



**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ**

***Πτυχιακή Εργασία***

**Θέμα : Σχεδίαση – Δημιουργία Δυναμικής Ιστοσελίδας Εύρεσης και  
Καταχώρησης Αγγελιών Εργασίας**

**Επιβλέπων Καθηγητής : κ. Λογκάκη Μαρία**

**Νικολαρόπουλος Αλέξανδρος Α.Μ. 1332  
Δρακόπουλος Δημήτριος-Αριστοτέλης Α.Μ. 2052**

**Ηράκλειο, 2011**

## Σύνοψη

Σκοπός της παρούσας πτυχιακής, ήταν η πλήρης κατανόηση ενός από τα δημοφιλέστερα συστήματα διαχείρισης περιεχομένου, του Joomla, με τη βοήθεια του οποίου θα υλοποιούσαμε μια ιστοσελίδα αναζήτησης και εύρεσης εργασίας. Αποφασίσαμε τη δημιουργία της συγκεκριμένης σελίδας, πιστεύοντας πως στην Ελλάδα ακόμα ο συγκεκριμένος τομέας, δεν έχει αναπτυχθεί επαρκώς και θέλαμε να παρουσιάσουμε τη δική μας πρόταση . Η ιστοσελίδα θα χωρίζει τους χρήστες της, σε εργαζομένους και εργοδότες, προσφέροντας σε αυτούς και τα αντίστοιχα προνόμια. Οι εργοδότες θα προσφέρουν τις αγγελίες τους, ενώ οι εργαζόμενοι θα τις αναζητούν για να υποβάλουν το βιογραφικό τους. Εισάγουμε την δυνατότητα σχολιασμού των αγγελιών, ενσωμάτωση σελίδων κοινωνικού δικτύου και άλλες ευκολίες που κάνουν την σελίδα μας πιο ελκυστική. Στα πρώτα κεφάλαια θα γίνει μια εισαγωγή στα συστήματα διαχείρισης περιεχομένου και ειδικότερα του Joomla. Εν συνεχεία θα γίνει η ανάλυση των απαιτήσεων που θέσαμε για την πτυχιακή μας και τέλος πως πραγματοποιήθηκε η υλοποίηση αυτών.

## **Abstract**

The purpose of the current thesis is to fully present and understand the inner mechanics of Joomla, one of the most widely used Content Management System (CMS). We will use Joomla in order to create a job search engine. It is our belief that Greece lacks of innovative sites in this domain and we would like to present a different approach in job seeking. Our implementation separates the users in two main groups, the employers and the employees and presents to them different tools and options. Social networking, advanced search options, comments and other Web 2.0 tools are offered through our site in order to make the whole process of job seeking more intuitive and interactive. We believe that our approach offers greater opportunities of communication between employers and employees. In the first chapters of the thesis we will present the different CMS that exist today in the market and we will focus on the capabilities of Joomla. From there we will present the goals and the requirements of our implementation and we will conclude the thesis with a thorough analysis of our implementation.

# Περιεχόμενα

<b>1. ΕΙΣΑΓΩΓΗ</b> .....	6
<b>2. ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΙΡΕΧΟΜΕΝΟΥ</b> .....	7
2.1 Ορισμός CMS.....	7
2.2 Ανατομία ενός CMS.....	8
2.2.1 Δημιουργία περιεχομένου.....	8
2.2.2 Διαχείριση Περιεχομένου.....	8
2.2.3 Δημοσίευση.....	9
2.2.4 Παρουσίαση.....	9
2.3 Οφέλη και πλεονεκτήματα.....	10
2.4 Τα δημοφιλέστερα CMS.....	11
<b>3. ΠΑΡΟΥΣΙΑΣΗ JOOMLA</b> .....	13
3.1 Ορισμός Joomla.....	13
3.2 Χαρακτηριστικά Joomla.....	14
3.3 Απαιτήσεις συστήματος για Joomla.....	14
3.4 Apache Web Server.....	15
3.4.1 Το ξεκίνημα και η εξέλιξη του Apache.....	16
3.4.2 Ο Apache σήμερα.....	16
3.4.3 Η προέλευση του ονόματος του.....	16
3.4.4 Χαρακτηριστικά και λειτουργίες του Apache HTTP.....	16
3.5 PHP.....	18
3.5.1 Χρήση PHP.....	18
3.5.2 Πλεονεκτήματα PHP.....	19
3.6 MySql.....	19
3.6.1 Πλεονεκτήματα της MySQL.....	20
<b>4. ΕΓΚΑΤΑΣΤΑΣΗ JOOMLA</b> .....	21
4.1 Εγκατάσταση Wamp.....	21
4.2 Οδηγός Εγκατάστασης Joomla.....	24
4.2.1 Βήμα 1 : Γλώσσα.....	24

