάγκες ανθρώπων από ολόκληρο τον κόσμο εφόσον είχαν κάποιο σύστημα διαχείρισης βάσεων δεδομένων SQL εγκατεστημένο. Με άλλα λόγια η SQL θα έκθετε το σύστημα Minerva σε ένα ευρύτερο κοινό από την PostQUEL, η οποία περιοριζόταν στο ΣΔΒΔ Postgres. Όπως εξελίχθηκαν τα πράγματα, σήμερα μέχρι και το ΣΔΒΔ Postgres «μιλάει» SQL.

Ο Hughes ήταν σε άσχημη θέση, από τη μία είχε το στάνταρ της SQL από την άλλη την πρόσβαση σε μηχανή βάσεων δεδομένων SQL. Αν βάσιζε τη γλώσσα ερωτημάτων του συστήματος Minerva στην SQL, δεν θα είχε μηχανή βάσης δεδομένων διότι η αγορά ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων δεν ήταν επιλογή λόγω του κόστους. Ο Hughes πήρε μια δημιουργική προσέγγιση στο πρόβλημα. Αποφάσισε ότι η λύση ήταν να δημιουργήσει μια εφαρμογή που θα μετέφραζε SQL σε PostQUEL επιτόπου (on-the-fly). Αυτό το πρόγραμμα θα συλλάμβανε όλες τις δηλώσεις SQL που στέλνονταν από την Minerva, θα τα μετέτρεπε σε PostQUEL και μετά θα τα έστελνε τη PostQUEL στο

ΣΔΒΔ Postgres. Ο Hughes την δημιούργησε αυτήν την εφαρμογή και την ονόμασε miniSQL ή mSQL.

### **Από τον μεταφραστή PostQUEL στο Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS).**

Για κάμποσο καιρό αυτή η ρύθμιση λειτουργούσε καλά για τις ανάγκες του Hughes. Δεν ενδιέφερε το σύστημα Minerva τι σύστημα διαχείρισης βάσεων δεδομένων χρησιμοποιείται αρκεί να καταλαβαίνει SQL. Απ' όσο γνωρίζει το σύστημα Minerva το ΣΔΒΔ Postgres καταλάβαινε SQL λόγω της mSQL που βρισκόταν στην μέση για να χειρίζεται την μετάφραση της PostQUEL. Δυστυχώς καθώς η Minerva μεγάλωνε, έγινε και σημαντικά βραδύτερη. Τελικά κατέστη σαφές ότι το ΣΔΒΔ Postgres ή οποιοδήποτε άλλο μεγάλο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων δεν ήταν ικανό να στηρίξει τα μικρά σε σύνολο χαρακτηριστικά που απαιτούσε το σύστημα Minerva με τους περιορισμένους πόρους που είχε διαθέσιμους. Για παράδειγμα το σύστημα Minerva απαιτούσε πολλαπλές ταυτόχρονες συνδέσεις στην βάση δεδομένων, για να υποστηριχθεί κάτι τέτοιο το ΣΔΒΔ Postgres απαιτούσε πολλαπλά στιγμιότυπα (instances) του διακομιστή της βάσης δεδομένων να τρέχουν συγχρόνως. Επιπρόσθετα, μερικοί πιθανοί συνεισφέροντες στο project Minerva δεν μπορούσαν να εμπλακούν επειδή το ΣΔΒΔ Postgres δεν υποστήριζε τα συστήματα τους και αυτοί από την άλλη δεν είχαν την επιλογή να αγοράσουν ακριβά συστήματα διαχείρισης βάσεων δεδομένων βασισμένα στην SQL.

Εν όψει αυτών των προβλημάτων ο Hughes αξιολόγησε εκ νέου την απόφαση του να χρησιμοποιήσει το ΣΔΒΔ Postgres. Δεδομένου ότι ήταν μεγάλη και πολύπλοκη ήταν πιθανόν να ήταν υπερβολικά περίπλοκη για τις ανάγκες του συστήματος Minerva. Τα περισσότερα ερωτήματα του συστήματος Minerva ήταν απλές δηλώσεις INSERT, DELETE και SELECT. Όλα τα άλλα πράγματα τα οποία κόστιζαν στο ΣΔΒΔ Postgres ως προς τις επιδόσεις τις απλά δεν ήταν απαραίτητα για το Minerva.

Ο Hughes ήδη είχε την mSQL που έκανε μετάφραση σε SQL. Το μόνο που χρειαζόταν ήταν να προσθέσει την ικανότητα αποθήκευσης και ανάκτησης δεδομένων, έτσι είχε έναν διακομιστή βάσεων δεδομένων προσαρμοσμένο στις ανάγκες του. Αυτή η εξέλιξη οδήγησε στο ΣΔΒΔ mSQL που υπάρχει μέχρι και σήμερα.

### **Η ιστορία του ΣΔΒΔ MySQL**

Θα ήταν λάθος να χαρακτηρίσουμε το ΣΔΒΔ MySQL ως μια απλή αντίδραση στις αποτυχίες του ΣΔΒΔ mSQL. Ο εφευρέτης της, Mickael Widenius (a.k.a. Monty) εργαζόταν με βάσεις δεδομένων από το 1979, στην Σουηδική εταιρία TcX. Μέχρι πρόσφατα ο Widenius ήταν ο μοναδικός προγραμματιστής στην TcX. Το 1979 ανέπτυξε ένα εταιρικό (in-house) εργαλείο βάσεων δεδομένων που ονομάζεται UNIREG για διαχείριση βάσεων δεδομένων. Από το 1979 το UNIREG έχει ξανά γραφτεί σε αρκετές διαφορετικές γλώσσες και επεκτάθηκε για να χειρίζεται μεγάλες βάσεις δεδομένων.

Το 1994, η TcX ξεκίνησε να αναπτύσσει εφαρμογές βασισμένες στο Διαδίκτυο και χρησιμοποίησε το UNIREG για υποστήριξη αυτής της προσπάθειας. Δυστυχώς το UNIREG δημιουργούσε πολύ βάρος (overhead) για να είναι επιτυχής σε δυναμικά δημιουργημένες ιστοσελίδες. Η TcX ξεκίνησε να κοιτάει προς την SQL και το ΣΔΒΔ mSQL. Εκείνη την εποχή ωστόσο το ΣΔΒΔ mSQL βρισκόταν ακόμα στις εκδόσεις 1.x.. Το ΣΔΒΔ mSQL 1.x δεν υποστήριζε δείκτες, ως εκ τούτου η επίδοσή του σε σύγκριση με το UNIREG ήταν πτωχή.

Ο Widenius επικοινωνήσε με τον David Hughes τον συγγραφέα του ΣΔΒΔ mSQL για να δει αν ενδιαφερόταν να συνδέσει το ΣΔΒΔ mSQL με τον χειριστή του UNIREG B+ ISAM. Ο Hughes βρισκόταν ήδη στον δρόμο για το mSQL 2, ωστόσο είχε ήδη θέση την υποδομή αναφορών (indexing). Η TcX αποφάσισε να δημιουργήσει ένα διακομιστή βάσεων δεδομένων οποίος θα ήταν περισσότερο συμβατός με τις απαιτήσεις του.

Η TcX ήταν έξυπνη αρκετά για να μην προσπαθήσει να ξανά εφεύρει την ρόδα (reinvent the wheel). Έχτισε πάνω στο UNIREG και επένδυσε στον αυξανόμενο αριθμό των εργαλείων mSQL από άλλους κατασκευαστές (third party) γράφοντας ένα API στο σύστημα τους που, τουλάχιστον αρχικά, ήταν πρακτικά παρόμοιο με το API του mSQL. Αυτό είχε ως αποτέλεσμα ένας χρήστης mSQL ο οποίος ήθελε να μεταφερθεί στον πιο πλούσιο σε χαρακτηριστικά διακομιστή βάσεων δεδομένων της TcX να κάνει ασήμαντες αλλαγές στον υπάρχον κώδικα. Ο κώδικας που υποστήριζε αυτή τη νέα βάση δεδομένων εν τούτοις ήταν τελείως πρωτότυπος.

Το Μάιο του 1995 η TcX είχε μια βάση δεδομένων που πληρούσε τις εσωτερικές ανάγκες της (το ΣΔΒΔ MySQL 1.0). Ένας επιχειρηματικός εταίρος, ο David Axmark της Detron HB, ξεκίνησε να πιέζει την TcX να εκδώσει αυτόν τον διακομιστή στο Διαδίκτυο (Internet release). Ο στόχος μίας έκδοσης στο Διαδίκτυο ήταν να χρησιμοποιηθεί ένα επιχειρηματικό μοντέλο που πρωτοπορήθηκε (pioneered) από τον Peter Deutsch της εταιρίας Aladdin. Το αποτέλεσμα ήταν μια ευέλικτη πνευματική ιδιοκτησία (copyright) η οποία κάνει το ΣΔΒΔ MySQL «περισσότερο ελεύθερο» από το ΣΔΒΔ mSQL.

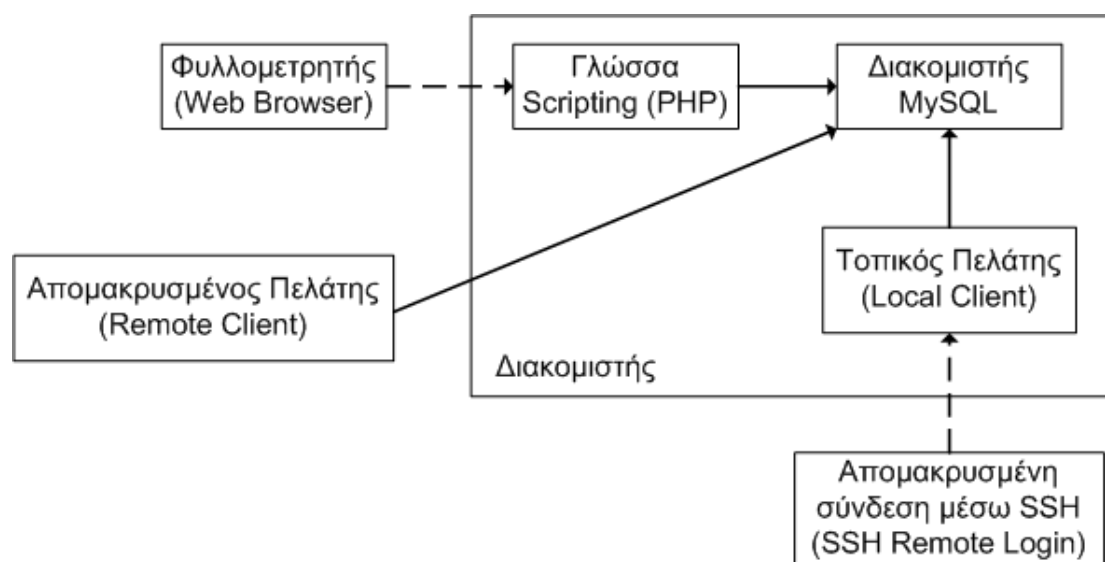
Όσο για το όνομα MySQL ο Widenius λέει “Δεν είναι εντελώς ξεκάθαρο από που προήλθε το όνομα MySQL. Ο βασικός κατάλογος της TcX και ένα μεγάλος αριθμός βιβλιοθηκών και εργαλείων είχαν το πρόθεμα «my» για παραπάνω από 10 χρόνια. Ωστόσο το όνομα της κόρης μου είναι επίσης My. Έτσι, πιο από τα δύο έδωσε το όνομα του στο ΣΔΒΔ MySQL είναι ακόμα ένα μυστήριο.”

Από την αρχική έκδοση στο Διαδίκτυο της MySQL, έχει κωδικοποιηθεί (ported to) για λειτουργικά συστήματα Unix, Win32 και OS/2. Η TcX εκτιμά ότι το σύστημα διαχείρισης βάσεων δεδομένων MySQL τρέχει σε περίπου 500.000 διακομιστές.

Το ΣΔΒΔ MySQL είναι μια ταχύτατα εξελισσόμενη πλατφόρμα βάσης δεδομένων λόγω της στρατιάς εθελοντών προγραμματιστών που βοηθάνε στην εξέλιξη της. Γι' αυτό δεν θα πρέπει να ξαφνιαστείτε αν σε λίγο καιρό αυτά που έχουν γραφτεί για αυτό το ΣΔΒΔ δεν ισχύουν πλέον. Την στιγμή που γράφτηκε αυτή η πτυχιακή το ΣΔΒΔ MySQL Community Edition (δωρεάν έκδοση) βρίσκεται στην έκδοση 5.1.50

## Πώς λειτουργεί το ΣΔΒΔ MySQL;

Το ΣΔΒΔ MySQL είναι ένα πρόγραμμα-διακομιστής βάσεων δεδομένων (συντά αποκαλείται και ως μηχανή βάσεων δεδομένων). Ο διακομιστής του MySQL και οι συσχετισμένοι πίνακες δεδομένων του, μπορούν και οι δύο να βρίσκονται εξίσου σε έναν υπολογιστή, αλλά η μηχανή των βάσεων δεδομένων μπορεί να διακομίσει τα δεδομένα που βρίσκονται στους πίνακες της σε μία πληθώρα τοποθεσιών.



Εικόνα 14: MySQL σε ένα διακομιστή

Ο διακομιστής MySQL έχει εγκατασταθεί σε έναν άλλο διακομιστή και μπορεί να υπάρξει πρόσβαση σε αυτόν κατ' ευθείαν από τον πελάτη της MySQL, ο οποίος στέλνει δηλώσεις ANSI SQL στο διακομιστή του ΣΔΒΔ MySQL για επεξεργασία και μετά παρουσιάζει το αποτέλεσμα από την επεξεργασία αυτή.

Οι πελάτες της μηχανής MySQL μπορεί να είναι:

- **Τοπικός πελάτης:**

Ένα πρόγραμμα που είναι στο ίδιο μηχάνημα που είναι και ο διακομιστής του ΣΔΒΔ MySQL. Ένα παράδειγμα αυτού είναι η γραμμή εντολών του πελάτη MySQL. Υπάρχουν επίσης τρίτοι πελάτες MySQL που έχουν GUI's μέσω των οποίων το ANSI SQL μπορεί να παραδοθεί στη μηχανή MySQL που τρέχει στη μνήμη του υπολογιστή.

- **Απομακρυσμένος πελάτης:**

Ένα πρόγραμμα σε ένα διαφορετικό μηχάνημα που μπορεί να συνδεθεί στο διακομιστή και να τρέξει τις αναφορές της SQL. Ο απομακρυσμένος πελάτης μπορεί να είναι ένας πελάτης απλού χαρακτήρα ή ένα GUI βασισμένο στον πελάτη MySQL (από άλλους κατασκευαστές – third party).

## **Τα χαρακτηριστικά και τα πλεονεκτήματα της MySQL**

### **Επεκτασιμότητα και Ευελιξία (Scalability and Flexibility)**

Τρέχει τα πάντα από:

- Βαθιά ενσωματωμένες εφαρμογές με ίχνος μόλις 1MB (Deeply embedded applications with a footprint of just 1MB), μέχρι και
- Μαζικές αποθήκες δεδομένων οι οποίες περιέχουν δεδομένα επιπέδου terabyte (Massive data warehouses holding terabytes of information).

### **Υψηλή απόδοση (High Performance)**

- Διαχωρισμός δεικτών και πινάκων (Table and Index Partitioning).
- Εξαιρετικά γρήγορο εργαλείων για φόρτωμα (Ultra-fast load utilities).
- Διακριτές κρυφές/προσωρινές μνήμες (Distinctive memory caches).
- Ευρετήρια με υποστήριξη πλήρους κειμένου (Full-text indexes).
- Και άλλα..

### **Υψηλή διαθεσιμότητα (High Availability)**

- Τρέχει υψηλής ταχύτητας αναπαραγωγή ρυθμίσεων κύριου/σκλάβου με αναπαραγωγή βασισμένη σε γραμμές και υβριδική αναπαραγωγή (high-speed master/slave replication configurations with Row-Based and Hybrid Replication).
- Ειδικευμένοι διακομιστές για Cluster που προσφέρουν στιγμιαία ανακατεύθυνση σε τυχόν σφάλμα (Specialized Cluster servers offering instant failover).

### **Υποστήριξη εύρωστων συναλλαγών (Robust Transactional Support)**

- Ολοκληρωμένη υποστήριξης συναλλαγών ACID (Complete ACID (atomic, consistent, isolated, durable) transaction support)
- Απεριόριστα κλειδώματα σε επίπεδο γραμμών (Unlimited row-level locking)
- Ικανότητα διανεμόμενων συναλλαγών (Distributed transaction capability)
- Υποστήριξη συναλλαγών πολλαπλών εκδόσεων (Multi-version transaction support)

### **Ισχυρά σημεία αποθήκης δεδομένων και ιστού (Web and Data Warehouse Strengths)**

- Υψηλή απόδοση μηχανής αναζήτησης (High-performance query engine).



- Τρομερά γρήγορη ικανότητα εισαγωγής δεδομένων, και (Tremendously fast data insert capability).
- Ισχυρή υποστήριξη για ειδικευμένες λειτουργίες ιστού, όπως γρήγορες αναζητήσεις πλήρους κειμένου (Strong support for specialized web functions, like fast full text searches).

### **Ισχυρή προστασία δεδομένων (Strong Data Protection)**

- Ισχυροί μηχανισμοί για τη διασφάλιση εισόδου μόνο εξουσιοδοτημένων χρηστών (Powerful mechanisms for ensuring only authorized users have access).
- Υποστήριξη SSH και SSL για αξιόπιστες και ασφαλές συνδέσεις (SSH and SSL support safe and secure connections).
- Ισχυρές λειτουργίες κρυπτογράφησης και αποκρυπτογράφησης (Powerful data encryption and decryption functions).

### **Περιεκτική ανάπτυξη εφαρμογών (Comprehensive Application Development)**

- Υποστήριξη για αποθηκευμένες διαδικασίες, σκανδάλες, λειτουργίες, όψεις, δρομείς, ANSI-standard SQL και πολλά άλλα. (Support for stored procedures, triggers, functions, views, cursors, ANSI-standard SQL).
- Βιβλιοθήκες σε μορφή plug-in για ενσωμάτωση βάσεων δεδομένων MySQL σε σχεδόν κάθε εφαρμογή (Plug-in libraries to embed MySQL database support into nearly any application).

### **Ευκολία στη διαχείριση (Management Ease)**

- Χρήση Προγραμματιστή Γεγονότων, αυτόματα προγραμματίζει κοινά επαναλαμβανόμενα καθήκοντα να εκτελούνται στον διακομιστή βάσεων δεδομένων (Use Event Scheduler automatically schedule common recurring SQL-based tasks to execute on the database server).

Ο μέσος χρόνος από το κατέβασμα του λογισμικού μέχρι την εγκατάσταση είναι λιγότερος από δεκαπέντε λεπτά (Average time from software download to complete installation is less than fifteen minutes).

# ΣΥΜΠΕΡΑΣΜΑΤΑ

---

Η ανάπτυξη ιστοσελίδων με τη χρήση συστημάτων διαχείρισης περιεχομένου μπορούν να μας βοηθήσουν να δημιουργήσουμε ιστότοπους επιπέδου portal που έχουν τα εξής οφέλη:

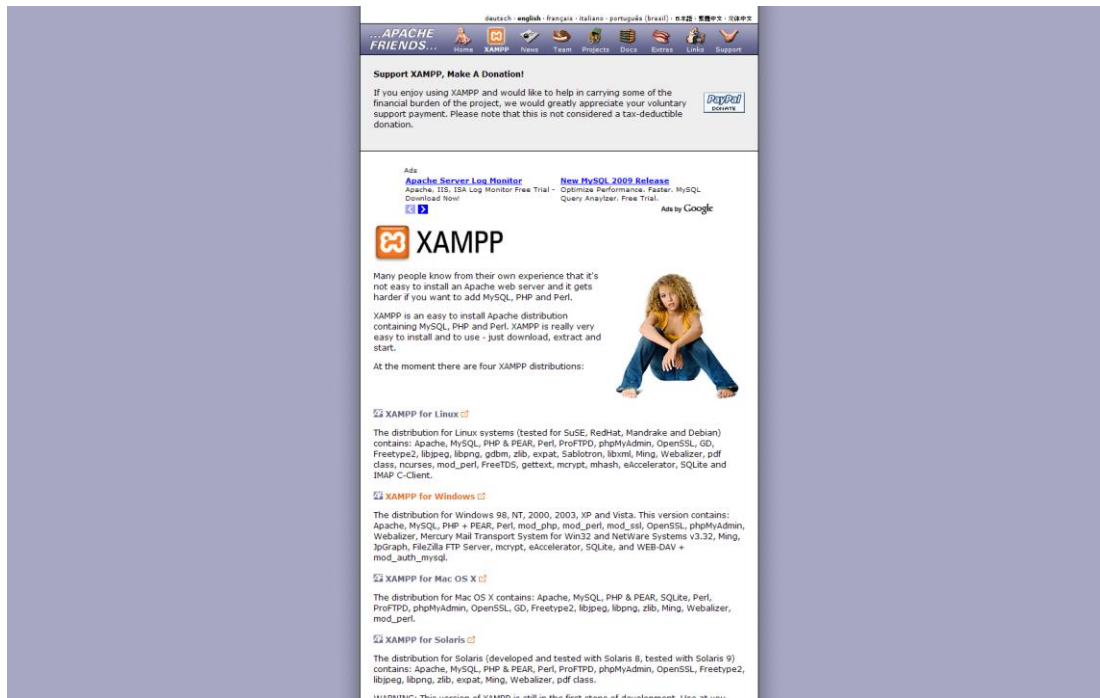
- Υπάρχουν πολλές εφαρμογές που μπορούμε να τις προσαρμόσουμε στην δικιά μας περίπτωση και να έχουμε ένα τέλειο αποτέλεσμα (παράδειγμα, ένα ημερολόγιο, ρολόι κτλ)
- Εκατοντάδες έτοιμα templates που μπορούμε να τα αγοράσουμε και να τοποθετήσουμε στο site μας.
- Δημοσιεύονται καινούριες εκδόσεις για να διορθώνουν τα προβλήματα των προηγούμενων.
- Διαθέτει πολλά extensions που μπορούν να κάνουν το site μας πολύ δυναμικό.

Όλο και περισσότερες σύγχρονες επιχειρήσεις και οργανισμοί έρχονται σήμερα σε αδιέξοδο λόγω της αδυναμίας τους να ανταποκριθούν στις ανάγκες για την διαχείριση της ολοένα αυξανόμενης πληροφορίας και του περιεχομένου που διαθέτουν. Η αστείρευτη εισαγωγή πληροφοριών και δεδομένων οδηγεί στη δημιουργία μεγάλων βάσεων δεδομένων με χιλιάδες εγγραφές και πεδία. Επέρχεται τελικά η συσσώρευση μεγάλου όγκου πληροφοριακού υλικού, που καθιστά δύσκολη τη διαχείριση και την ταξινόμηση του. Οι αδυναμίες αυτές εστιάζονται κυρίως στις δυνατότητες των εφαρμογών που χρησιμοποιούν για τη διαχείριση της πληροφορίας τους. Το υψηλό κόστος που απαιτείται για την υλοποίηση και συντήρησή τους (λόγω του εξειδικευμένου προσωπικού που απαιτούν εξαιτίας της πολυπλοκότητάς τους), ο μεγάλος χρόνος ανάπτυξης που χρειάζεται, αλλά κυρίως οι περιορισμένες δυνατότητες που τελικά αυτές μπορεί να προσφέρουν (κυρίως λόγω του πεπερασμένου ορίζοντα βάσει του οποίου αρχικά δημιουργούνται), τις καθιστούν συχνά περιοριστικές για την ανάπτυξη της επιχείρησης. Η διαχείριση περιεχομένου (content management) αναμένεται να οδηγήσει στη διαχείριση γνώσης (knowledge management). Έτσι η διαχείριση περιεχόμενου – μέσω ενός αυτοματοποιημένου εργαλείου, φτιαγμένου για αυτό το σκοπό – είναι απαραίτητη στις περιπτώσεις που υπάρχει μεγάλη ποσότητα πληροφορίας, μεταβαλλόμενου περιεχομένου (που καλείται να αλλάζει δυναμικά), από ποικίλες πηγές, η οποία απευθύνεται σε πολλούς αποδέκτες. Σήμερα η χρήση των Συστημάτων Διαχείρισης Περιεχομένου είναι ιδιαίτερα διαδεδομένη και αρκετά εξελιγμένη. Είτε αυτή έχει να κάνει με συστήματα ανοιχτού κώδικα, είτε με κλειστού για αυτό και ολοένα και περισσότερες επιχειρήσεις στρέφονται προς τα εκεί.

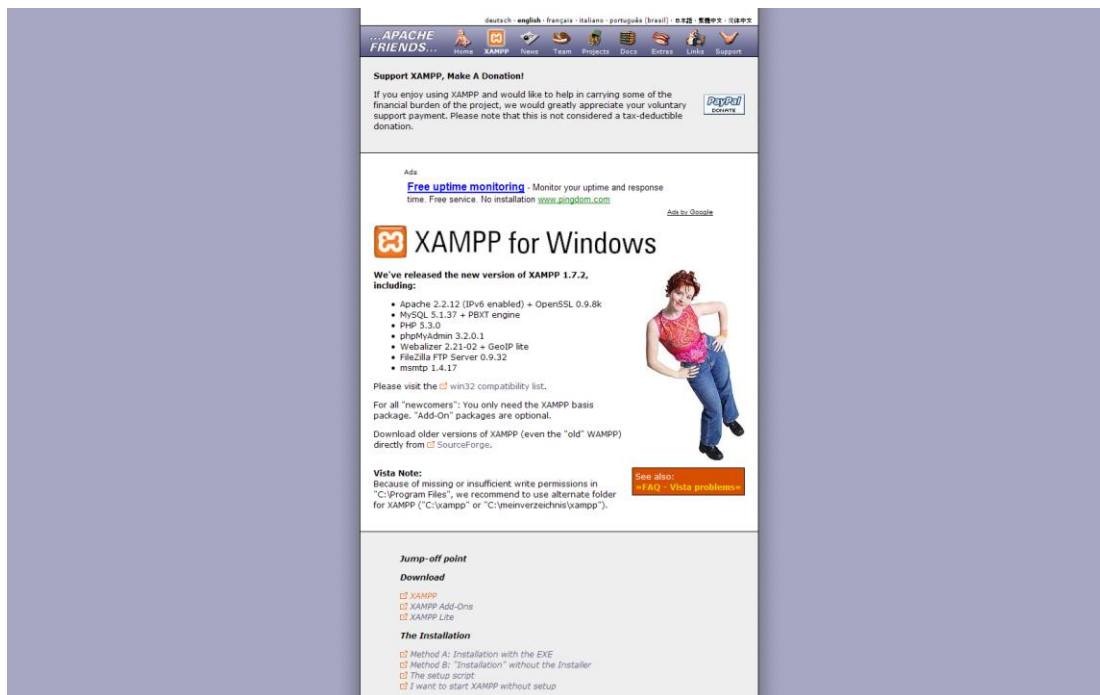
# XAMPP

## Εγκατάσταση του XAMPP

1. Πλοήγηση στον ιστότοπο του XAMPP στην διεύθυνση [www.nat32.com/XAMPP](http://www.nat32.com/XAMPP).



2. Διαλέγουμε από τα Downloads το XAMPP.



- Επιλέγουμε την υπερσύνδεση για κατέβαση του XAMPP για Windows σε μορφή «.exe».

**XAMPP**

You can download XAMPP for Windows in two different variations:

Self-extracting RAR archive:  
Easy and safe: XAMPP as self-extracting RAR archive with autostart of the setup script.

ZIP archive:  
For purists: XAMPP as ordinary ZIP archive.

**Attention:**  
If you extract the files, there can be false-positives virus warnings. [See also: FAQ - Virus warnings](#)

Version	Size	Content
<b>XAMPP Windows 1.7.2</b> [Basic package]		Apache 2.2.12 (IPv6 enabled), MySQL 5.1.37 + PBKT engine, PHP 5.3.0, OpenSSL 0.9.8j, phpMyAdmin 2.2.0.1, XAMPP Control Panel 2.5.8, XAMPP CLI Bundle 1.6, Webalizer 2.21-02, Mercury Mail Transport System v4.62, memmp 1.4.17, FileZilla FTP Server 0.9.32, SQLite 2.8.17, SQLite 3.6.16, ADOdb 3.09a, accelerator 0.6-rc1, libpng 2.0.2, Ming 0.42 For Windows 2000, 2003, XP, VISTA. See <a href="#">README</a>
<a href="#">EXE</a>	44 MB	Self-extracting RAR archive MDS (checksum) 14c2a3a35f752b9aee674548472348b
<a href="#">ZIP</a>	84 MB	Gewöhnliches ZIP archive MDS (checksum) 02c0c8a935d99087e0bac3ae7d
<b>XAMPP Windows 1.7.2</b> [Update package]		Not available yet

**XAMPP Add-Ons**

The following packages are extensions (add-ons) for the above XAMPP package. You don't need them for normal work.

Version	Size	Content
<b>Tomcat 6.0.20</b> 1.7.2		Tomcat 6.0.20 and mod_jk 1.2.20 for the XAMPP base package 1.7.2 For Windows 2000, 2003, XP, VISTA. See <a href="#">README</a>
<a href="#">EXE</a>	30 MB	Self-extracting RAR archive MDS (checksum) 6f9e150e436274b995980d811e013bc
<a href="#">ZIP</a>	42 MB	ZIP archive MDS (checksum) 4c37b5fe3d3e4e2916418246619a1194
<b>Perl Add-On</b>		Since XAMPP 1.7.2 part of the basic package.

**XAMPP Lite**

"XAMPP Lite" is an additional extra small XAMPP edition. It's the successor of the old MiniXAMPP. But in contrast to the big XAMPP it's not updated really often.

Version	Size	Content
<b>XAMPP Lite 1.7.2</b>		Apache 2.2.12 (IPv6 enabled), MySQL 5.1.37 + PBKT engine, PHP 5.3.0, OpenSSL 0.9.8j, phpMyAdmin 2.2.0.1, XAMPP Control Panel 2.5.8, XAMPP CLI Bundle 1.6, Webalizer 2.21-02, memmp 1.4.17, SQLite 2.8.17, SQLite 3.6.16, Ming 0.42

Αφού μας προωθήσει αυτή η υπερσύνδεση στο sourceforge.net όπου είναι αποθηκευμένο το αρχείο, θα μας εμφανιστεί η επιλογή για αποθήκευση.

**sourceforge** FIND AND DEVELOP OPEN SOURCE SOFTWARE

Welcome, Guest | [Log In](#) | [Create Account](#)

[Find Software](#) | [Develop](#) | [Create Project](#) | [Community](#) | [Site Support](#) | [About](#)

Search:  [Search](#)

Your XAMPP download will start shortly...

Rate this project: [1](#) [2](#) [3](#) [4](#) [5](#) [Share](#) [Subscribe](#) [Donate](#)

Problems with the download? Please use this direct link or try another mirror.

from our advertisers

[peer1](#) [Free Development Platform](#) [Provision Drupal, Joomla, WP, etc. Build.](#) [SQL Powers Web 2.0](#)

Sponsor Offer: [HP ProLiant G6 Servers](#) Perform like a accountant.

Ads by Google

Mirror provided by: [THE SUNET ARCHIVE](#)

**About SourceForge**

- About SourceForge
- Contact Us
- Jobs @ Geeknet
- Advertising

**Find Software**

- Search Software
- Browse by Category
- Most Popular Overall
- Most Active Overall

**Develop Software**

- Create Project
- Project Help Wanted
- New Projects

**Community**

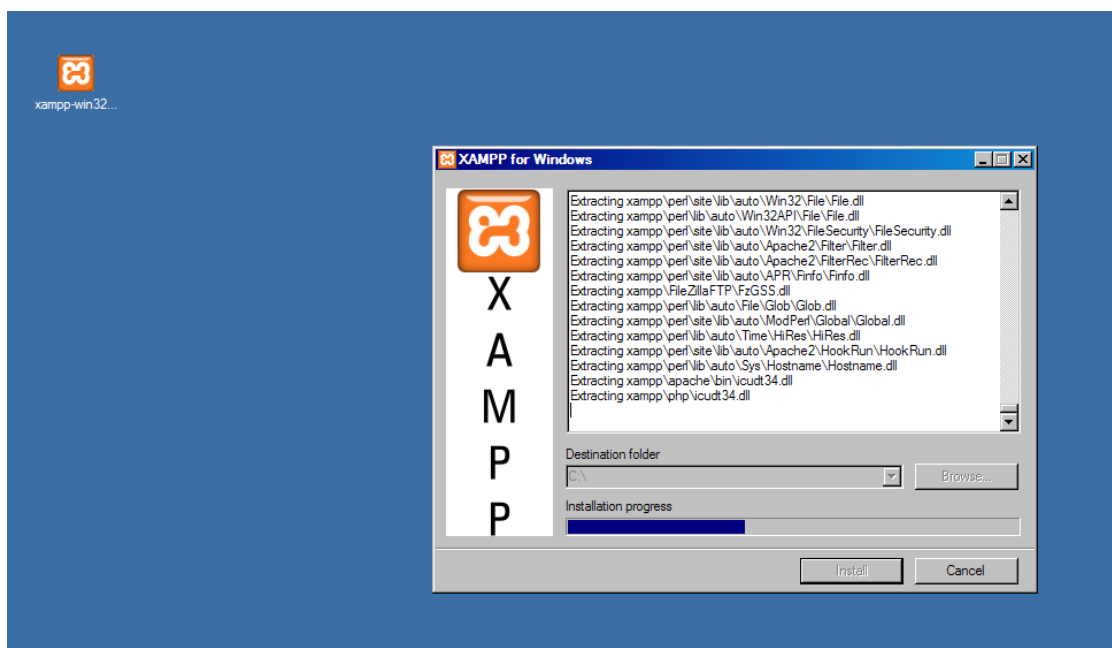
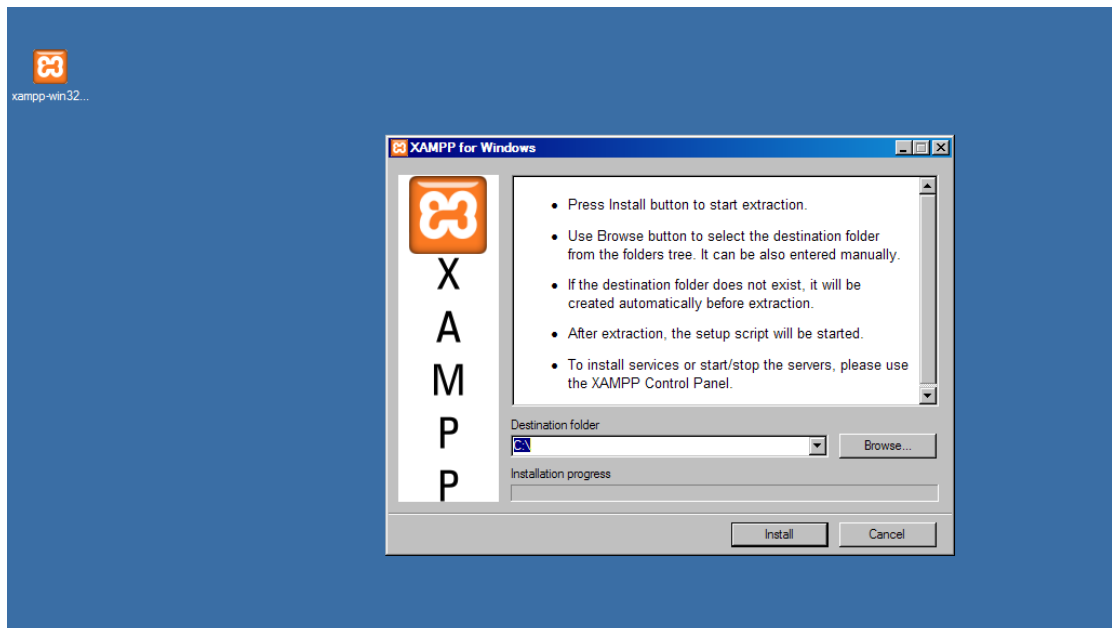
- SourceForge Forums
- The Blog
- @sourceforge on Twitter
- Calendar

**Help**

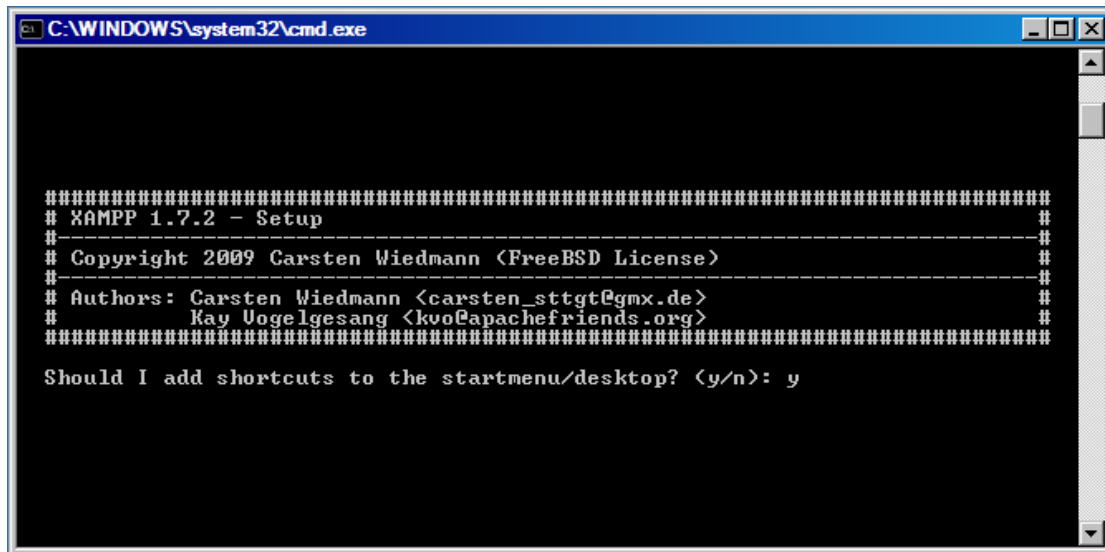
- Site Documentation
- Submit Support Requests
- Site Outages
- Feedback

Copyright © 2009 Geeknet, Inc. All rights reserved. [Terms of Use](#)

4. Εκτελούμε το αρχείο που κατεβάσαμε και συμπληρώνουμε την τοποθεσία εγκατάστασης.



5. Πατάμε το «y» για να μας δημιουργήσει μια συντόμευση στην επιφάνεια εργασίας.

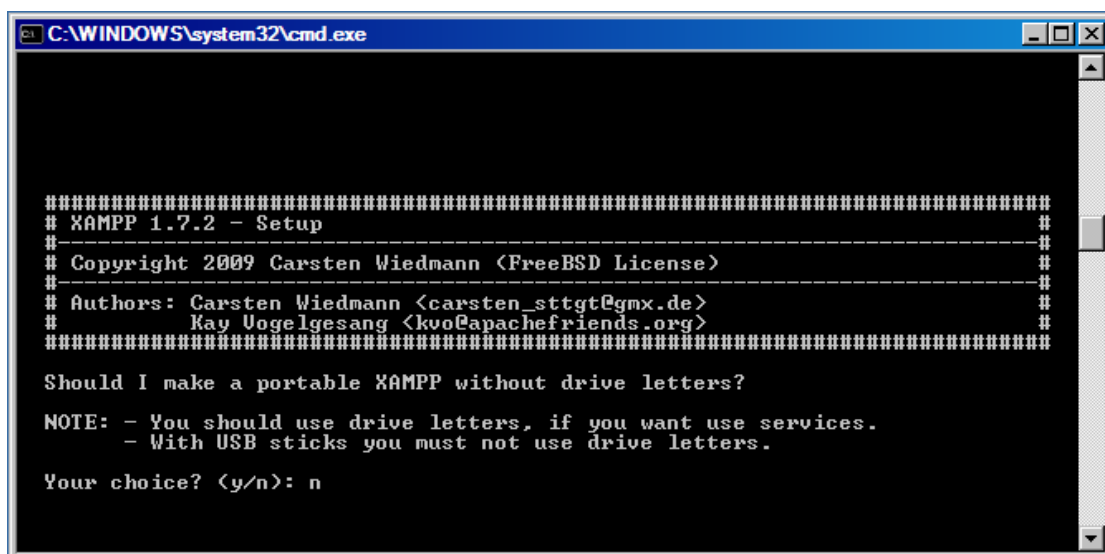


```
C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
#          Kay Vogelgesang <kvo@apachefriends.org>
#####

Should I add shortcuts to the startmenu/desktop? (y/n): y
```

6. Πατάμε το «n» ώστε να μην μας εγκαταστήσει το XAMPP (portable) σε usb stick.



```
C:\WINDOWS\system32\cmd.exe

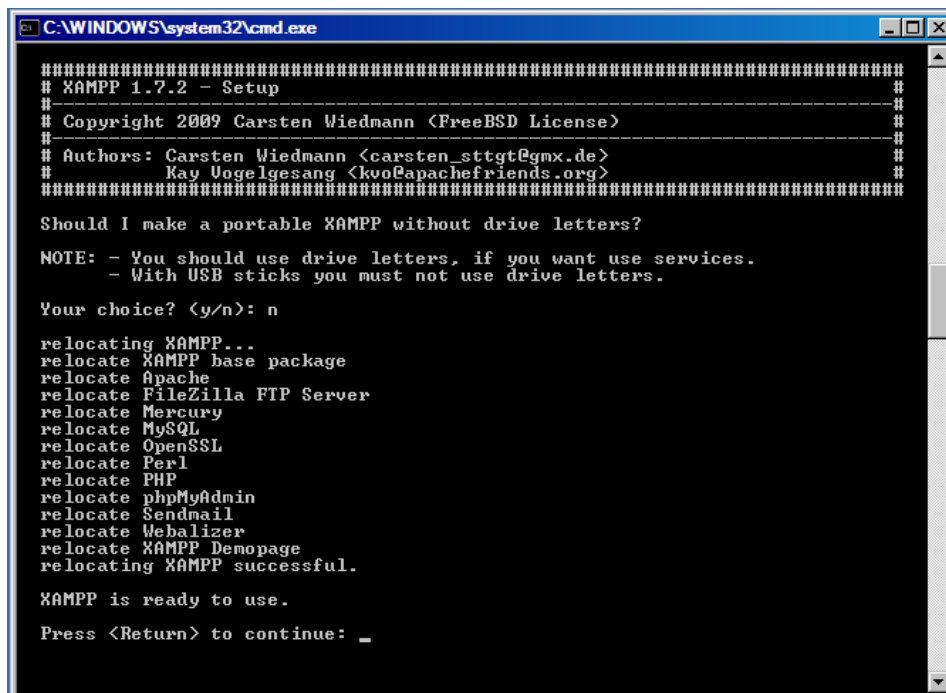
#####
# XAMPP 1.7.2 - Setup
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
#          Kay Vogelgesang <kvo@apachefriends.org>
#####

Should I make a portable XAMPP without drive letters?

NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): n
```

7. Πατάμε «Enter» ή «Return» για να συνεχιστεί η παραμετροποίηση του XAMPP.



```
C:\WINDOWS\system32\cmd.exe
#####
# XAMPP 1.7.2 - Setup
#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
#          Kay Vogelgesang <kvo@apachefriends.org>
#####

Should I make a portable XAMPP without drive letters?

NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

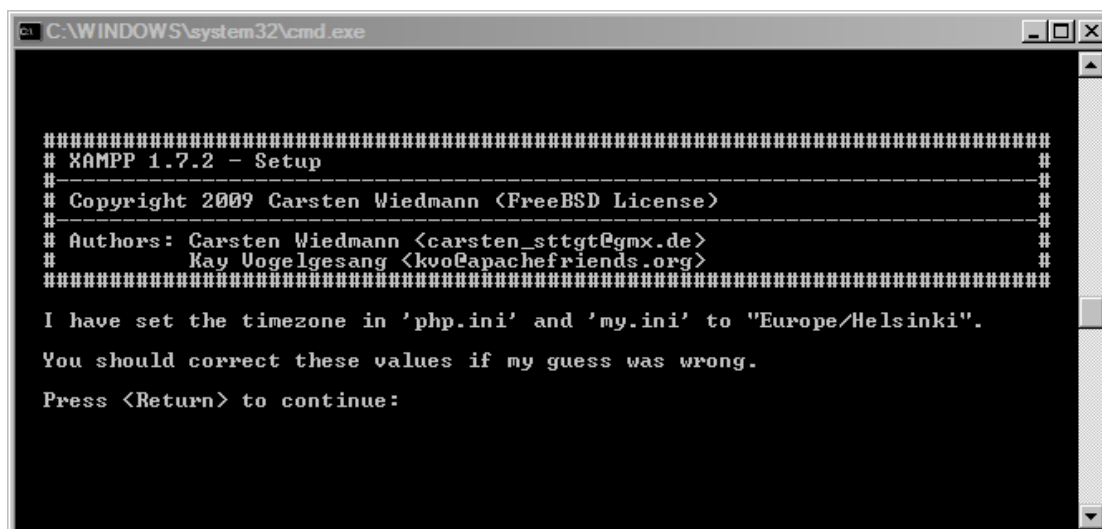
Your choice? <y/n>: n

relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.

XAMPP is ready to use.

Press <Return> to continue: _
```

Εδώ το XAMPP έθεσε την χρονική ζώνη (timezone) στα αρχεία «php.ini» και «my.ini».