4.2.2 Βήμα 2 : Προληπτικός Έλεγχος.....	25
4.2.3 Βήμα 3 : Άδεια Χρήσης.....	25
4.2.4 Βήμα 4 : Βάση Δεδομένων.....	26
4.2.5 Βήμα 5 : Ρυθμίσεις FTP.....	28
4.2.6 Βήμα 6 : Ρυθμίσεις.....	28
4.2.7 Βήμα 7 : Τέλος.....	29
4.3 Περιβάλλον Διαχείρισης Joomla.....	31
<b>5. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ.....</b>	<b>35</b>
5.1 Εισαγωγή.....	35
5.2 Βάση Δεδομένων.....	36
5.3 Σχεδίαση Σελίδας (template) .....	36
5.4 Γενικό Μενού.....	38
5.5 Εγγραφή χρήστη.....	39
5.6 Αλλαγή στοιχείων χρήστη.....	40
5.7 Αναζήτηση.....	41
5.8 Εισαγωγή / Ενημέρωση Αγγελίας.....	41
5.9 Παρουσίαση Αγγελίας.....	42
5.10 Σχόλια.....	43
5.11 Είσοδος στην Ιστοσελίδα ( Log In ) .....	43
<b>6. ΥΛΟΠΟΙΗΣΗ ΙΣΤΟΣΕΛΙΔΑΣ.....</b>	<b>44</b>
6.1 Εισαγωγή.....	44
6.2 Βάση Δεδομένων.....	44
6.3 Σχεδίαση Σελίδας (template) .....	51
6.4 Γενικό Μενού.....	52
6.5 Εγγραφή χρήστη.....	55
6.5.1 Υλοποίηση φόρμας .....	56
6.5.2 Έλεγχος Φόρμας.....	59
6.5.3 Υποβολή στοιχείων της φόρμας.....	61
6.6 Αλλαγή στοιχείων χρήστη.....	63
6.7 Αναζήτηση.....	65
6.8 Εισαγωγή / Ενημέρωση Αγγελίας.....	73
6.9 Παρουσίαση Αγγελίας.....	79

6.10 Σχόλια.....	88
6.11 Είσοδος στην Ιστοσελίδα ( Log In ).....	91
<b>7. ΙΣΤΟΣΕΛΙΔΑ ΣΤΟ ΔΙΑΔΙΚΤΥΟ.....</b>	<b>92</b>
Συμπεράσματα – Επίλογος.....	92
Βιβλιογραφία.....	93

# Κεφάλαιο 1 Εισαγωγή

Στην εποχή μας, η έννοια του ιστότοπου έχει γίνει σε όλους μέρος της καθημερινότητας μας και εμφανίζεται σε όλους τους τομείς της ζωή μας.

Πολλές ιστοσελίδες μαζί συνθέτουν έναν ιστότοπο. Οι σελίδες ενός ιστότοπου εμφανίζονται κάτω από το ίδιο όνομα χώρου (domain) π.χ. microsoft.com. Οι ιστοσελίδες αλληλοσυνδέονται και μπορεί ο χρήστης να μεταβεί από τη μία στην άλλη κάνοντας «κλικ», επιλέγοντας δηλαδή συνδέσμους που υπάρχουν στο κείμενο ή στις φωτογραφίες της ιστοσελίδας. Οι σύνδεσμοι προς άλλες σελίδες εμφανίζονται συνήθως υπογραμμισμένοι και με μπλε χρώμα για να είναι γρήγορα ξεκάθαρο στον επισκέπτη ότι πρόκειται για σύνδεσμο προς άλλη ιστοσελίδα, χωρίς όμως πάντα να είναι αυτό απαραίτητο.

Η **κατασκευή** ιστοσελίδων είναι κάτι που μπορεί να γίνει πολύ εύκολα με συστήματα διαχείρισης περιεχομένου, αλλά υπάρχουν και αυτοματοποιημένοι μηχανισμοί κατασκευής ιστοσελίδων που επιτρέπουν σε απλούς χρήστες να δημιουργήσουν εύκολα και γρήγορα προσωπικές ή και εμπορικές ιστοσελίδες. Από την άλλη μεριά υπάρχουν και πολλές εταιρίες, που εξειδικεύονται στη δημιουργία ελκυστικών και λειτουργικών ιστοσελίδων που έχουν σαν στόχο να οδηγήσουν τους επισκέπτες στην αγορά κάποιου προϊόντος, στην επικοινωνία με τον ιδιοκτήτη του ιστότοπου ή απλά στο ανέβασμα του εταιρικού προφίλ μιας επιχείρησης.