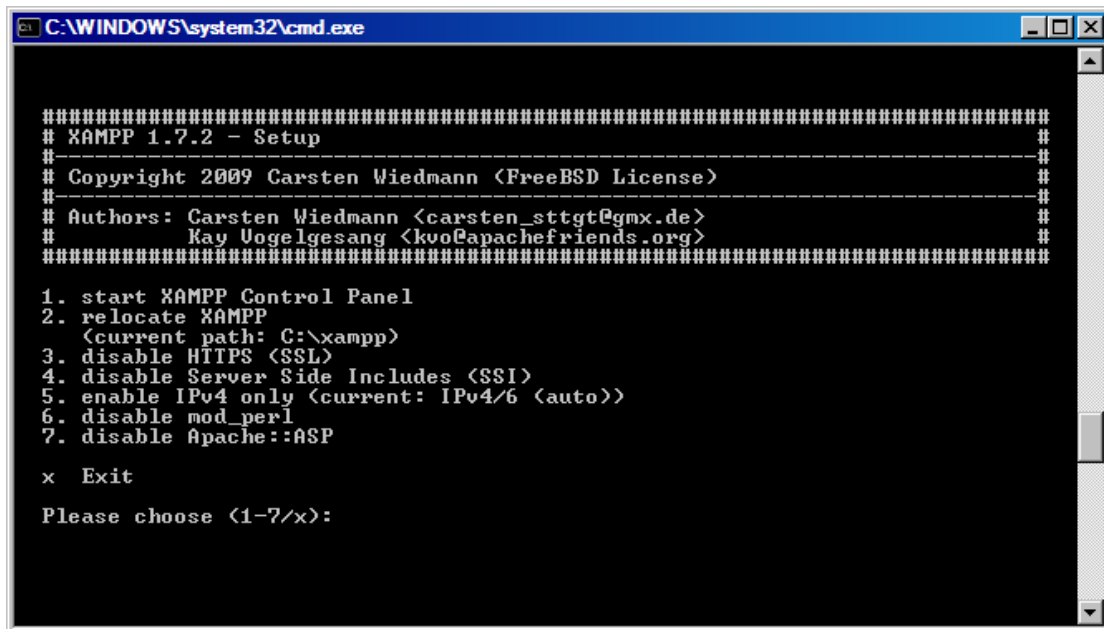
```
C:\WINDOWS\system32\cmd.exe
#####
# XAMPP 1.7.2 - Setup
#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
#          Kay Vogelgesang <kvo@apachefriends.org>
#####

I have set the timezone in 'php.ini' and 'my.ini' to "Europe/Helsinki".

You should correct these values if my guess was wrong.

Press <Return> to continue:
```

8. Το XAMPP έχει ολοκληρώσει τη παραμετροποίηση/εγκατάσταση και μας ζητάει μέσω ενός μενού να ενεργοποιήσουμε/απενεργοποιήσουμε περεταίρω ρυθμίσεις.



```
C:\WINDOWS\system32\cmd.exe

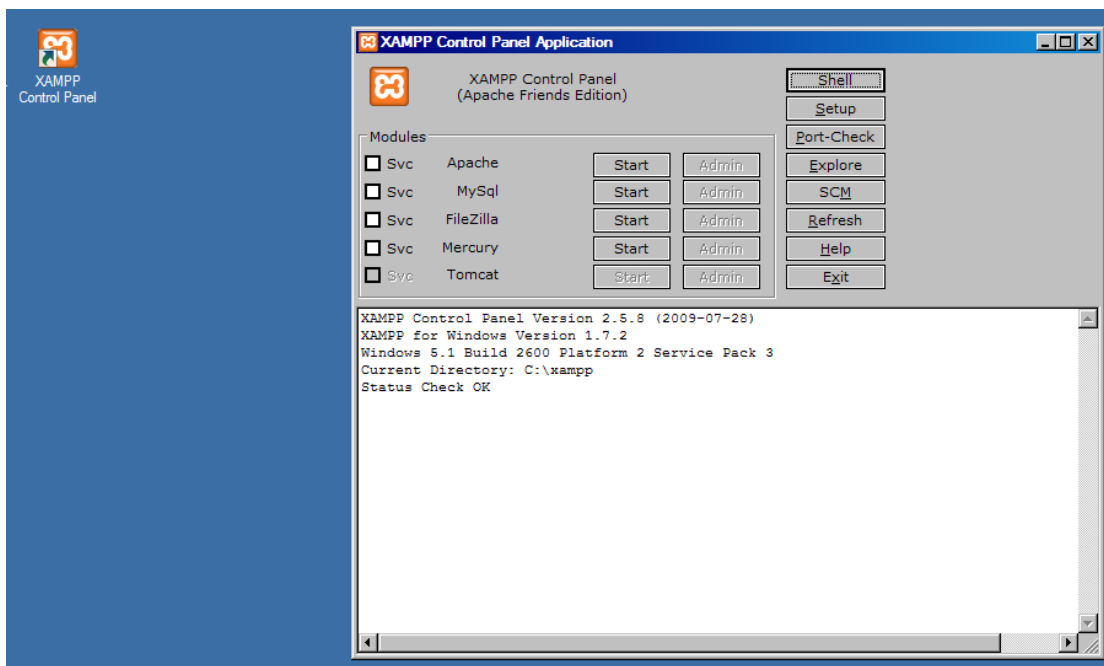
#####
# XAMPP 1.7.2 - Setup
#-----
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
#-----
# Authors: Carsten Wiedmann <carsten_stt@tgm.de>
#          Kay Vogelgesang <kvo@apachefriends.org>
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP

x Exit

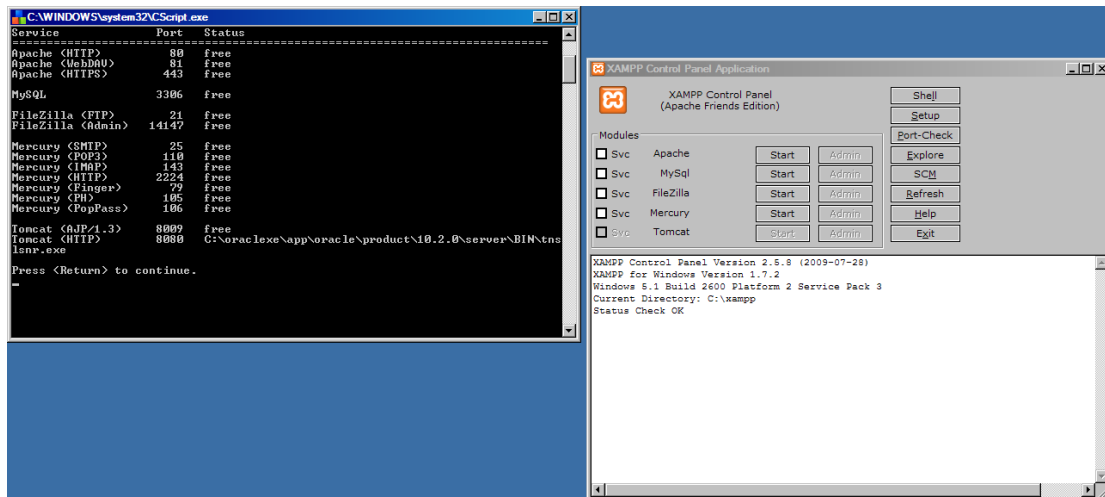
Please choose (1-7/x):
```

9. Τέλος μας εμφανίζεται το control panel του XAMPP με τις επιλογές εκκίνησης του Apache, MySQL, FileZilla, Mercury και του Tomcat εφόσον έχουμε εγκαταστήσει το αντίστοιχο add-on από την ιστοσελίδα του XAMPP (προαιρετικό).

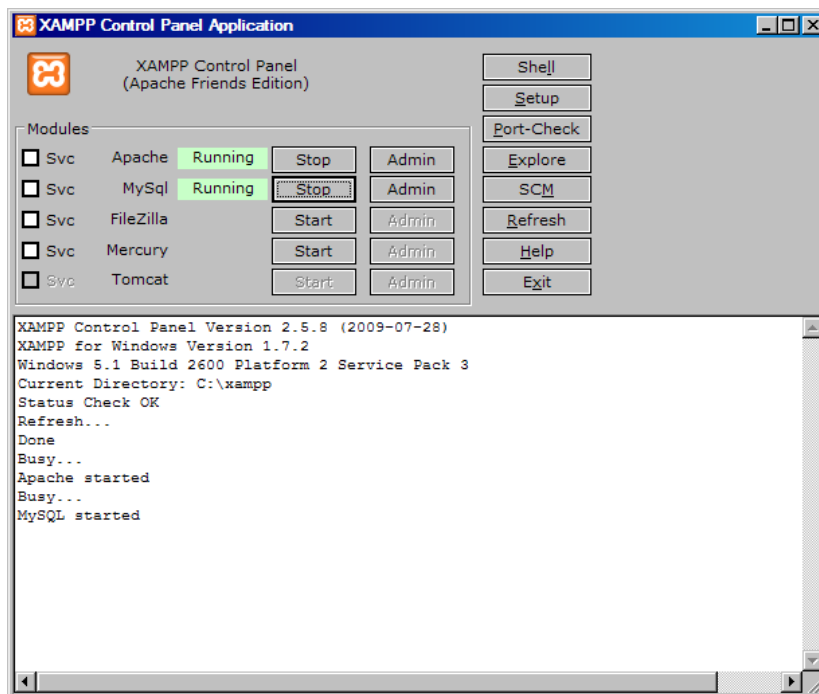




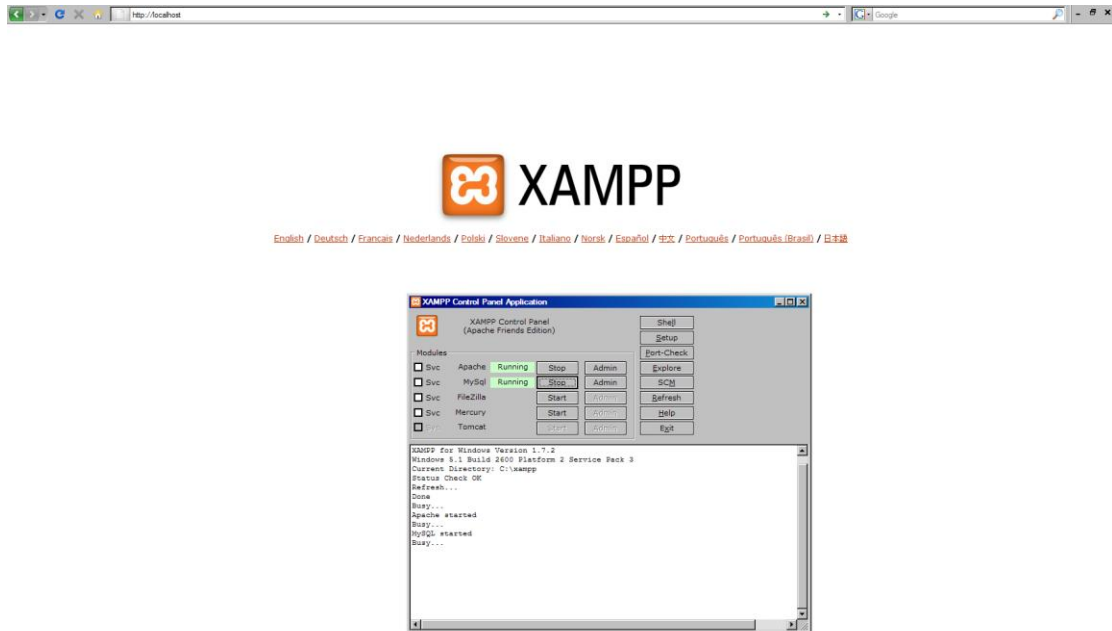
10. Πατάμε το Port-Check από το control panel για να γίνει έλεγχος των port που θα χρησιμοποιηθούν από τα προγράμματα του XAMPP για τυχόν συγκρούσεις των port. Κλασικό παράδειγμα είναι το πρόγραμμα Skype το οποίο χρησιμοποιεί τα ίδια port με τον Apache (http/https, 80/443) και η Oracle Express Edition που χρησιμοποιεί το ίδιο με τον Tomcat (8080).



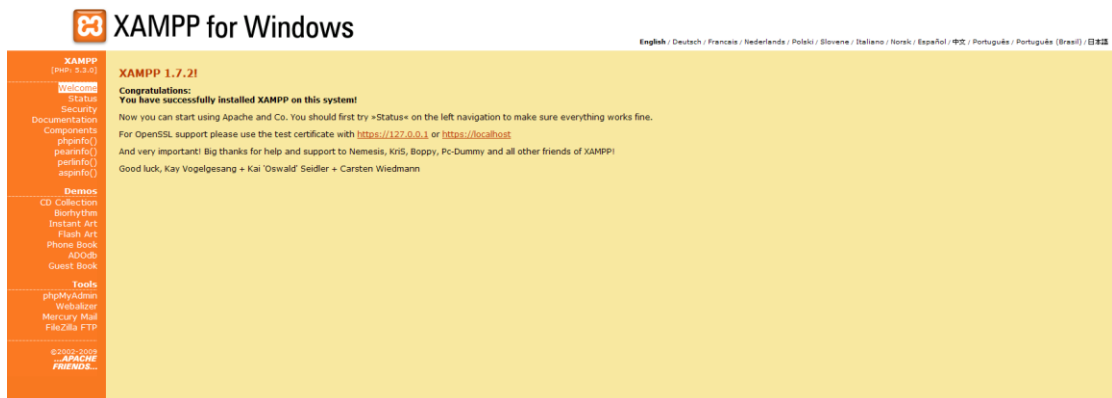
11. Πατάμε Start στο Control Panel του XAMPP για να κάνουμε εκκίνηση του Apache web server και της μηχανής MySQL (απαραίτητα στοιχεία για τοπική εγκατάσταση και διαχείριση ιστοτόπων μέσω CMS).



12. Για να ελέγξουμε την σωστή εγκατάσταση και λειτουργία του εικονικού μας εξυπηρετητή, ανοίγουμε τον φυλλομετρητή μας (π.χ. Firefox) και πηγαίνουμε στην διεύθυνση **http://localhost** ή στην loopback διεύθυνση **http://127.0.0.1**.



Έχοντας επιλέξει την γλώσσα που θέλουμε μεταφερόμαστε στην κύρια σελίδα ελέγχου του XAMPP. Από εδώ μπορούμε να μεταβούμε σε διάφορες σημαντικές λειτουργίες και εργαλεία του διακομιστή μας (π.χ. phpMyAdmin, Webalizer, FileZilla FTP, Status, Security).



13. Καλή πρακτική είναι πριν συνεχίσουμε με την εγκατάσταση του CMS, να ελέγξουμε με την διαδικασία **phpinfo()** την έκδοση και άλλες σημαντικές πληροφορίες της PHP που έχουμε εγκατεστημένη. Αυτό γίνεται για λόγους διαπίστωσης της συμβατότητας του διακομιστή μας με το CMS που θα χρησιμοποιήσουμε.

The screenshot shows the XAMPP for Windows interface on the left and the output of the `phpinfo()` function in a browser window on the right. The browser window title is "PHP Version 5.3.0". The output is a table with the following key information:

System	Windows NT SMOKEMACHINE 5.1 build 2600 (Windows XP Professional Service Pack 3) i686
Build Date	Jul 2 2009 21:02:53
Compiler	MSVC9 (Visual C++ 9.0)
Architecture	32
Configure Command	cmd.exe /c:php.exe && configure.js
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,TS,VC6
PHP Extension Build	API20090626,TS,VC6
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	ftp, ftps, php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, ssl, ssh2, ftps
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

Below the table, it says "This program makes use of the Zend Scripting Language Engine: Zend Engine v2.3.0, Copyright (c) 1998-2009 Zend Technologies" and "Powered By Zend Engine 2". At the bottom, there are links for "PHP Credits" and "Configuration".

Τέλος πατάμε το phpMyAdmin στα εργαλεία που μας προσφέρει το XAMPP ή μεταβαίνουμε στην διεύθυνση <http://localhost/phpMyAdmin> για να ελέγξουμε αν έχει εγκατασταθεί σωστά και να συλλέξουμε σχετικές πληροφορίες με το σύστημα διαχείρισης βάσεων δεδομένων που έχουμε εγκατεστημένο (MySQL) έτσι ώστε να ξέρουμε αν είναι συμβατό με το CMS που θα χρησιμοποιήσουμε.

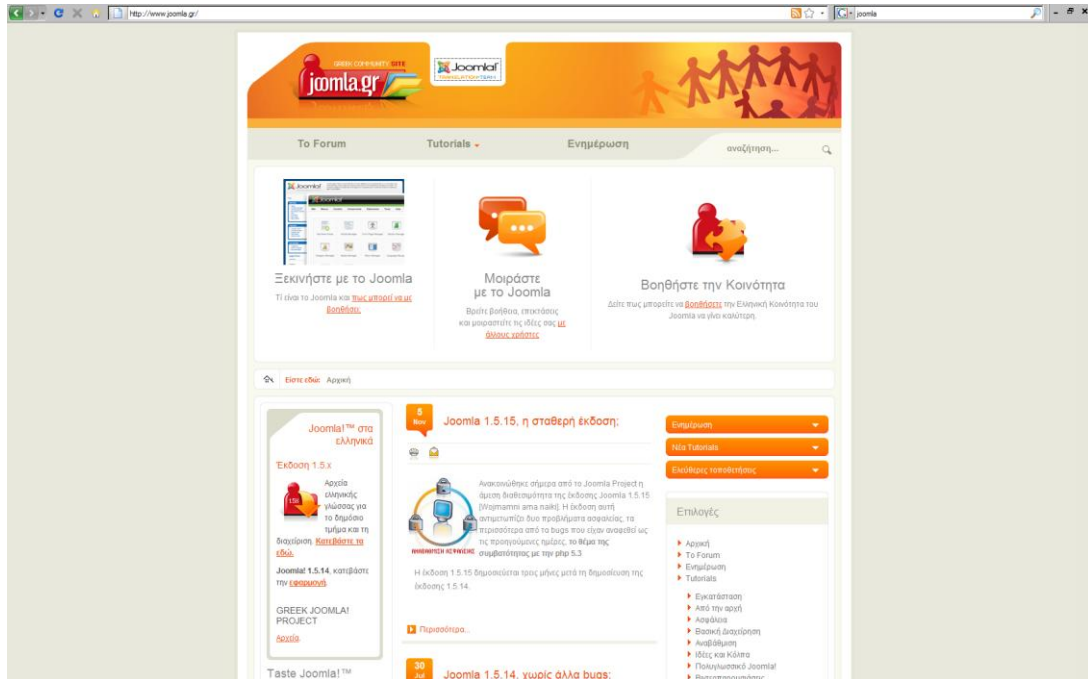
The screenshot shows the phpMyAdmin web interface in a browser window. The address bar shows `http://localhost/phpmyadmin/`. The interface is in English and shows the "MySQL localhost" configuration page. Key information includes:

- MySQL localhost:** Create new database (name: , collation: )
- Interface:** Language: English, Theme / Style: Original, Custom color: , Font size: 32%
- MySQL:** Server: localhost via TCP/IP, Server version: 5.1.37, Protocol version: 10, User: root@localhost, MySQL charset: UTF-8 Unicode (utf8)
- Web server:** Apache/2.2.12 (Win32) DAV/2 mod\_ssl/2.2.12 OpenSSL/0.9.8k mod\_autoindex\_color php/5.3.0 mod\_fastcgi/2.4.4 PHP/5.1.37, MySQL client version: 5.1.37, PHP extension: mysql
- phpMyAdmin:** Version information: 3.2.0-1, Documentation, Wiki, Official Homepage, [ChangeLog] [Subversion] [Lists]

# Joomla! CMS

## Εγκατάσταση του Joomla! CMS

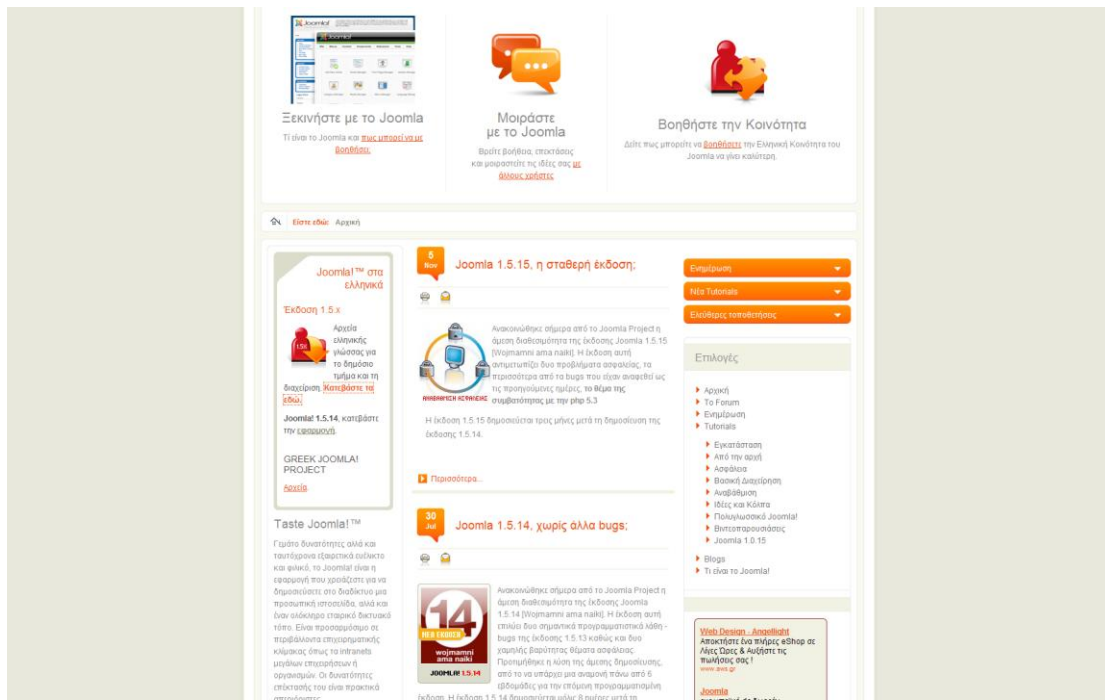
1. Πλοήγηση στον ιστότοπο της Ελληνικής κοινότητας του Joomla! στην διεύθυνση [www.joomla.gr](http://www.joomla.gr).