Στη συγκεκριμένη πτυχιακή θα ασχοληθούμε με την χρήση του πιο δημοφιλούς συστήματος διαχείριση περιεχομένου, του Joomla, για να κατασκευάσουμε την ιστοσελίδα μας.

## Κεφάλαιο 2 Σύστημα διαχείρισης περιεχομένου

### 2.1 Ορισμός CMS

Παρά την μεγάλη εξάπλωση του διαδικτύου και τον ολοένα αυξανόμενο αριθμό ιστοσελίδων, πολλές εταιρείες ή χρήστες δεν είναι εξοικειωμένοι με την τεχνολογία αυτή. Ας δούμε λοιπόν πιο αναλυτικά τι ακριβώς είναι ένα Σύστημα Διαχείρισης Περιεχομένου (Content Management System ή CMS) και ποιος ο ρόλος του για την επιτυχία ή όχι μιας ιστοσελίδας.

Ο όρος Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems ή CMS), αναφέρεται στις εφαρμογές που επιτρέπουν τη δημιουργία, διαχείριση, διανομή, δημοσίευση και τον εντοπισμό της πληροφορίας, τη δυνατότητα να διαχειριστείτε τη δομή του site σας, την εμφάνιση των δημοσιευμένων σελίδων και την πλοήγηση που προσφέρεται στους επισκέπτες και χρήστες της.

Είναι αυτό που αποκαλούμε πολλές φορές "**δυναμικό περιεχόμενο**" σε ένα website και δεν είναι άλλο παρά οι πληροφορίες που παρουσιάζονται στην ιστοσελίδα και μπορούν να δημιουργηθούν και ν' αλλάξουν οποιαδήποτε στιγμή από τους ίδιους τους διαχειριστές του χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ή επεξεργασία ιστοσελίδων, καθώς τα περιεχόμενα συντάσσονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") HTML editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word.

Αυτό σημαίνει ότι δεν χρειάζεται να δημιουργηθούν πολλές ξεχωριστές ιστοσελίδες για την παρουσίαση των πληροφοριών στο δικτυακό τόπο, αλλά αρκεί ένας ενιαίος σχεδιασμός στα σημεία όπου θέλουμε να εμφανίζεται το περιεχόμενό μας, καθώς και να υπάρχει ο ειδικός σε κάποια συγκεκριμένη γλώσσα προγραμματισμού (ASP, PHP, ColdFusion, Perl, CGI κ.λπ.), ο οποίος αναλαμβάνει να εμφανίσει τις σωστές πληροφορίες στις σωστές θέσεις.

Έτσι, για το δικτυακό τόπο μιας εφημερίδας π.χ., που απαιτεί εύλογα καθημερινή ενημέρωση αλλά δεν χρησιμοποιεί κάποιο Σύστημα Content Management, θα πρέπει ο υπεύθυνος για το σχεδιασμό του (designer) να δημιουργήσει μια σελίδα με τα γραφικά, την πλοήγηση και το περιβάλλον διεπαφής (interface) του website, ο υπεύθυνος ύλης να τοποθετήσει το περιεχόμενο στα σημεία της ιστοσελίδας που θέλει, και να ενημερωθούν οι σύνδεσμοι των υπόλοιπων σελίδων ώστε να συνδέονται με την καινούργια. Αφού την αποθηκεύσει, πρέπει να την ανεβάσει στο website μαζί με τις υπόλοιπες ιστοσελίδες που άλλαξαν.

Αντιθέτως, αν ο δικτυακός τόπος λειτουργεί με χρήση κάποιου Συστήματος Content Management, το μόνο που έχει να κάνει ο διαχειριστής του είναι να ανοίξει τη σχετική φόρμα εισαγωγής νέου άρθρου στη διαχειριστική εφαρμογή της ιστοσελίδας και να γράψει ή να επικολλήσει (copy-paste) τα στοιχεία που επιθυμεί.



Αυτόματα, μετά την καταχώριση γίνονται από το ίδιο το σύστημα διαχείρισης περιεχομένου όλες οι απαραίτητες ενέργειες, ώστε το άρθρο να είναι άμεσα διαθέσιμο στους επισκέπτες και όλοι οι σύνδεσμοι προς αυτό ενημερωμένοι.