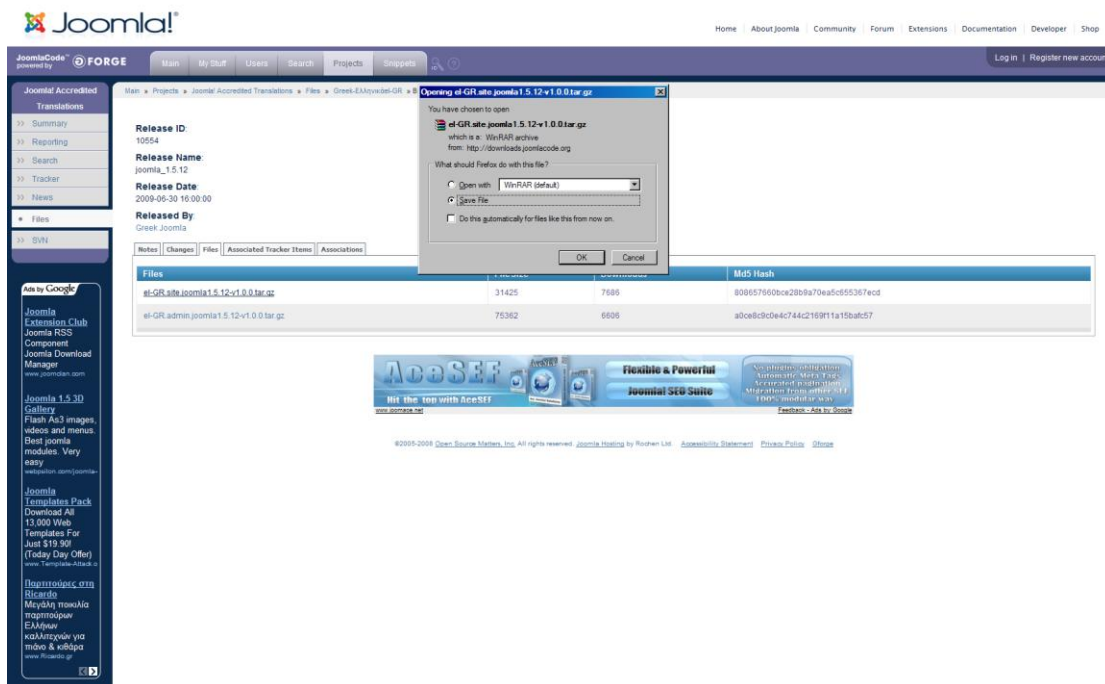
2. Πατάμε την υπερσύνδεση για κατέβασμα της τελευταίας σταθερής (stable) έκδοσης του Joomla! CMS.

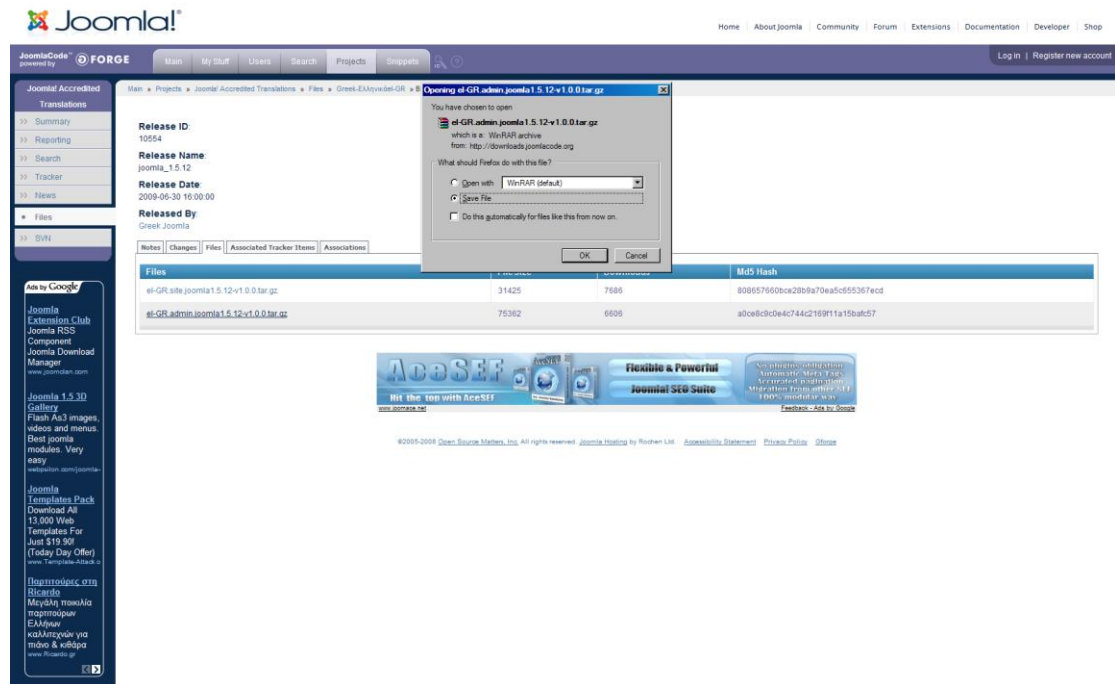




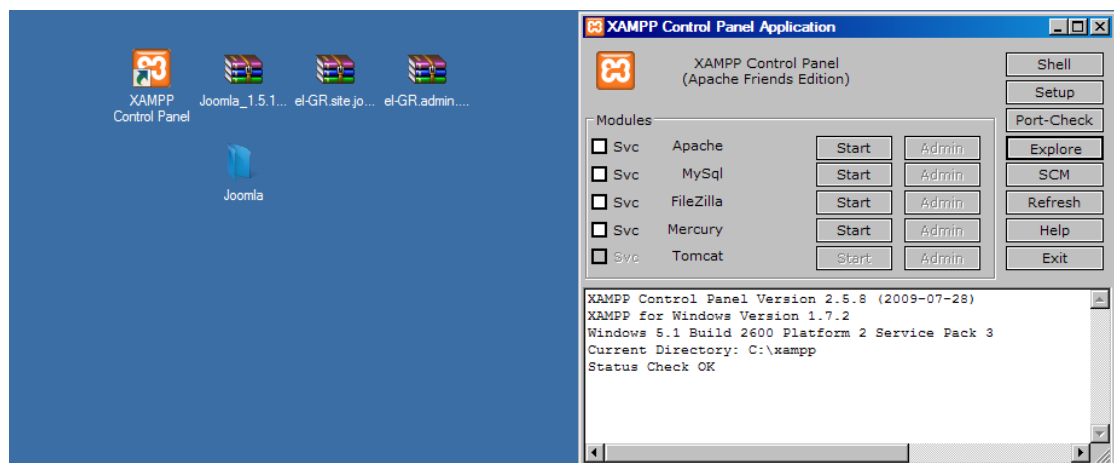


Επιλέγουμε τα συμπιεσμένα αρχεία ελληνικής γλώσσας για το δημόσιο τμήμα (site) και το τμήμα διαχείρισης (admin).

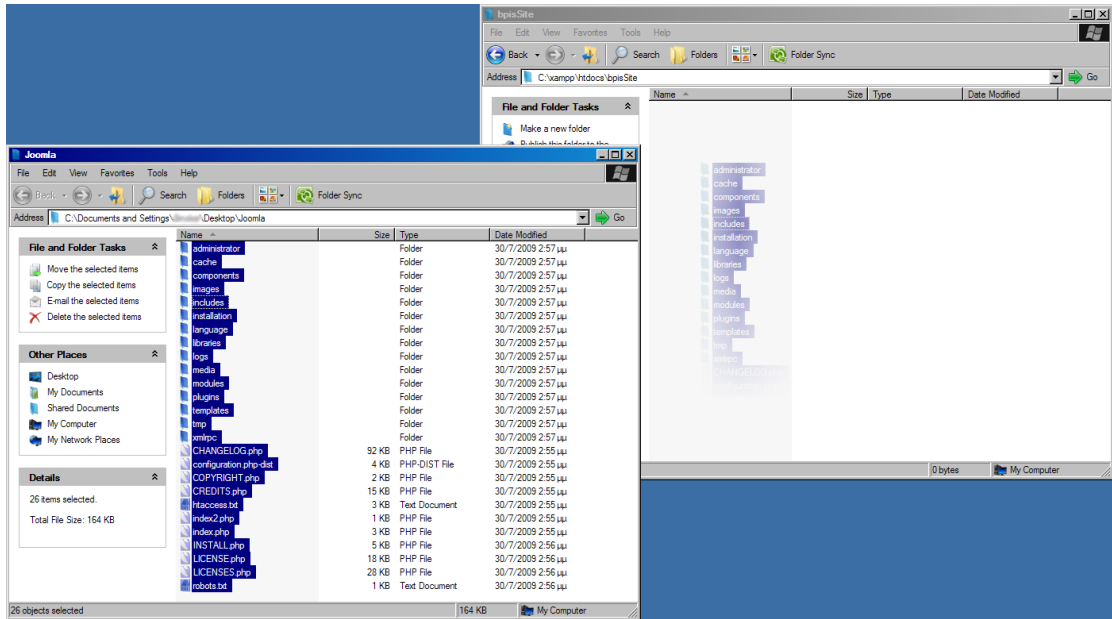




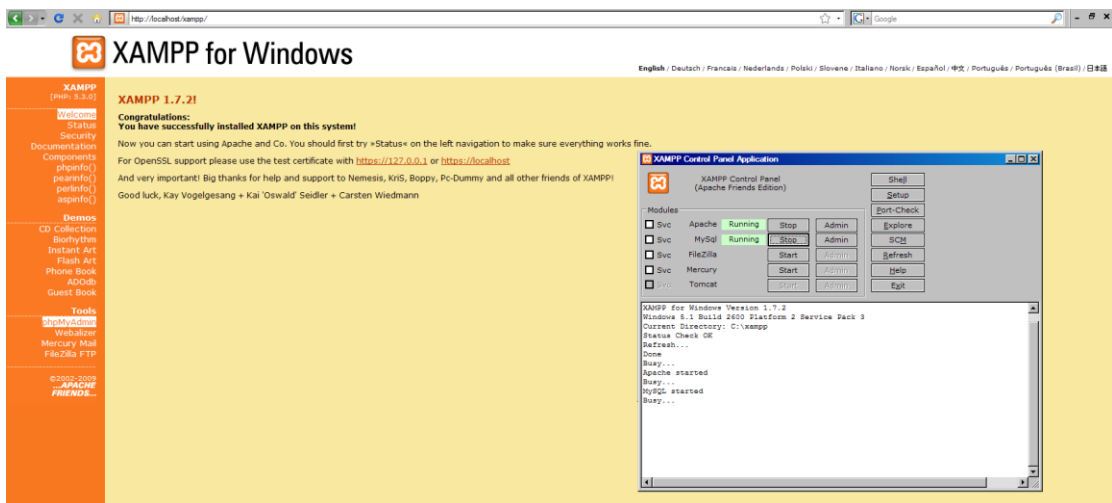
- Εξερευνούμε τα αρχεία του εξυπηρετητή μας πατώντας το κουμπί Explore από το Control panel του XAMPP.



- Μεταφέρουμε τον αποσυμπιεσμένο φάκελο του Joomla! μέσα στον φάκελο **htdocs** που βρίσκεται μέσα στην εγκατάσταση του XAMPP και τον μετονομάζουμε όπως θέλουμε (π.χ. bprisSite). Η συνήθης διεύθυνση είναι `c:\xampp\htdocs\όνομα_του_site_μας\`.

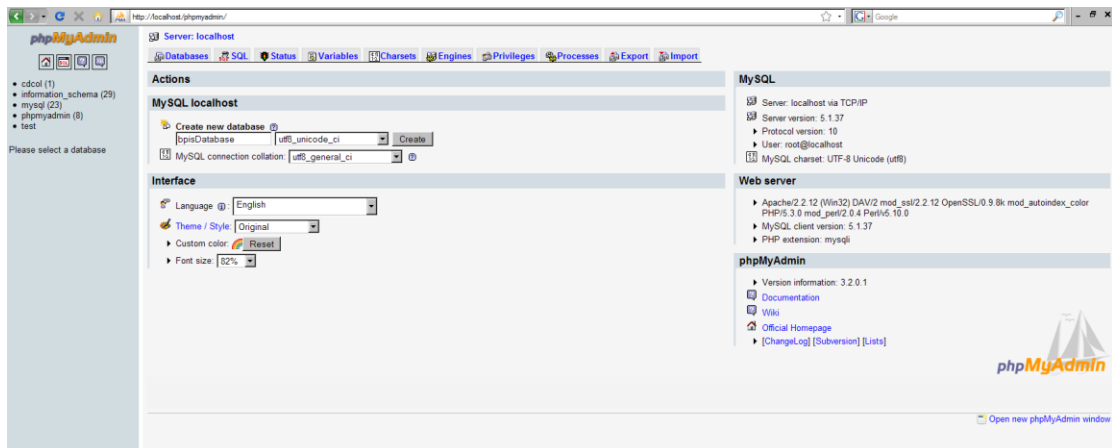


7. Έχοντας κάνει εκκίνηση του εικονικού μας εξυπηρετητή και των κύριων στοιχείων του, πλοηγούμαστε με τον φυλλομετρητή μας στο κύριο μενού του XAMPP.

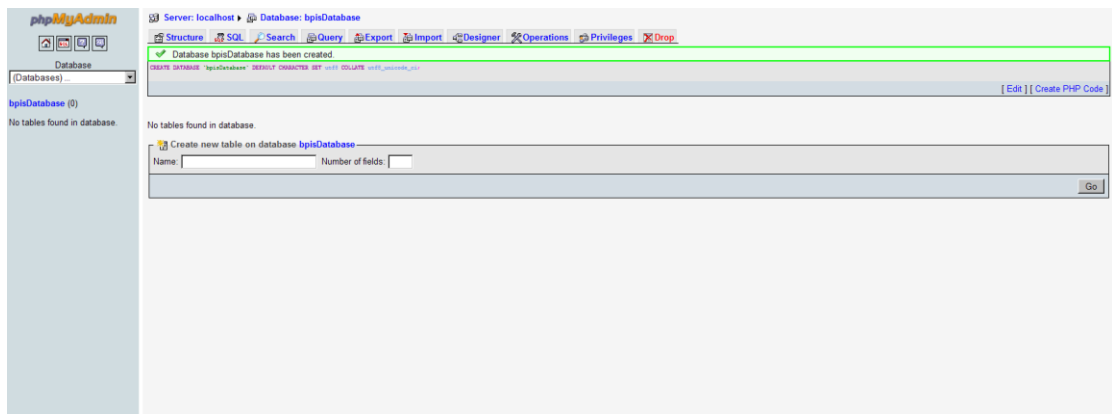


8. Έπειτα μεταβαίνουμε στο εργαλείο phpMyAdmin. Συμπληρώνουμε στο πεδίο **Create new database** το όνομα της βάσης μας (π.χ. bprisDatabase) και μετά επιλέγουμε κωδικοποίηση **utf8\_unicode\_ci** και στο πεδίο **MySQL connection collation** επιλέγουμε την ίδια κωδικοποίηση. Τέλος πατάμε **Create** για να δημιουργηθεί η βάση δεδομένων που θα χρησιμοποιήσει το CMS μας.

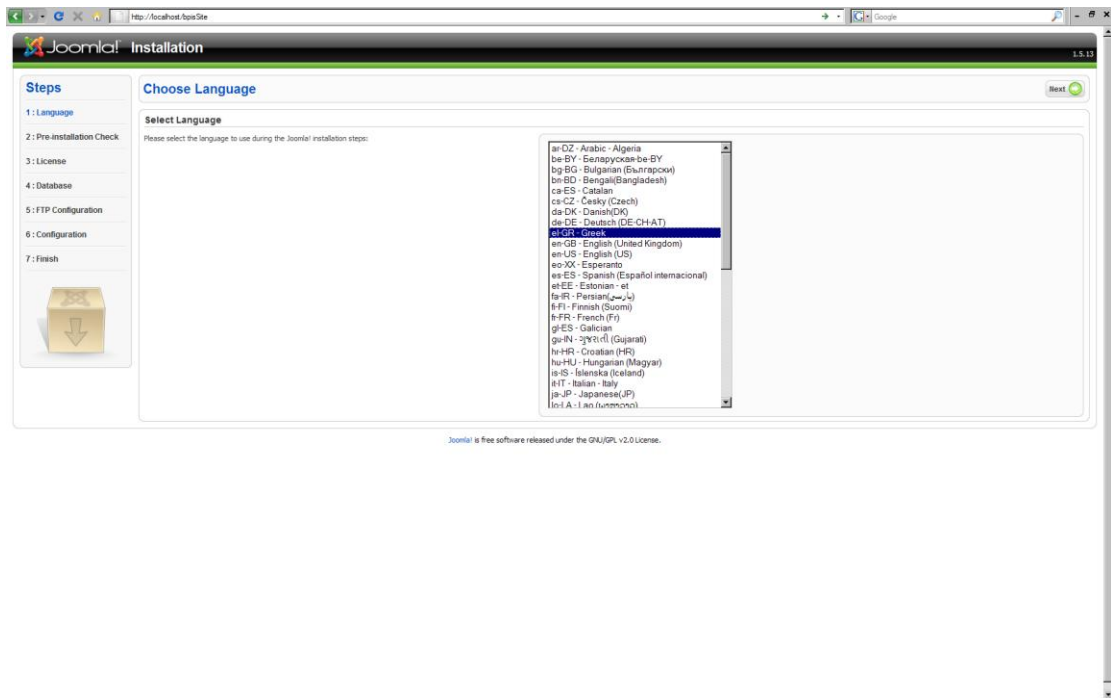




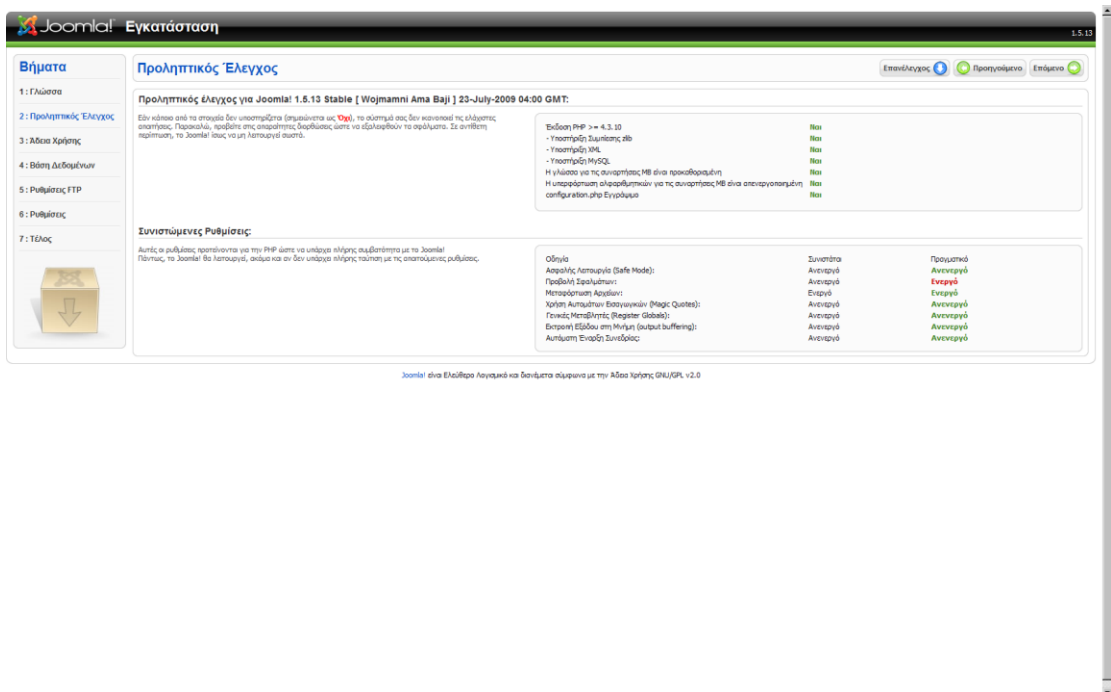
Παρατηρούμε τα μηνύματα του εργαλείου για την επιτυχή ή αποτυχημένη δημιουργία της βάσης δεδομένων.



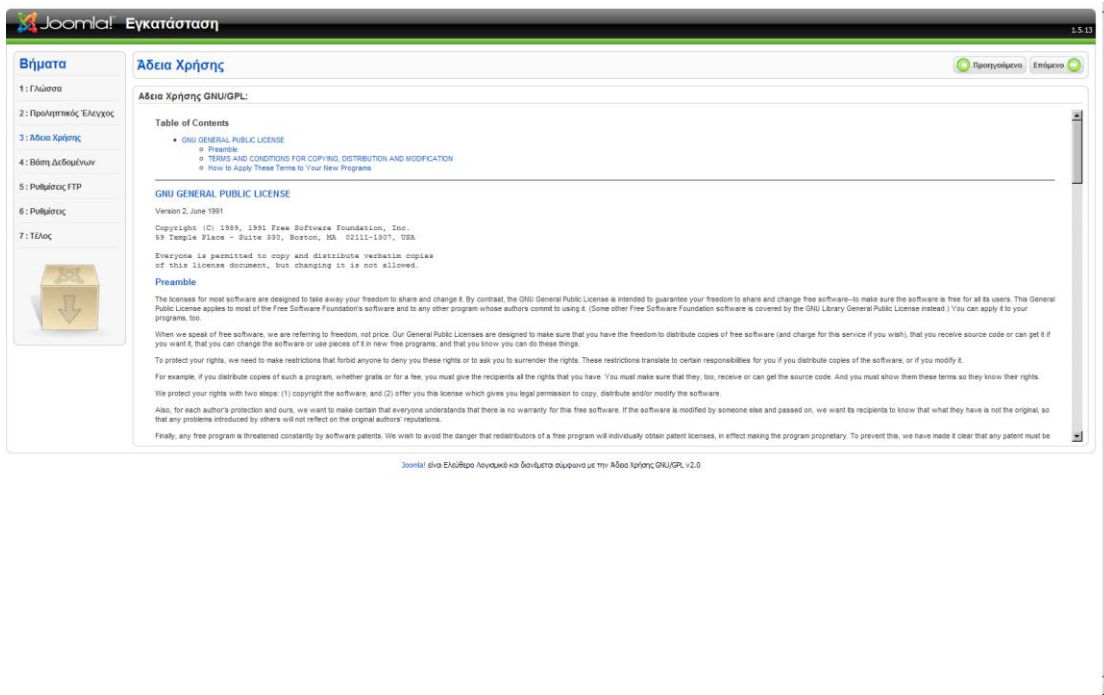
9. Πλοηγούμαστε στην διεύθυνση [http://localhost/όνομα\\_του\\_site\\_μας](http://localhost/όνομα_του_site_μας) . (π.χ. <http://localhost/bpisSite>) για να ξεκινήσει η διαδικασία εγκατάστασης του Joomla! CMS. Το πρώτο βήμα μας επιτρέπει να επιλέξουμε την γλώσσα της εγκατάστασης.