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

## 2.2 Ανατομία ενός CMS

Η λειτουργικότητα ενός συστήματος διαχείρισης περιεχομένου, μπορεί να αναλυθεί στις εξής βασικές κατηγορίες:

- ✓ δημιουργία περιεχομένου
- ✓ διαχείριση περιεχομένου
- ✓ δημοσίευση
- ✓ παρουσίαση

### 2.2.1 Δημιουργία περιεχομένου

Το πρώτο τμήμα ενός Συστήματος Διαχείρισης Περιεχομένου, είναι ένα εύχρηστο περιβάλλον συγγραφής, σχεδιασμένο να λειτουργεί όπως οι βασικές λειτουργίες του Word. Έτσι, ακόμη και οι χρήστες χωρίς ιδιαίτερη τεχνική γνώση, μπορούν να δημιουργούν νέες σελίδες ή να ενημερώνουν το περιεχόμενο παλαιών, χωρίς να χρειάζεται να γνωρίζουν HTML ή κάποια άλλη γλώσσα δημιουργίας ιστοσελίδων.

Επίσης, το CMS, επιτρέπει τη διαχείριση της δομής του site. Δηλαδή που θα τοποθετηθούν οι νέες σελίδες και τον τρόπο αλληλοσύνδεσης τους. Κάποια συστήματα επιτρέπουν την εύκολη αναδιάρθρωση του site χωρίς να δημιουργούνται "σπασμένοι" σύνδεσμοι.

Σχεδόν όλα τα συστήματα CMS, προσφέρουν ένα εύχρηστο περιβάλλον επεξεργασίας κειμένου μέσα από τον browser σας, γεγονός που απλοποιεί ακόμη περισσότερο την υλοποίηση του συστήματος και επιτρέπει την απομακρυσμένη ενημέρωση του περιεχομένου και του site.

Αυτή ακριβώς η δυνατότητα αποτελεί ένα από τα κλειδιά της επιτυχίας του CMS. Παρέχοντας έναν απλό και εύχρηστο μηχανισμό συντήρησης του site, η δημιουργία του περιεχομένου και των σελίδων μπορεί να ενσωματωθεί στη διαδικασία της καθημερινής λειτουργίας της επιχείρησης. Για παράδειγμα, ο υπεύθυνος προώθησης συντηρεί και ενημερώνει το τμήμα των δελτίων τύπου, ενώ ο υπεύθυνος των προϊόντων ενημερώνει τον προϊοντικό κατάλογο σας.

## 2.2.2 Διαχείριση Περιεχομένου

Αυτή η κατηγορία λειτουργιών προσφέρει μία πλειάδα χρήσιμων χαρακτηριστικών όπως:

- 1) Παρακολούθηση όλων των εκδόσεων μιας σελίδας καθώς και ποιος έκανε τις αλλαγές και πότε.
- 2) Διασφάλιση ότι οι αλλαγές σε κάθε τμήμα του site, μπορούν να γίνουν μόνο από τα άτομα που έχουν τα ανάλογα δικαιώματα αλλαγών στο συγκεκριμένο τμήμα.
- 3) Δυνατότητες ελέγχου της ροής της εργασίας. Για παράδειγμα, όταν δημιουργείται ή επεξεργάζεται από τον συντάκτη μια σελίδα, αποθηκεύονται δεδομένα όπως ημερομηνίες δημιουργίας, αλλαγής, δημοσίευσης και αρχειοθέτησης καθώς και άλλες πληροφορίες ελέγχου που δίνουν δυνατότητα ελέγχου της κατάστασης του περιεχομένου και της σελίδας.

## 2.2.3 Δημοσίευση

Όταν ολοκληρωθεί η δημιουργία του τελικού κειμένου, αυτό μπορεί να δημοσιευθεί στο site μας. Τα συστήματα διαχείρισης περιεχομένου διαθέτουν ισχυρούς μηχανισμούς δημοσίευσης, οι οποίοι επιτρέπουν την αυτόματη μορφοποίηση της σελίδας κατά τη δημοσίευση της. Μπορούν επίσης να επιτρέψουν την ταυτόχρονη δημοσίευση του ίδιου περιεχομένου σε πολλαπλά site.