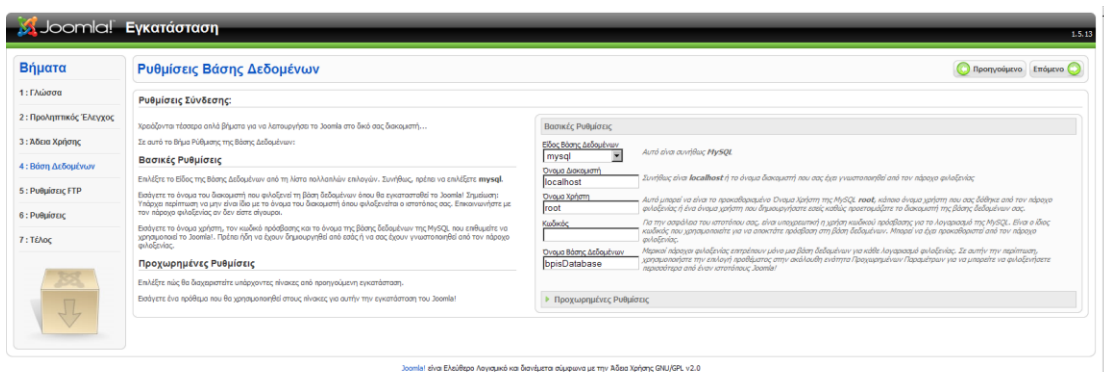
10. Σε αυτό το βήμα το Joomla! CMS μας προβάλλει έναν προληπτικό έλεγχο των ελάχιστων απαιτήσεων και ποιες συνιστώμενες ρυθμίσεις είναι ενεργοποιημένες/απενεργοποιημένες. Για να συνεχίσουμε στο επόμενο βήμα θα πρέπει να ελέγξουμε ότι είναι σημειωμένο με κόκκινο, ειδικά στις ελάχιστες απαιτήσεις.



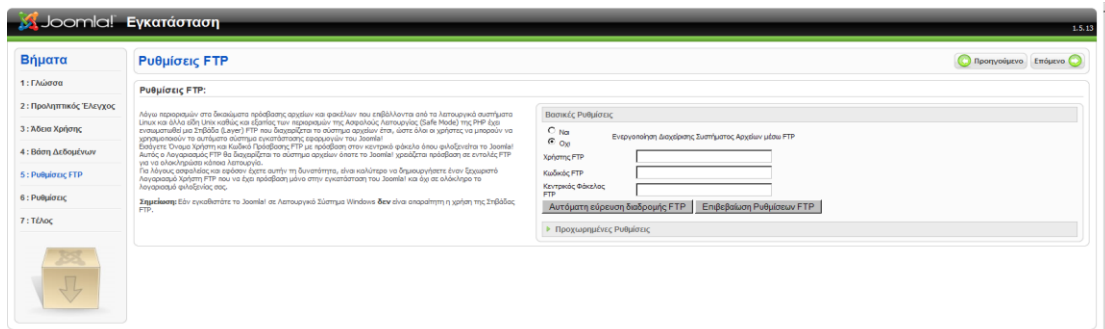
11. Το Joomla! CMS σε αυτό το βήμα μας προβάλλει την άδεια χρήσης του την οποία και θα πρέπει να διαβάσουμε προσεκτικά ώστε να αποφευχθούν τυχόν νομικά ζητήματα κατά την χρήση του.



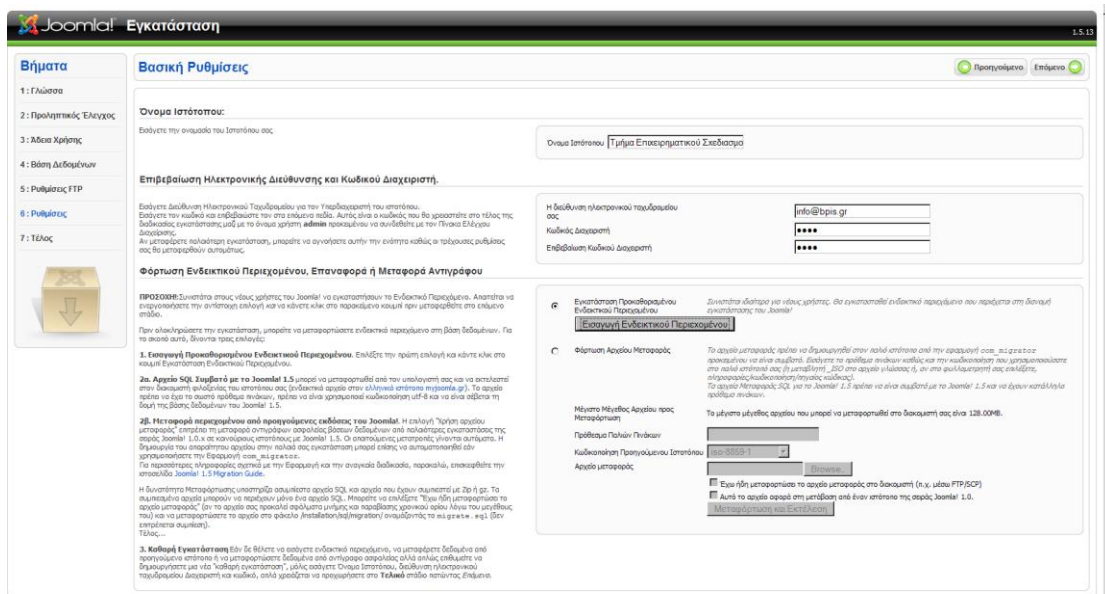
12. Εδώ θα ρυθμίσουμε το CMS κατάλληλα ώστε να μπορέσει να χρησιμοποιήσει την βάση δεδομένων. Στο πεδίο *είδος βάσης δεδομένων* διαλέγουμε **MySQL**, στο πεδίο *Όνομα Διακομιστή* γράφουμε την διεύθυνση του εικονικού μας εξυπηρετητή δηλαδή **localhost**, στο *Όνομα Χρήστη* της βάσης δεδομένων θέτουμε **root** το οποίο είναι το αρχικό, στο πεδίο *κωδικός* το αφήνουμε κενό. Τέλος θέτουμε στο *Όνομα Βάσης Δεδομένων* το όνομα της βάσης μας (π.χ. **bpisDatabase**).



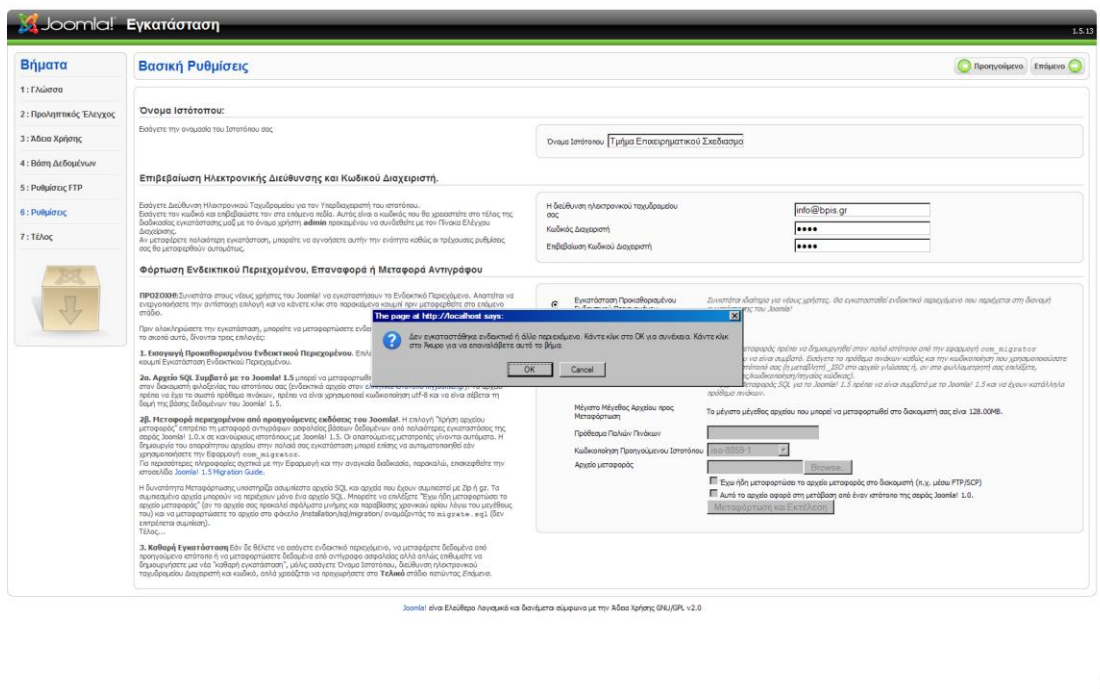
13. Σε αυτό το βήμα θα πρέπει να ρυθμίσουμε τη Στιβάδα FTP αλλά εφόσον εγκαταστήσαμε το Joomla! τοπικά (στον υπολογιστή μας) δεν είναι απαραίτητο για αυτό και επιλέγουμε όχι στις βασικές ρυθμίσεις.



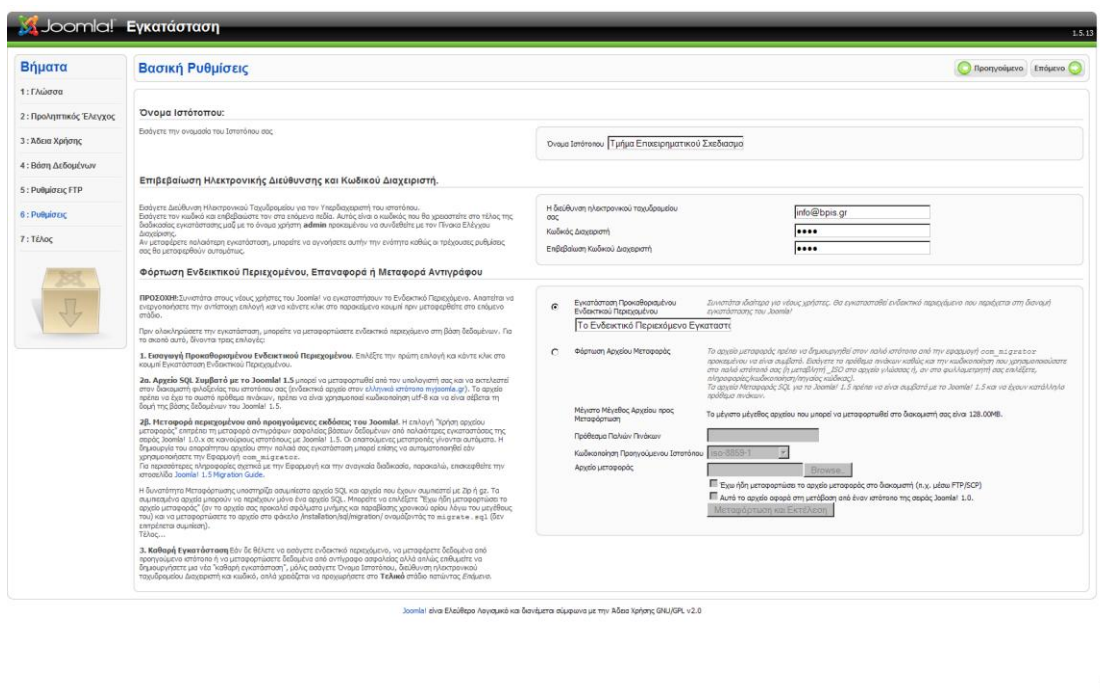
14. Εδώ θέτουμε στο Όνομα Ιστοτόπου το όνομα που επιθυμούμε να φαίνεται στο portal μας (π.χ. **Τμήμα ΕΣ&ΠΣ**), στο Η διεύθυνση ηλεκτρονικού ταχυδρομείου σας θέτουμε την ηλεκτρονική διεύθυνση του διαχειριστή του portal (π.χ. **info@bpris.gr**), στον Κωδικό Διαχειριστή θέτουμε τον κωδικό (π.χ. **admin**) που θα χρησιμοποιούμε για να εισέλθουμε στο διαχειριστικό κομμάτι του Joomla! CMS μαζί με όνομα χρήστη **admin** και επιβεβαιώνουμε στο παρακάτω πεδίο.



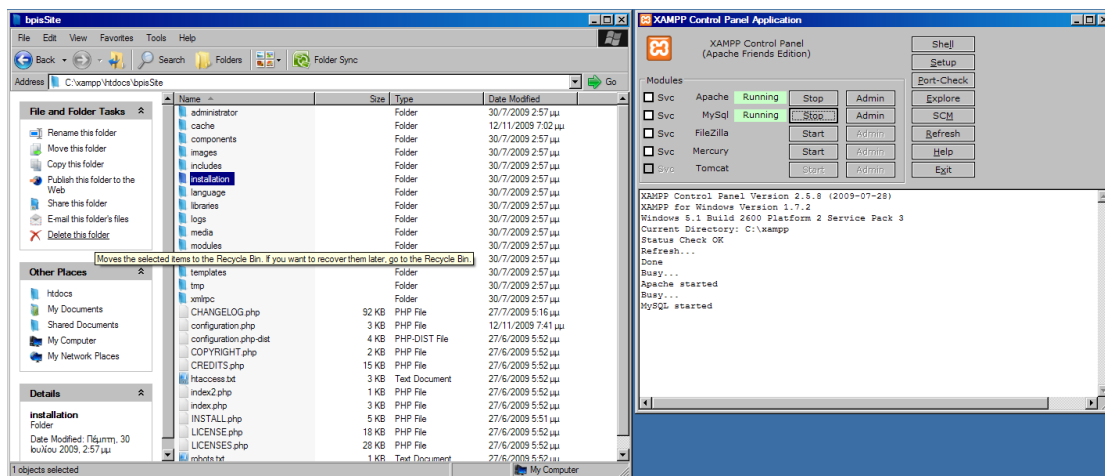
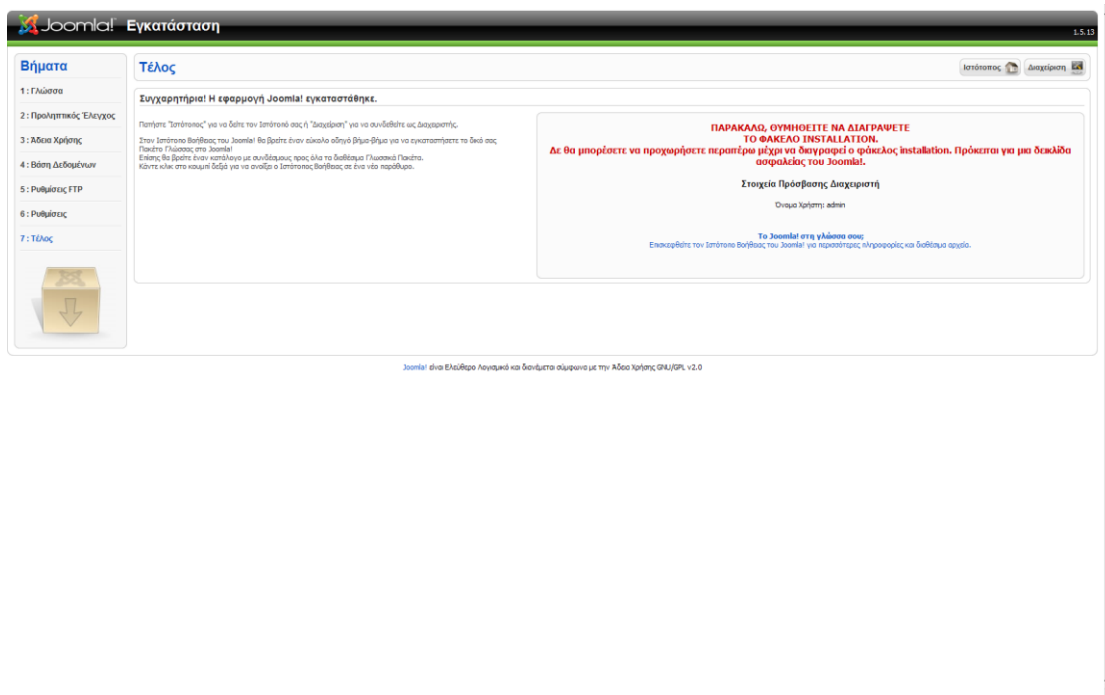
15. Πατώντας επόμενο το Joomla! CMS μας εμφανίζει ένα μήνυμα και μας υπενθυμίζει ότι δεν έχουμε εγκαταστήσει ενδεικτικό υλικό ή άλλο περιεχόμενο στον ιστότοπο μας. Πατάμε Άκυρο αν θέλουμε να επιστρέψουμε στην σελίδα που ήμασταν αλλιώς συνέχεια.



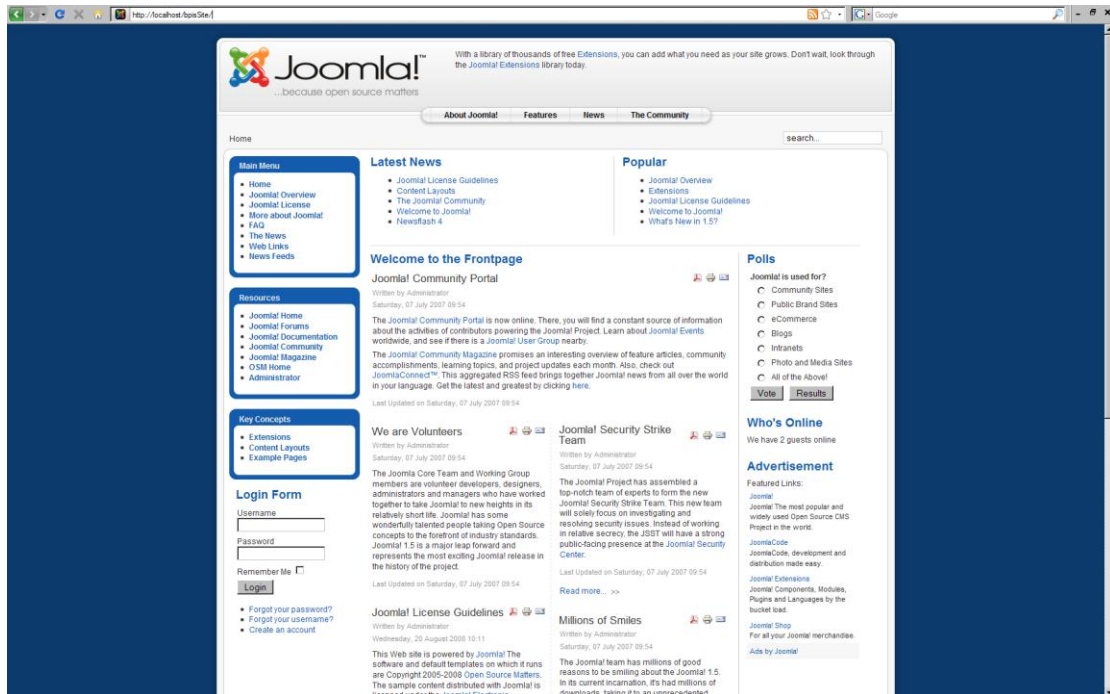
16. Εδώ έχουμε πατήσει το κουμπί *Εισαγωγή Ενδεικτικού Περιεχομένου* και μας ενημερώνει μετονομάζοντας το κουμπί σε *Το Ενδεικτικό Περιεχόμενο Εγκαταστάθηκε*. Πατάμε επόμενο.



17. Τέλος δεχόμαστε τα συγχαρητήρια για την εγκατάσταση του Joomla! CMS και παρατηρούμε το σημείωμα για διαγραφή του φακέλου **Installation** ο οποίος βρίσκεται μέσα στον φάκελο **htdocs** στην τοποθεσία **c:\XAMPP\htdocs\όνομα\_του\_site\_μας** (π.χ. **c:\XAMPP\htdocs\bpcSite**).

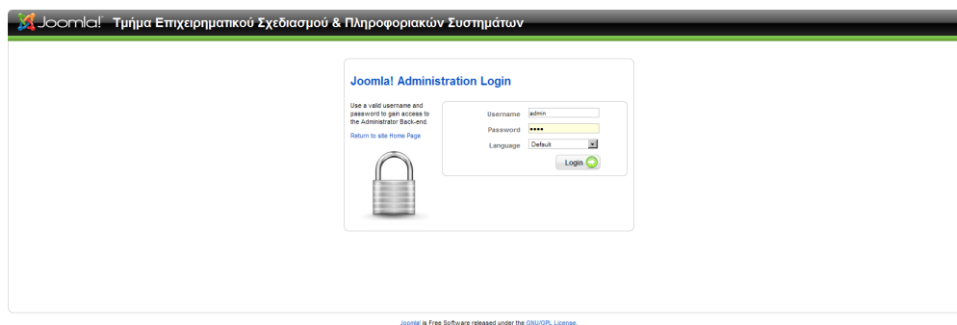


18. Τέλος πατάμε το κουμπί ιστότοπος στον φυλλομετρητή μας ή πλοηγόμασε στην διεύθυνση [http://localhost/όνομα\\_του\\_site\\_μας](http://localhost/όνομα_του_site_μας) (π.χ. brisSite). Παρατηρούμε το δημιουργημένο portal μας με εγκατεστημένο το ενδεικτικό περιεχόμενο.



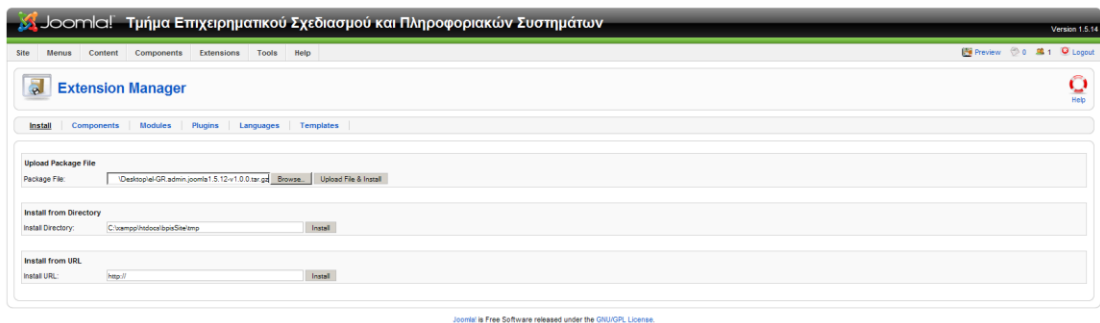
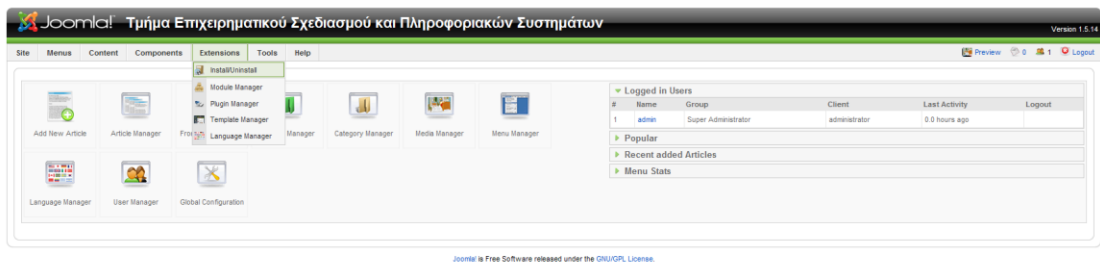
## Διαχείριση (Administration)

1. Για να κάνουμε διαχείριση του portal πατάμε το κουμπί στο τέλος της εγκατάστασης **Διαχείριση** ή πλοηγούμαστε μέσω του φυλλομετρητή μας στην διεύθυνση [http://localhost/όνομα\\_του\\_site\\_μας/administrator/](http://localhost/όνομα_του_site_μας/administrator/) (π.χ. <http://localhost/bpisSite/administrator/>) και εισάγουμε όνομα χρήστη και κωδικό που είχαμε δηλώσει κατά την εγκατάσταση.