Φυσικά, κάθε site έχει διαφορετική εμφάνιση. Έτσι το CMS επιτρέπει στους γραφίστες και web designers να καθορίσουν την εμφάνιση που θα χρησιμοποιηθεί από το σύστημα. Οι δυνατότητες δημοσίευσης διασφαλίζουν τη σταθερή και ομοιόμορφη εμφάνιση των σελίδων σε όλο το site, διαθέτοντας υψηλά επίπεδα εμφάνισης.

Τέλος, επιτρέπει στους συντάκτες του περιεχομένου, να επικεντρωθούν στην συγγραφή της πληροφορίας, αφήνοντας στο σύστημα τη φροντίδα της παρουσίασης της.

## 2.2.4 Παρουσίαση

Το CMS θα φροντίσει για την πλοήγηση στο site, δημιουργώντας τα απαραίτητα links ανάλογα με το υπάρχον περιεχόμενο.

Επίσης, διευκολύνει την υποστήριξη διαφορετικών browsers ή και χρήστες με δυσκολίες. Το Σύστημα Διαχείρισης Περιεχομένου μπορεί να χρησιμοποιηθεί για να κάνει το site σας δυναμικό και διαδραστικό, αυξάνοντας κατακόρυφα την αποτελεσματικότητά του.

Αν και δημιουργία και παρουσίαση περιεχομένου σε μορφή HTML είναι ένα από τα δυνατότερα σημεία ενός CMS, εντούτοις μπορεί να προσφέρει πολύ περισσότερα. Μια εξαιρετικά ενδιαφέρουσα δυνατότητα των συστημάτων αυτών είναι η ταυτόχρονη δημοσίευση του περιεχομένου σε διαφορετικά format για εκτύπωση και ηλεκτρονική διανομή (PDF, Word, κλπ), ασύρματες/φορητές συσκευές (Wap, κλπ), RSS Feeds ή XML.

## 2.3 Οφέλη και πλεονεκτήματα

Τα οφέλη που προκύπτουν από την εφαρμογή και χρήση ενός συστήματος διαχείρισης περιεχομένου είναι πολλά και σημαντικά:

- ✓ **Γρήγορη ενημέρωση, διαχείριση και αρχειοθέτηση της πληροφορίας**

Η δυνατότητα αναζήτησης του περιεχομένου που καταχωρείται και η αυτόματη αρχειοθέτηση του, η εύκολη χρήση και η άμεση γνώση του τελικού αποτελέσματος, όπως γίνεται με τους γνωστούς κειμενογράφους, παρέχει μεγαλύτερη συνέπεια στην εμφάνιση του διαδικτυακού σας τόπου.

- ✓ **Αυξημένη ευελιξία, βελτιωμένη πλοήγηση και αυξημένη δυνατότητα επέκτασης**

Ο διαχωρισμός του περιεχομένου από το σχεδιασμό και την πλοήγηση (navigation) του δικτυακού τόπου δίνει την δυνατότητα αλλαγής του σχεδιασμού ή τρόπου πλοήγησης χωρίς να είναι απαραίτητη η ενημέρωση όλων των σελίδων από τον ίδιο το χρήστη.

- ✓ **Υποστήριξη αποκεντροποιημένου συστήματος ενημέρωσης**

Η ενημέρωση του περιεχομένου μπορεί να γίνει από οπουδήποτε, από πολλούς χρήστες και από διαφορετικούς υπολογιστές.

- ✓ **Αυξημένη ασφάλεια**

Η ασφάλεια και η προστασία του σχεδιασμού του site από λανθασμένες ενέργειες, που θα μπορούσαν να δημιουργήσουν προβλήματα στην εμφάνισή του και η αυτόματη δημιουργία των συνδέσμων μεταξύ των σελίδων καθώς και η αποφυγή προβλημάτων ανύπαρκτων σελίδων (404 error pages) δίνει την δυνατότητα απρόσκοπτης λειτουργίας στην ιστοσελίδα.

- ✓ **Μείωση της επανάληψης της πληροφορίας**

Η μη επαναλαμβανόμενη πληροφορία ή η μείωση αυτής, σημαίνει πρακτικά μικρότερος φόρτος στον εξυπηρετητή (server) και χρήση λιγότερου χώρου, αφού δεν υπάρχουν πολλές επαναλαμβανόμενες στατικές σελίδες, από τη στιγμή που η ανάπτυξη των σελίδων γίνεται δυναμικά. Όλο το περιεχόμενο καταχωρείται στην βάση δεδομένων, την οποία μπορούμε πιο εύκολα και γρήγορα να την προστατεύσουμε τηρώντας αντίγραφα ασφαλείας.