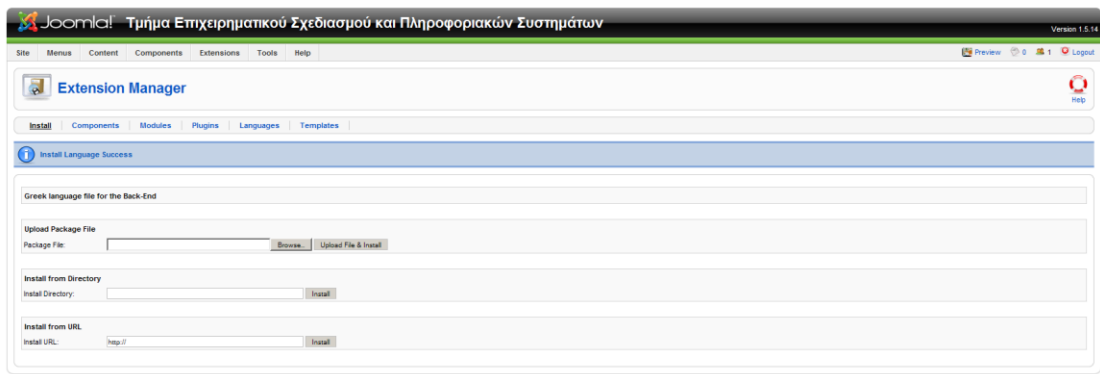


2. Για να εγκαταστήσουμε τα αρχεία Ελληνικής γλώσσας για το δημόσιο τμήμα και τη διαχείριση. Πλοηγούμαστε Extensions -> Install/Uninstall και πατώντας το κουμπί browse επιλέγουμε το αρχείο el-GR.admin.joomla.x.x.x-v.x.x.zip ή tar.gz

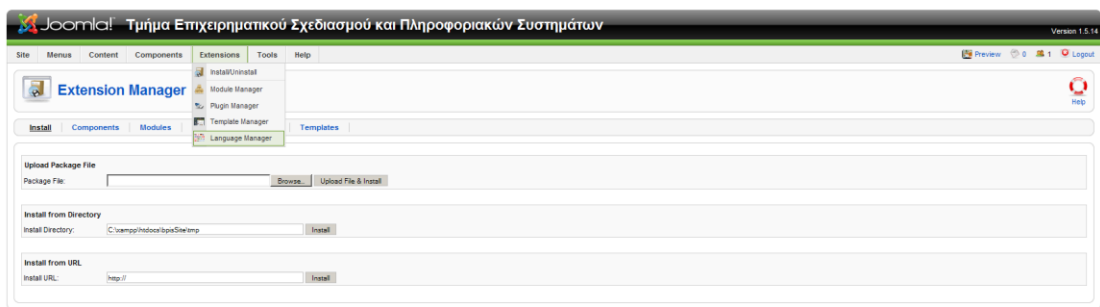




3. Πατώντας το κουμπί Upload File & Install, εμφανίζεται με μπλέ πλαίσιο ένα μήνυμα που μας αναφέρει την επιτυχημένη εγκατάσταση της Ελληνικής γλώσσας για το διαχειρηστικό τμήμα του CMS. Παρομοίως εγκαταθeshτούμε και το αρχείο Ελληνικής γλώσσας για το portal.



4. Αφου εγκαταστήσουμε την Ελληνική γλώσσα και για τα δύο κομμάτια του CMS πλοηγούμαστε Extensions->Language Manager για να επιλέξουμε ποια γλώσσα θα χρησιμοποιεί το διαχειριστικό τμήμα και το portal.



5. Διαλέγουμε Greek και πατάμε το κουμπί Default ενώ είναι πατημένη η υπερσύνδεση Site για να διαλέξουμε την Ελληνική γλώσσα να είναι η προεπιλεγμένη γλώσσα για το Site, αντίστοιχα κάνουμε και για το διαχειριστικό κομμάτι αφού πατήσουμε την υπερσύνδεση Administrator.

Joomla! Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων Version: 1.5.14

Site Menu Content Components Extensions Tools Help Preview 0 1 Logout

### Language Manager

Site Administrator

#	Language Name	Default	Version	Date	Author	Author E-mail
1	Greek		1.5.11	2009-05-12	Joomla.gr team	joomla@myjoomla.gr
2	English(United Kingdom)	★	1.5.9	2008-03-15	Joomla! Project	admin@joomla.org

Display # 20

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

Joomla! Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων Version: 1.5.14

Site Menu Content Components Extensions Tools Help Preview 0 1 Logout

### Language Manager

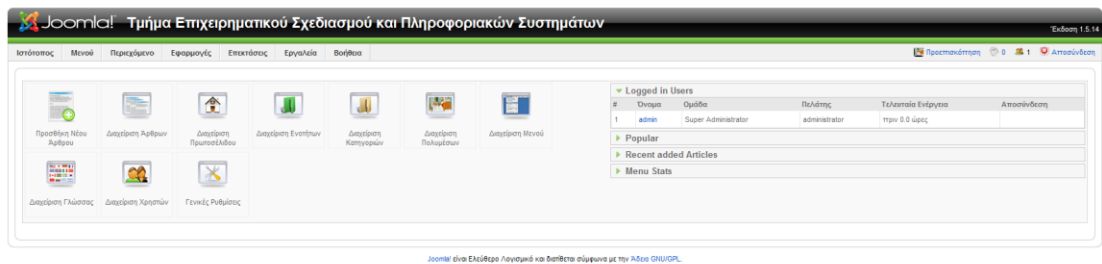
Site Administrator

#	Language Name	Default	Version	Date	Author	Author E-mail
1	Greek		1.5.11	2009-05-12	Joomla.gr team	joomla@myjoomla.gr
2	English(United Kingdom)	★	1.5.9	2008-03-15	Joomla! Project	admin@joomla.org

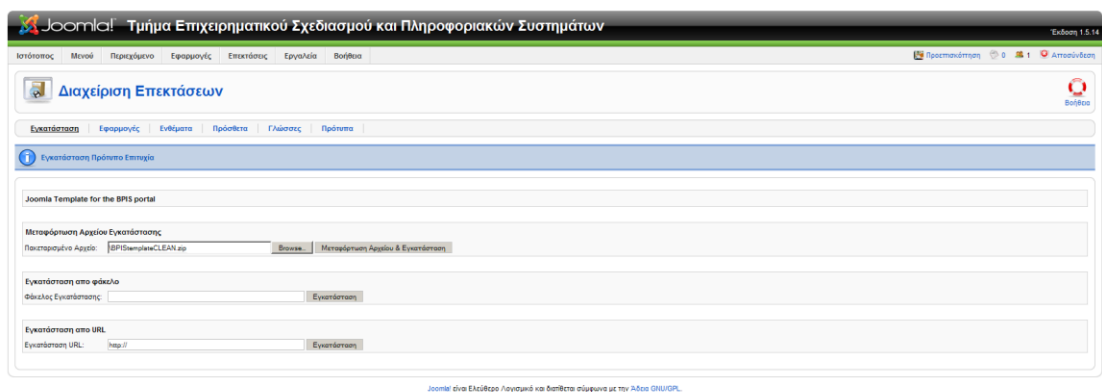
Display # 20

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

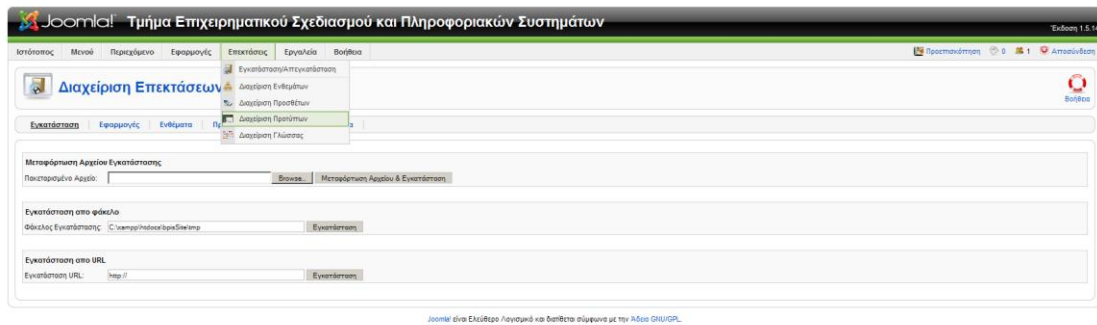
Έχοντας εγκαταστήσει σωστά τα αρχεία Ελληνικής γλώσσας βλέπουμε άμεσα το διαχειριστικό κομμάτι πλήρως Εξελληνισμένο.



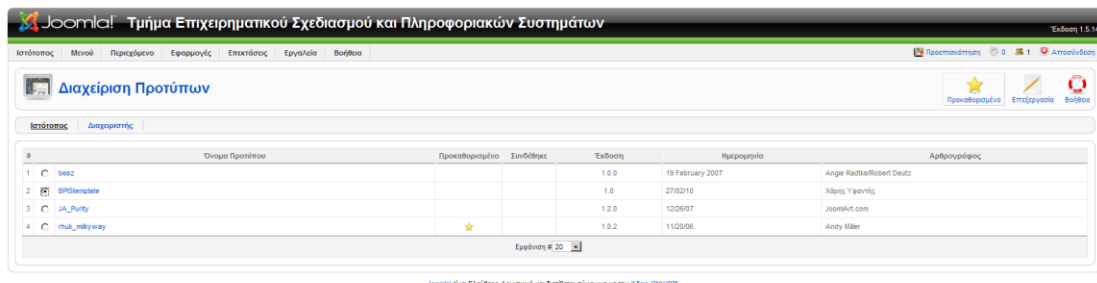
6. Για να κάνουμε εγκατάσταση ενός πρότυπου (template) ακολουθούμε την ίδια διαδικασία που ακολουθήσαμε για να κάνουμε εγκατάσταση των Ελληνικών. Πατώντας το κουμπί Μεταφόρτωση Αρχείου & Εγκατάσταση εμφανίζεται με μπλέ πλαίσιο ένα μήνυμα που μας αναφέρει την επιτυχή εγκατάσταση του προτύπου μας.

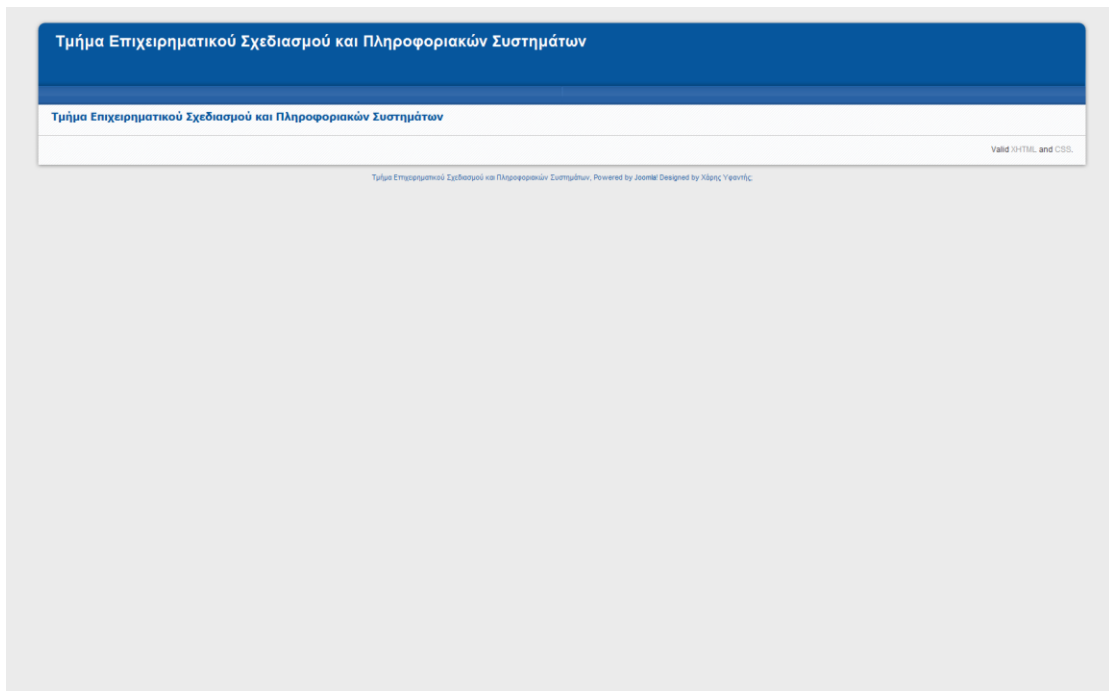


## 7. Πλοηγούμαστε Επεκτάσεις -> Διαχείριση Προτύπων



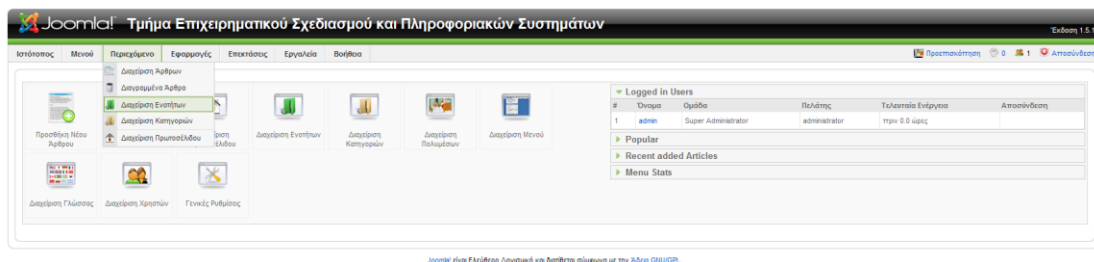
8. Επιλέγουμε το εγκατεστημένο πρότυπο και πατάμε το κουμπί Προκαθορισμένο και τέλος πατάμε το κουμπί προεπισκόπηση για να δούμε το portal με το νέο πρότυπο.

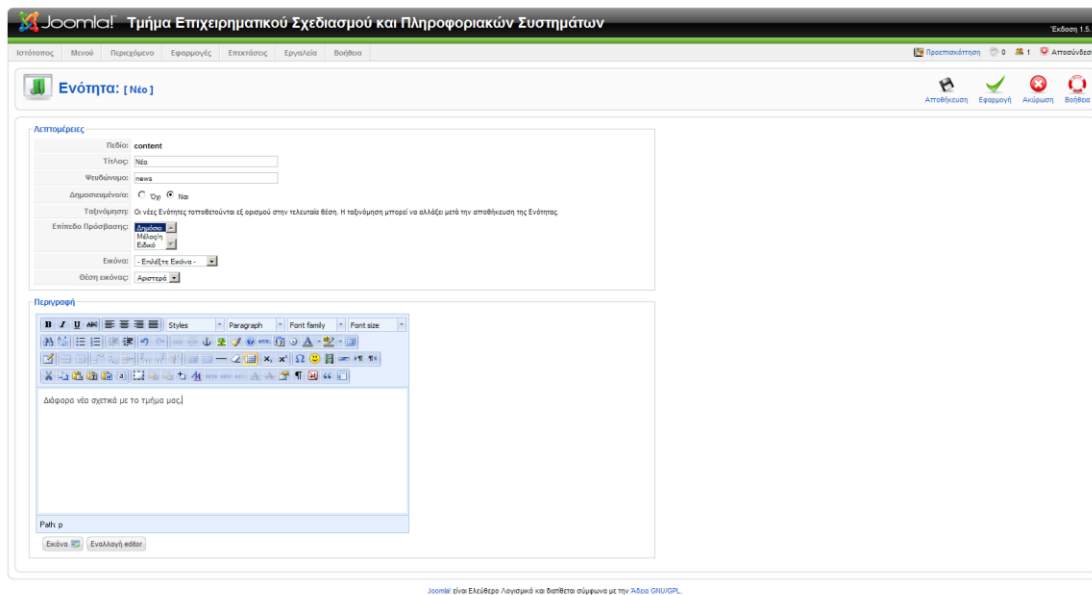




Το portal σε αυτήν την φάση είναι εντελώς άδειο και αυτό έχει γίνει σκόπιμα ώστε να μπορούμε να δούμε εύκολα βήμα βήμα κάθε αλλαγή που κάνουμε.

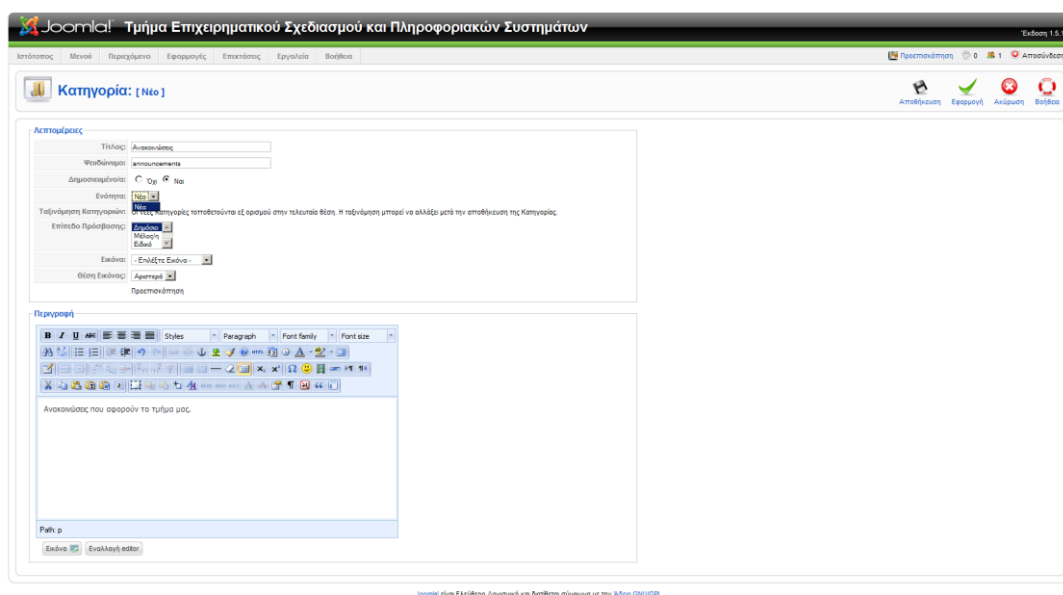
9. Γενικά το Joomla! CMS για να διαχωρίσει και να κατηγοριοποιήσει άρθρα για ευκολότερη διαχείριση χρησιμοποιεί δύο επίπεδα διαχωρισμού, τις Κατηγορίες (Categories) και τις Ενότητες (Sections). Με τις ενότητες στο υψηλότερο επίπεδο και τις κατηγορίες το κατώτερο. Δηλαδή μία ενότητα μπορεί να περιέχει πολλές κατηγορίες χωρίς όμως να ισχύει το αντίστροφο. Για να δημιουργήσουμε μία ενότητα πλοηγούμαστε Περιεχόμενο -> Διαχείριση Ενότητων.





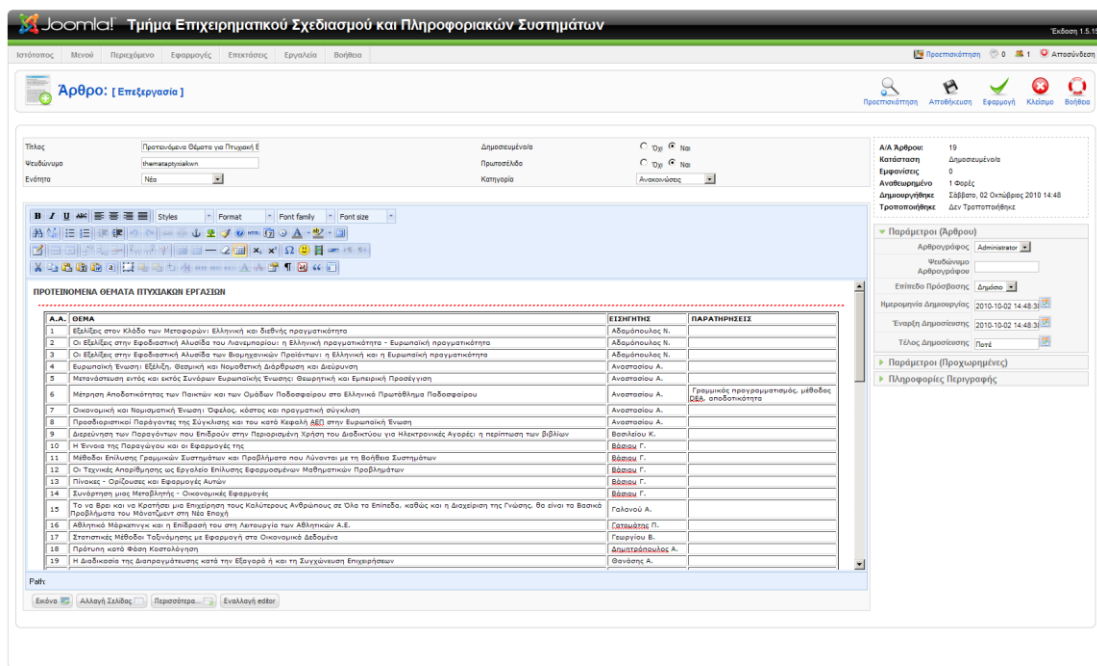
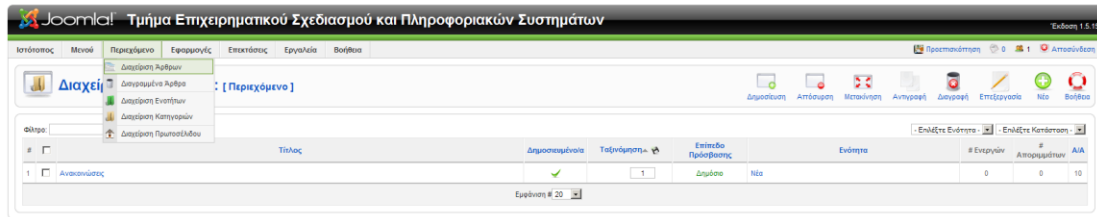
Έχοντας πατήσει στο κουμπί Νέο. Φτάνουμε στο σημείο που μπορούμε να δημιουργήσουμε μία ενότητα (π.χ. Νέα). Εκεί επιλέγουμε αυτή η ενότητα να είναι δημοσιευμένη και το επίπεδο πρόσβασης να είναι δημόσιο, μπορούμε επίσης να γράψουμε και μία σύντομη περιγραφή της ενότητας αυτής (π.χ. Διάφορα νέα σχετικά με το τμήμα μας). Τέλος πατάμε Αποθήκευση ή Εφαρμογή.

10. Έχοντας δημιουργήσει την ενότητα Νέα, μπορούμε να δημιουργήσουμε διάφορες κατηγορίες για αυτήν την ενότητα. Πλοηγούμαστε Περιεχόμενο->Διαχείριση κατηγοριών. Πατώντας το κουμπί Νέο φτάνουμε στο σημείο που μπορούμε να δημιουργήσουμε την κατηγορία μας.



Παρόμοια με την δημιουργία ενότητας εδώ δημιουργούμε την κατηγορία Ανακοινώσεις. Μπορούμε να διαλέξουμε αν θα είναι η κατηγορία δημοσιευμένη, σε ποια ενότητα ανήκει, κτλ.

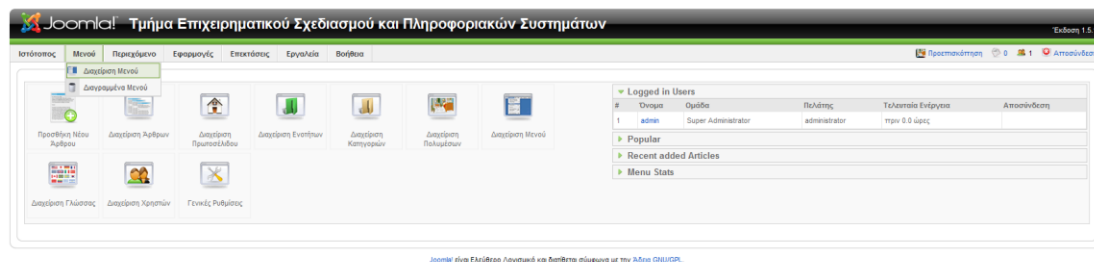
11. Έχοντας δημιουργήσει την ενότητα και την κατηγορία μας μπορούμε να δημιουργήσουμε ένα άρθρο το οποίο θα κατηγοριοποιηθεί σχετικά με την ενότητα και την κατηγορία μας. Πλοηγούμαστε Περιεχόμενο->Διαχείριση Άρθρων.

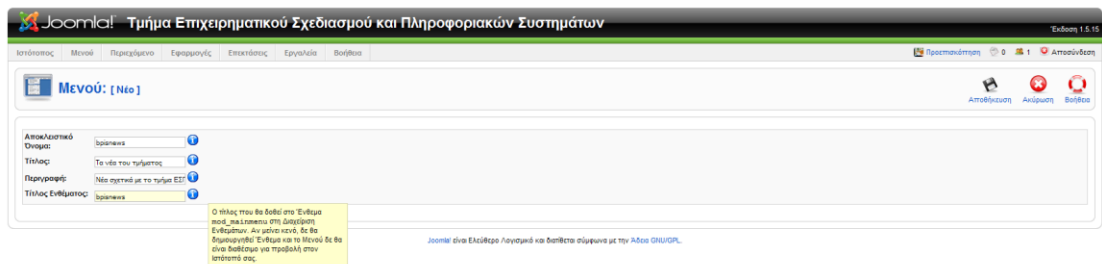




Σε αυτήν την οθόνη δημιουργούμε το άρθρο μας (π.χ. Προτεινόμενα θέματα για πτυχιακή Εργασία) και επιλέγουμε για ενότητα Νέα και κατηγορία Ανακοινώσεις), επιλέγουμε να είναι δημοσιευμένο και να εμφανίζεται στο πρωτοσέλιδο. Στα δεξιά του κειμενογράφου (editor) μπορούμε να παραμετροποιήσουμε το άρθρο μας σχετικά με το ποιός είναι ο αρθρογράφος, το επίπεδο πρόσβασης, την ημερομηνία δημιουργίας, την ημερομηνία έναρξης της δημοσίευσης και την ημερομηνία τέλους της δημοσίευσης. Τέλος είναι πολύ σημαντικό στο Πληροφορίες Περιγραφής στο πεδίο Μηχανές αναζήτησης να θέσουμε τις λέξεις “index, follow” έτσι ώστε το άρθρο μας αλλά και γενικά το portal μας να είναι SEO (Search engine optimized) ή SEF (Search engine Friendly). Ακόμα πρέπει να αναφερθεί ότι δεν είναι αναγκαίο να κατηγοριοποιηθούν τα άρθρα μας, δηλαδή δεν είναι ανάγκη να δημιουργήσουμε ενότητες και κατηγορίες, αλλά όπως θα δείτε και παρακάτω η κατηγοριοποίηση χρησιμεύει πάρα πολύ στον τρόπο που θα φαίνονται τα άρθρα μας στο portal.

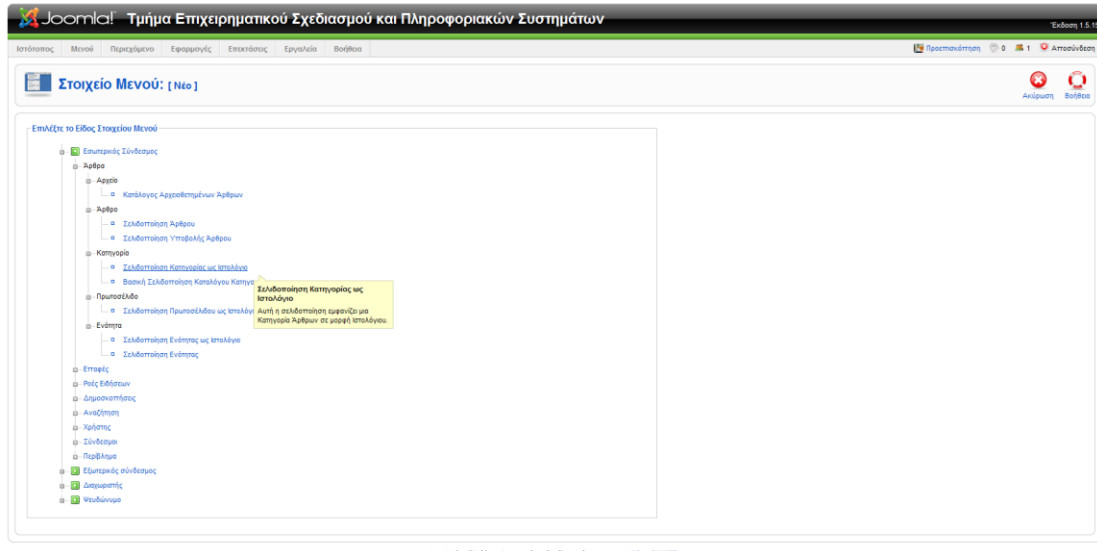
12. Αφού δημιουργήσαμε και το άρθρο μας θα πρέπει να δημιουργήσουμε και ένα μενού το οποίο θα περιέχει μία παραπομπή («κουμπί ή υπερσύνδεση») στο άρθρο μας. Πλοηγούμαστε Μενού->Διαχείριση Μενού.



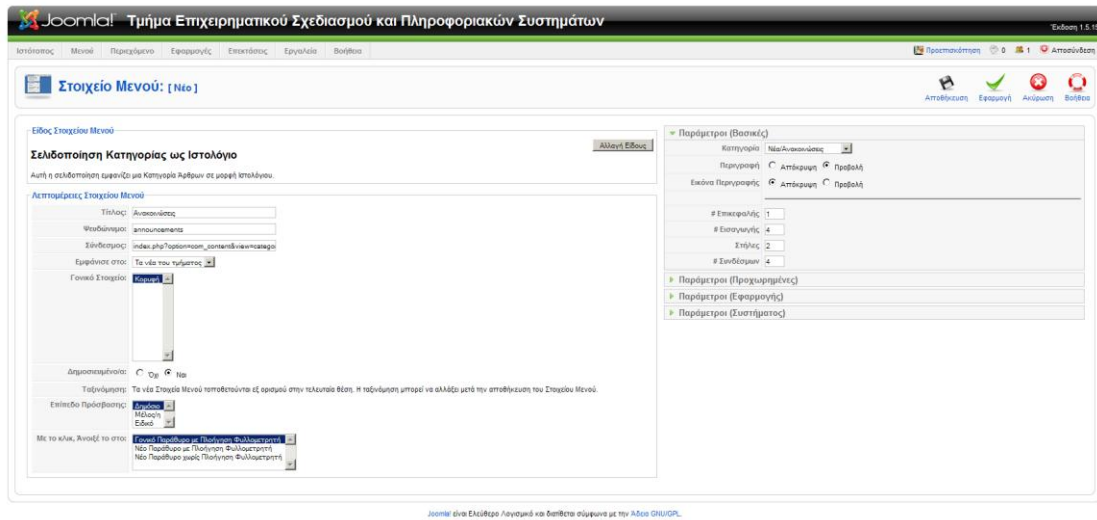


Εδώ βάζουμε το αποκλειστικό (μοναδικό) όνομα του μενού μας με λατινικούς χαρακτήρες. Με αυτό το όνομα το μενού θα αναφέρεται στα εσωτερικά του CMS σε επίπεδο κώδικα. Του δίνουμε έναν τίτλο και μία περιγραφή. Τέλος του θέτουμε και ένα τίτλο για το ένθεμα `mod_mainmenu`. Δεν μπορούμε να το αφήσουμε κενό διότι μετά δεν θα είναι διαθέσιμο για χρήση, επίσης αυτό το πεδίο ορίζεται μόνο κατά την δημιουργία του μενού και δεν μπορεί να αλλάξει αργότερα.

13. Αφού δημιουργήσαμε το μενού μας μπορούμε πλέον να προσθέσουμε στο μενού αυτό μία παραπομπή στο άρθρο που δημιουργήσαμε προηγουμένως. Πλοηγούμαστε Μενού->Όνομα του μενού που δημιουργήσαμε ή πατάμε στο όνομα του μενού που δημιουργήσαμε ή στο εικονίδιο Στοιχείο/α Μενού.

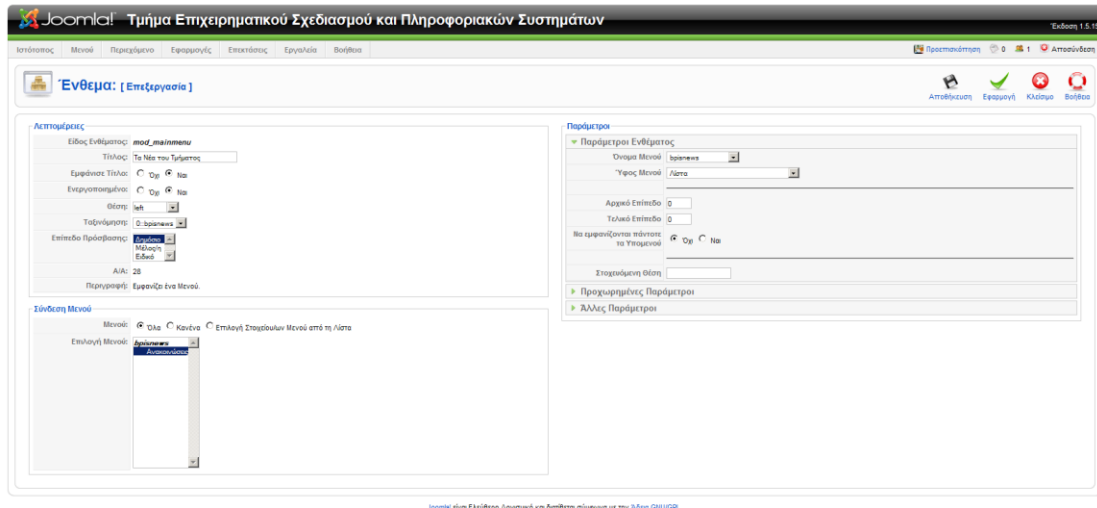


Εδώ έχουμε διάφορες επιλογές για το πώς θα φαίνονται τα άρθρα μας (μορφή) αν πλοηγηθούμε από αυτήν την παραπομπή. Εμείς επιλέξαμε η παραπομπή μας να μας εμφανίζει όλα τα άρθρα τα οποία ανήκουν στην κατηγορία Ανακοινώσεις με μορφή ιστολόγιου (blog).



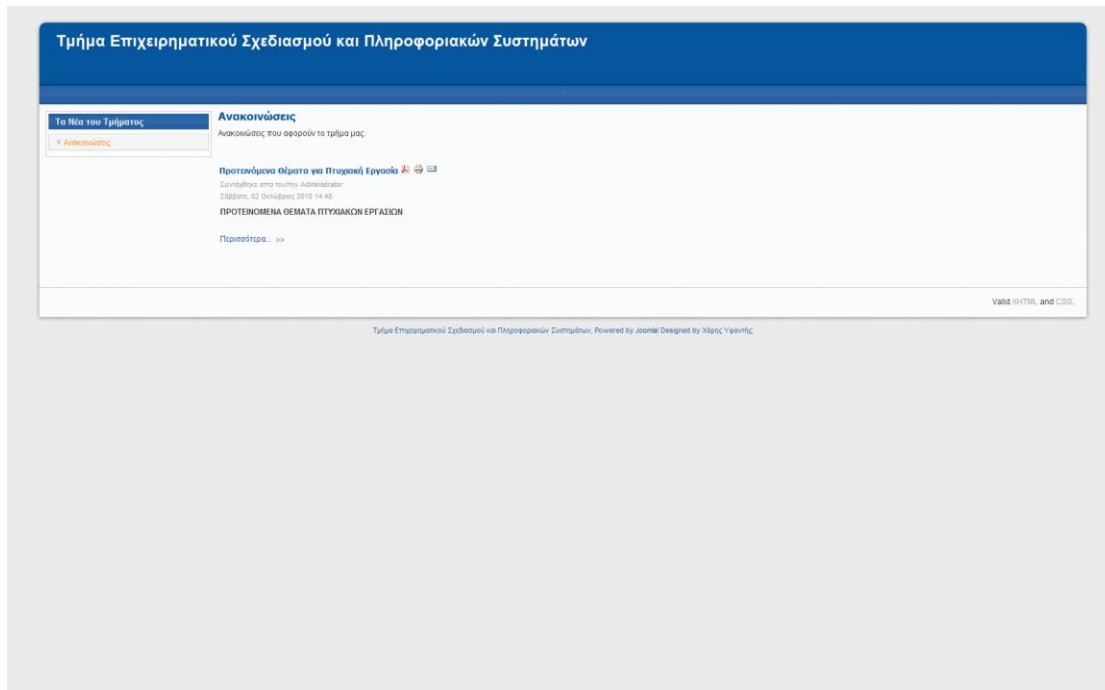
Κάνοντας αυτό φτάνουμε στο σημείο στο οποίο μπορούμε να ορίσουμε ένα τίτλο ο οποίος θα φαίνεται σαν υπερσύνδεση στο μενού μας και τέλος μπορούμε να διαλέξουμε ποια κατηγορία θέλουμε να φαίνεται και πώς π.χ. (#Επικεφαλής = 1 , #Εισαγωγής = 4, κτλ.)

14. Τέλος για να εμφανιστή το μενού μας, πλοηγούμαστε Επεκτάσεις->Διαχείριση Ενθεμάτων και επιλέγουμε το μενού που είχαμε δημιουργήσει και εμφανίζεται εδώ με τον τίτλο του ενθέματος που είχα δώσει στο βήμα 12.

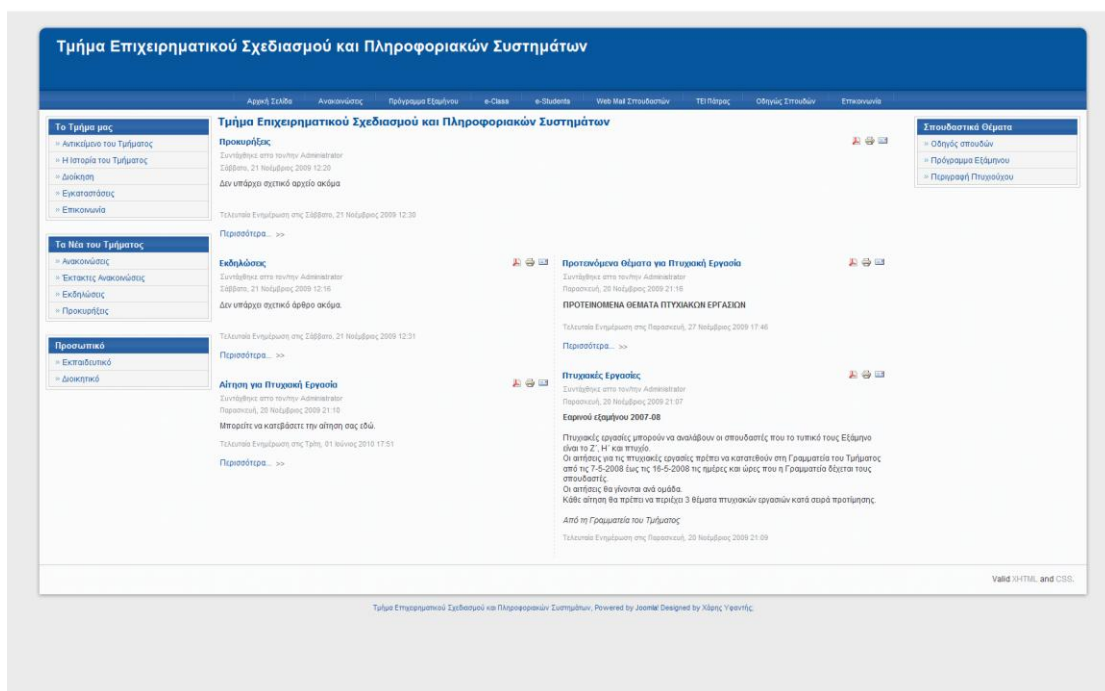


Του θέτουμε τίτλο που θα εμφανίζεται πάνω από το μενού σαν επικεφαλίδα, τη θέση του στην ιστοσελίδα σύμφωνα με το πρότυπο μας (κάθε πρότυπο έχει δικές του θέσεις). Τέλος μπορούμε επίσης να του ορίσουμε το ύψος του μενού, δηλαδή αν θα φαίνεται ως κάθετο ή οριζόντιο μενού.

Έτσι αποθηκεύοντας τις αλλαγές μας και πατώντας προεπισκόπηση βλέπουμε το portal μας να έχει ένα μενού στα δεξιά και την κατηγορία μας να εμφανίζει το άρθρο που ορίσαμε και το μενού.

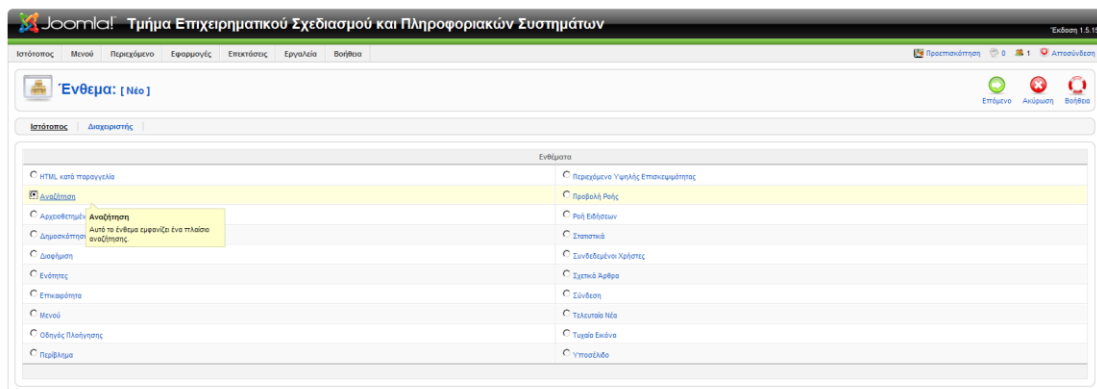
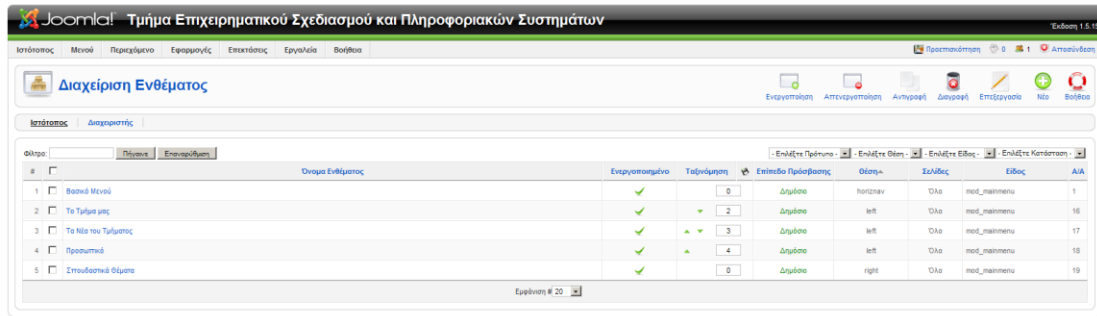


Ακολουθώντας τα βήματα που περιγράψαμε αρκετές φορές μπορούμε να δημιουργήσουμε μια πλήρες ιστοσελίδα με πολύ περιεχόμενο καλά οργανωμένο.



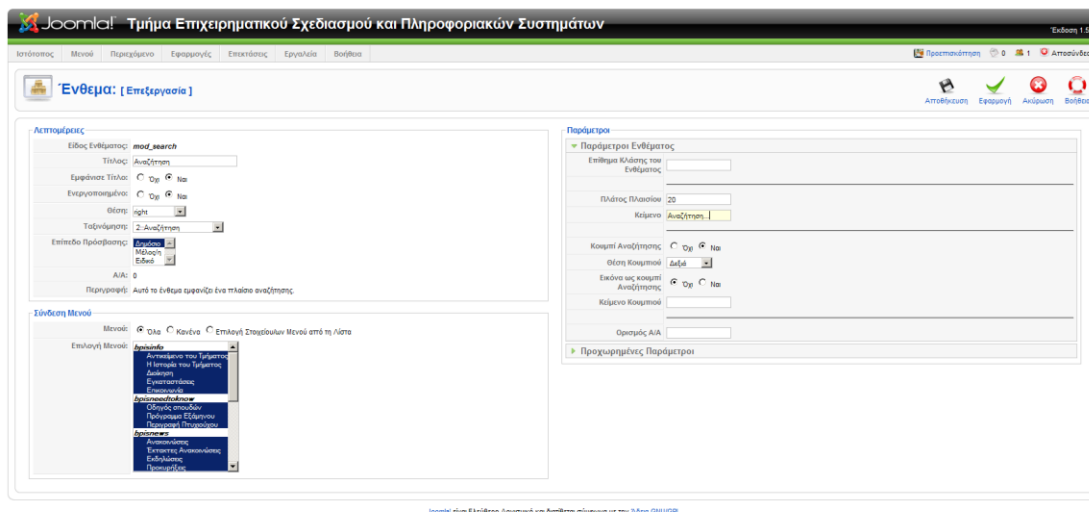
15. Επειδή το Joomla! CMS ακολουθεί «αρθρωτή» (modular) λογική, αντιμετωπίζει τα μενού και όλων των ειδών τα εργαλεία ως ενθέματα ή προεκτάσεις (plug-ins). Το Joomla! CMS έρχεται με ένα σύνολο ενθεμάτων και προεκτάσεων προεγκατεστημένα έτσι μπορούμε να βάλουμε σχετικά εύκολα ένα ένθεμα αναζήτησης ή ροή ειδήσεων (RSS feed). Για να βάλουμε ένα ένθεμα πλοηγούμαστε

Επεκτάσεις->Διαχείριση Ενθεμάτων και πατάμε Νέο.



Εκεί επιλέγουμε το ένθεμα που θέλουμε να προσθέσουμε στο portal μας, παραδείγματος χάριν επιλέγουμε το ένθεμα Αναζήτηση.

16. Πατώντας το κουμπί επόμενο ερχόμαστε στο σημείο που μπορούμε να παραμετροποιήσουμε το ένθεμα μας. Θέτουμε έναν τίτλο, την θέση του, αν είναι δημοσιευμένο ή όχι και κάποιες πιο συγκριμένες παραμέτρους ανάλογα με το ένθεμα που έχουμε επιλέξει, όπως πλάτος του πλαισίου της αναζήτησης κτλ.



Πατώντας αποθήκευση και προεπισκόπηση μπορούμε να δούμε το ένθεμα αναζήτηση στο portal μας.

**Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων**

Αρχική Σελίδα Ανακοινώσεις Πρόγραμμα Εξέλιξης e-Class e-Students Web Mail Σπουδαστών ΤΠΕ Πάρος Οδηγός Σπουδών Επικοινωνία

**Το Τμήμα μας**

- » Αντικείμενο του Τμήματος
- » Η Ιστορία του Τμήματος
- » Διοίκηση
- » Εκπαίδευση
- » Επισκοπικά

**Τα Ίδια του Τμήματος**

- » Ανακοινώσεις
- » Εκπαιδευτικές Ανακοινώσεις
- » Εκδηλώσεις
- » Προκηρύξεις

**Προσωπικό**

- » Εκπαιδευτικό
- » Διοικητικό

**Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων**

**Προκηρύξεις**

Συντάχθηκε από τον/την Administrator  
Σάββατο, 21 Νοεμβρίου 2009 12:20  
Δεν υπάρχει σχετικό αρχείο ακόμα

Τελευταία Ενθάρτυση στις 3:58μμ., 21 Νοεμβρίου 2009 12:30

Περισσότερα... >>

**Εκδηλώσεις**

Συντάχθηκε από τον/την Administrator  
Σάββατο, 21 Νοεμβρίου 2009 12:16  
Δεν υπάρχει σχετικό αρχείο ακόμα.

Τελευταία Ενθάρτυση στις 3:58μμ., 21 Νοεμβρίου 2009 12:21

Περισσότερα... >>

**Αίτηση για Πτυχιακή Εργασία**

Συντάχθηκε από τον/την Administrator  
Παρασκευή, 20 Νοεμβρίου 2009 21:18  
Μπορείτε να κατεβάσετε την αίτηση ος εδώ.

Τελευταία Ενθάρτυση στις Τρίτη, 01 Ιουνίου 2010 17:51

Περισσότερα... >>

**Προτεινόμενα Θέματα για Πτυχιακή Εργασία**

Συντάχθηκε από τον/την Administrator  
Παρασκευή, 20 Νοεμβρίου 2009 21:18

**ΠΡΟΤΕΙΝΟΜΕΝΑ ΘΕΜΑΤΑ ΠΤΥΧΙΑΚΩΝ ΕΡΓΑΣΙΩΝ**

Τελευταία Ενθάρτυση στις Παρασκευή, 27 Νοεμβρίου 2009 17:48

Περισσότερα... >>

**Πτυχιακές Εργασίες**

Συντάχθηκε από τον/την Administrator  
Παρασκευή, 20 Νοεμβρίου 2009 21:07

**Εαρινό εξάμηνο 2007-08**

Πτυχιακές εργασίες μπορούν να αναλάβουν οι σπουδαστές που το τυπικό τους Εξέλιγμα είναι το Z, H' και πτυχίο.  
Οι αιτήσεις για τις πτυχιακές εργασίες πρέπει να κατατεθούν στη Γραμματεία του Τμήματος από τις 7-5-2008 έως τις 18-5-2008 τις ημέρες και ώρες που η Γραμματεία δέχεται τους σπουδαστές.  
Οι αιτήσεις θα γίνονται ανά ομάδα.  
Κάθε αίτηση θα πρέπει να περιέχει 3 θέματα πτυχιακών εργασιών κατά σειρά προτίμησης.  
Από τη Γραμματεία του Τμήματος  
Τελευταία Ενθάρτυση στις Παρασκευή, 20 Νοεμβρίου 2009 21:09

**Σπουδαστικά Θέματα**

- » Οδηγός σπουδών
- » Πρόγραμμα Εξέλιξης
- » Περιγραφή Πτυχιακού

**Αναζήτηση**

Αναζήτηση...

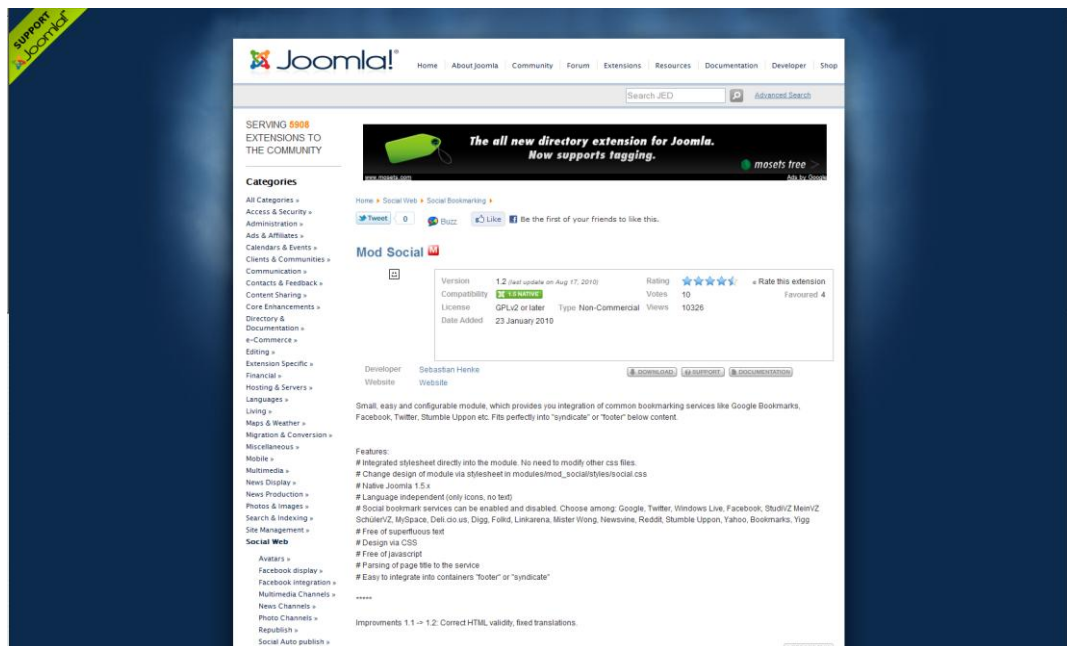
Valid XHTML and CSS.

Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων, Powered by Joomla! Designed by Χάρης Γιαννής

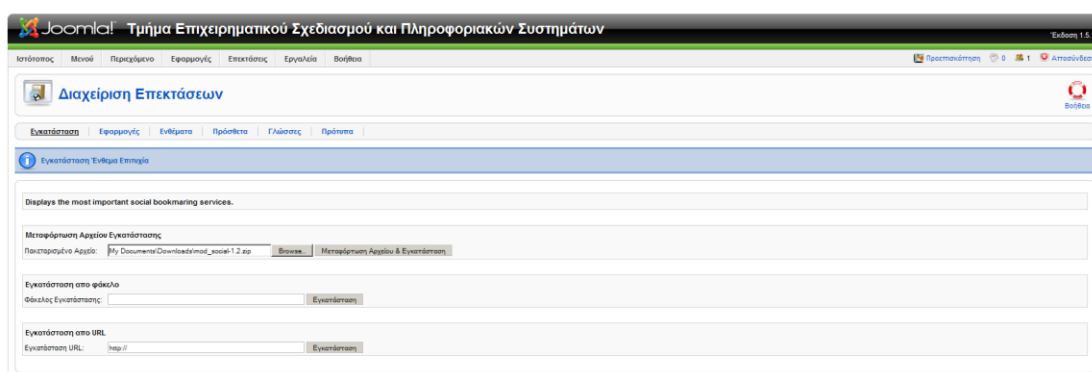
Ακολουθώντας την ίδια διαδικασία μπορούμε να ενσωματώσουμε πολλά χρήσιμα εργαλεία τα οποία δεν κάνουν μόνο πιο πλήρες το portal μας αλλά βοηθάνε και στην λειτουργικότητα (usability) του portal.



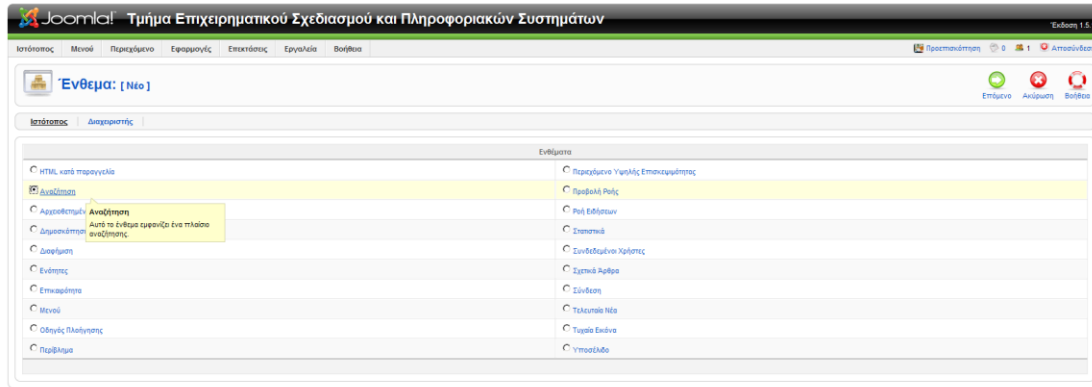
17. Μπορούμε επίσης να βάλουμε και ενθέματα ή προεκτάσεις τα οποία δεν είναι προεγκατεστημένα, το JED (Joomla! Extension Directory) έχει μια πληθώρα εργαλείων για όλες τις δουλειές. Εμείς διαλέξαμε να εγκαταστήσουμε ένα ένθεμα για Social Bookmarking (Επισήμανσης και δημοσίευσης στις υπηρεσίες κοινωνικής δικτύωσης) το οποίο ονομάζεται Mod Social. Πλοηγούμαστε με τον φυλλομετρητή μας στον ιστότοπο <http://extensions.joomla.org/> και κατεβάζουμε το συμπιεσμένο αρχείο του ενθέματος.



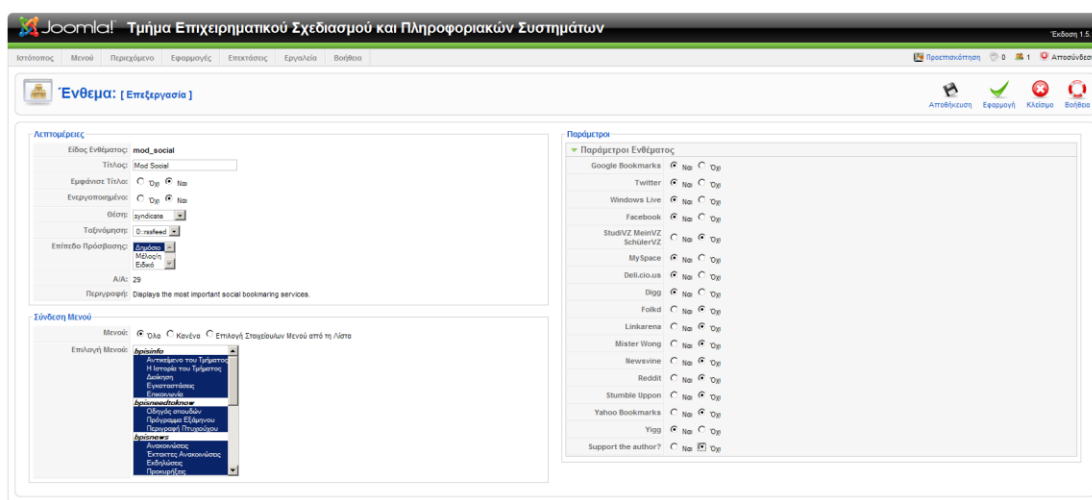
18. Έχοντας κατεβάσει το ένθεμα μας πλοηγούμαστε στο διαχειριστικό κομμάτι του Joomla! CMS και πηγαίνουμε Εργαλεία->Εγκατάσταση/Απεγκατάσταση. Επιλέγουμε το αρχείο mod\_social-x.x.zip και το εγκαθιστούμε.



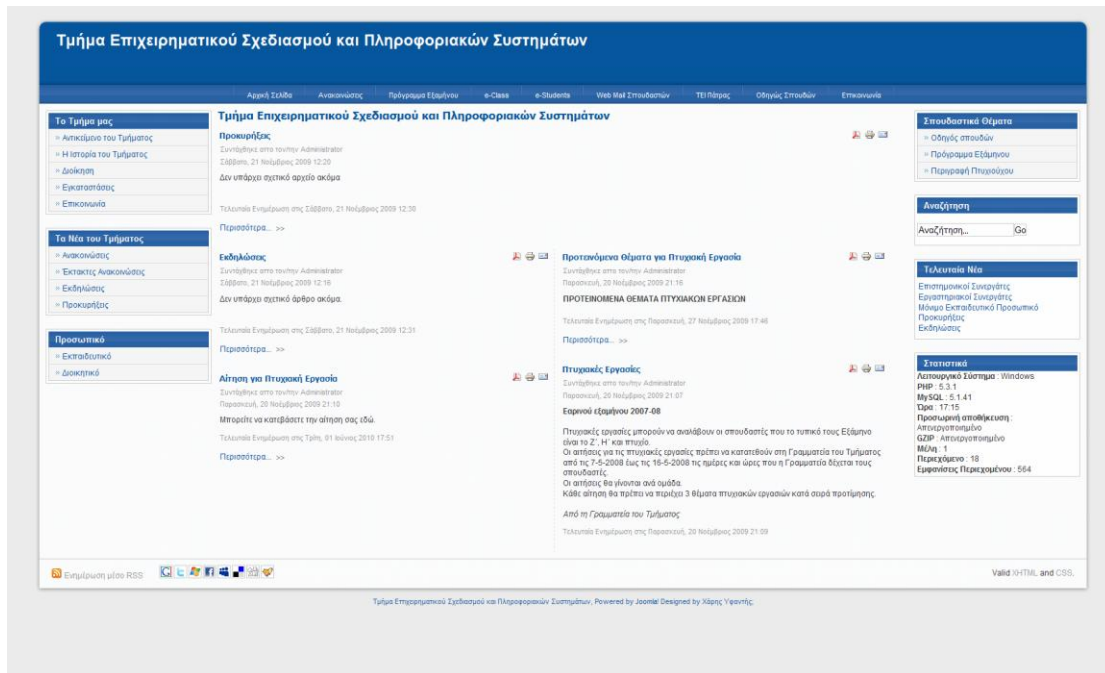
19. Αφού το εγκαταστήσουμε πρέπει να το ενεργοποιήσουμε για να το χρησιμοποιήσουμε. Πλοηγούμαστε Εργαλεία->Διαχείριση Ενθεμάτων και επιλέγουμε το Mod Social και πατάμε το κουμπί Επεξεργασία ή απλά πατάμε το όνομα Mod Social που είναι υπερσύνδεση.



20. Αφού πατήσουμε επεξεργασία φτάνουμε στο σημείο που μπορούμε να παραμετροποιήσουμε το ένθεμα μας. Εκεί θέτουμε έναν τίτλο, αν θα είναι ενεργοποιημένο ή την θέση του. Στην δεξιά πλευρά της οθόνης αυτό το ένθεμα μας επιτρέπει να διαλέξουμε ποιά «κουμπιά» Επισήμανσης και δημοσίευσης στις υπηρεσίες κοινωνικής δικτύωσης θα φαίνονται στο portal.



Έχοντας θέσει για θέση syndicate βλέπουμε χαμηλά στο portal τα «κουμπιά» για τις υπηρεσίες κοινωνικής δικτύωσης.



Ακολουθώντας αυτά τα απλά βήματα μπορούμε να εξελίξουμε και να συμπληρώσουμε το portal μας ανάλογα με τις ανάγκες, επιτυγχάνοντας έτσι ένα πολύ χρηστικό και δυναμικό portal που καλύπτει όλες τις απαιτήσεις των χρηστών του.

## Βιβλιογραφία

*About Apache*. (n.d.). Ανάκτηση από Apache HTTP Server Project:

[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)

*Absolute Beginners Guide to Joomla!* (n.d.). Ανάκτηση από Joomla Documentation:

<http://docs.joomla.org/Beginners>

*Apache HTTP Server*. (n.d.). Ανάκτηση από Wikipedia:

[http://en.wikipedia.org/wiki/Apache\\_web\\_server](http://en.wikipedia.org/wiki/Apache_web_server)

Ben Laurie, P. L. *Apache The Definitive Guide, Thrid Edition*. O'Reilly & Associates.

*Content management system*. (n.d.). Ανάκτηση από Wikipedia:

[http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)

*Content Management System, Joomla*. (n.d.). Ανάκτηση από Infinity:

<http://www.infinity.gr/el/joomla>

*Εισαγωγή στα Συστήματα Διαχείρισης Περιεχομένου*. (n.d.). Ανάκτηση από

SlideShare: <http://www.slideshare.net/rodotheos/cms-421346>

Gianni, A. (2002, Απρίλιος 8). *Database-Driven Web Sites, Taking your Web presence to the next level*. Ανάκτηση από Techsoup.org:

<https://www.techsoup.org/learningcenter/databases/archives/page10135.cfm>

Gutmans, A. (2004, Μαρτίου 18). *What's New in PHP 5?* Ανάκτηση από Zend

Developer Zone: <http://devzone.zend.com/article/1714>

*History of PHP*. (n.d.). Ανάκτηση από php.net:

<http://www.php.net/manual/en/history.php.php>

Jason Gerner, E. N. (2006). *Professional LAMP, Linux, Apache, MySQL and PHP5 Web Development*, ISBN: 978-0-7645-9723-7. Wiley Publishing, Inc.

*Joomla*. (n.d.). Ανάκτηση από Wikipedia: <http://en.wikipedia.org/wiki/Joomla>

*List of AMP packages*. (n.d.). Ανάκτηση από Wikipedia:

[http://en.wikipedia.org/wiki/List\\_of\\_AMP\\_packages](http://en.wikipedia.org/wiki/List_of_AMP_packages)

Meloni, J. (2006). *Teach Yourself PHP, MySQL and Apache: All in One, Third Edition*, ISBN: 978-0-672-32873-2. Sams Publishing.

*MySQL*. (n.d.). Ανάκτηση από Wikipedia: <http://en.wikipedia.org/wiki/MySQL>

*PHP*. (n.d.). Ανάκτηση από Wikipedia: <http://en.wikipedia.org/wiki/Php>

- Rahmel, D. (2009). *Beginning Joomla! Second Edition* ISBN: 978-1-4302-1643-8. Apress.
- Randy Jay Yarger, G. R. (1999). *MySQL & mSQL*. O'Reilly & Associates.
- Steven Holzner, N. C. (2009). *Joomla! for Dummies*, ISBN: 978-0-470-43287-7. Wiley Publishing, Inc.
- The Foundation*. (n.d.). Ανάκτηση από The Apache Software Foundation:  
<http://www.apache.org/foundation/>
- Tiggeler, E. (2010). *Joomla! 1.5 Beginner's Guide*, ISBN: 978-1-847199-90-4. Packt Publishing.
- Top Reasons to Use MySQL*. (n.d.). Ανάκτηση από MySQL:  
<http://www.mysql.com/why-mysql/topreasons.html>
- Valade, J. (2004). *PHP & MySQL for Dummies, Second Edition*, ISBN: 0-7645-5589-8. Wiley Publishing, Inc.
- Valade, J. *PHP 5 for Dummies* ISBN: 978-0-7645-4166-7. John Wiley & Sons.
- Walker, C. (2006, Σεπτεμβρίου 11). *Types of portal: a definition*. Ανάκτηση από Step Two Designs: [http://www.steptwo.com.au/papers/cmb\\_portaldefinitions/index.html](http://www.steptwo.com.au/papers/cmb_portaldefinitions/index.html)
- Web Server Survey*. (n.d.). Ανάκτηση από NetCraft:  
<http://news.netcraft.com/archives/category/web-server-survey/>
- XAMPP*. (n.d.). Ανάκτηση από Wikipedia: <http://en.wikipedia.org/wiki/Xampp>
- Μετάφραση όρων πληροφορικής*. (n.d.). Ανάκτηση από Βικιπαίδεια:  
[http://el.wikipedia.org/wiki/Βικιπαίδεια:Μετάφραση\\_όρων\\_πληροφορικής](http://el.wikipedia.org/wiki/Βικιπαίδεια:Μετάφραση_όρων_πληροφορικής)
- Ξεκινώντας από το μηδέν*. (n.d.). Ανάκτηση από joomla.gr:  
<http://www.joomla.gr/tutorials/use-joomla/409-from-zero>
- Τι είναι Portal ?* (n.d.). Ανάκτηση από Internet Promotions:  
<http://www.ip.gr/el/dictionary/165-Portal>