- ✓ **Μειωμένο κόστος συντήρησης**

Δεν απαιτούνται ειδικές τεχνικές γνώσεις από τους διαχειριστές με συνέπεια την ελαχιστοποίηση λήψης τεχνικής υποστήριξης.

## 2.4 Τα δημοφιλέστερα CMS



Το **WordPress** είναι ένα σύγχρονο σύστημα διαδικτυακών δημοσιεύσεων και διαχείρισης περιεχομένου μ' άλλα λόγια, ένα μέσο για να φτιάξετε έναν ιστότοπο (site) ή ιστολόγιο (blog) ωραία και απλά και να δημοσιεύετε κάθε είδους περιεχόμενο: άρθρα, σκέψεις, φωτογραφίες, βίντεο, πληροφορίες για προϊόντα και υπηρεσίες – ό,τι θέλετε! Το WordPress διατίθεται δωρεάν και είναι ελεύθερο λογισμικό ανοιχτού κώδικα.



Το **Drupal** είναι ένα ανοιχτού κώδικα Σύστημα διαχείρισης περιεχομένου (CMS). Χρησιμοποιείται για την δημιουργία και την διαχείριση πολλών και διαφορετικών ιστότοπων. Η σταθερότητα, ευελιξία και η ασφάλεια του είναι μερικά απ τα σημεία αναφοράς του.

A promotional banner for Joomla! 1.7. The banner has a dark blue background. On the left, there is a small screenshot of the Joomla! user interface. In the center, the Joomla! logo (four interlocking rings in green, orange, red, and blue) is followed by the text "Joomla!" in white. To the right of the logo is a blue square with the number "1.7" in white. Below the logo and version number, the text "A True Open Source Community." is written in white. Underneath that, a paragraph of text describes Joomla! as the world's most popular open source CMS, used by individuals, small businesses, and large organizations. At the bottom of the banner, there are two buttons: a yellow "Download" button and a blue "Demo" button. A link "Learn more about 1.7 ..." is also present.

**Joomla!** 1.7

**A True Open Source Community.**

Joomla is the world's most popular open source CMS (content management system). With 2.7 percent of the Web running on Joomla, the software is used by individuals, small & medium-sized businesses, and large organizations worldwide to easily create & build a variety of websites & web-enabled applications.

[Learn more about 1.7 ...](#)

**Download** **Demo**

Γεμάτο δυνατότητες αλλά και ταυτόχρονα εξαιρετικά ευέλικτο και φιλικό, το **Joomla!** είναι η εφαρμογή που χρειάζεστε για να δημοσιεύσετε στο διαδίκτυο μια προσωπική ιστοσελίδα, αλλά και έναν ολόκληρο εταιρικό δικτυακό τόπο. Είναι προσαρμόσιμο σε περιβάλλοντα επιχειρηματικής κλίμακας όπως τα intranets μεγάλων επιχειρήσεων ή οργανισμών και χρησιμοποιείται από το 2% των ιστοχώρων παγκοσμίως κατέχοντας το 11% στην αγορά των CMS. Με αυτό το σύστημα διαχείρισης περιεχομένου θα ασχοληθούμε και εμείς.

## Κεφάλαιο 3 Παρουσίαση Joomla

### 3.1 Ορισμός Joomla

**Joomla!™** είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου το οποίο έχει χαρακτηριστεί από τους δεκάδες χιλιάδες χρήστες του ως το καλύτερο CMS στον κόσμο, διότι έχει τεράστιες δυνατότητες αλλά και ταυτόχρονα εξαιρετικά ευέλικτο και φιλικό περιβάλλον εργασίας και διαχείρισης.

Μόλις εγκατασταθεί, ο διαχειριστής του δικτυακού τόπου μπορεί να συνδεθεί με την backend κονσόλα διαχείρισης και να δημιουργήσει τμήματα, κατηγορίες, άρθρα με περιεχόμενο, δημοσκοπήσεις και άλλα πολλά! Διαχείριση ολόκληρου του frontend της ιστοσελίδας από ένα εύρηστο περιβάλλον διαχείρισης.

Όταν δημιουργείται ένα νέο περιεχόμενο, ένας **WYSIWYG (What You See Is What You Get)** επεξεργαστής κειμένου που επιτρέπει απλές αλλαγές χωρίς τη γνώση της **HTML**.

Παρόλο ότι δεν είναι απαραίτητο, ο χρήστης να έχει γνώση της **HTML** και **CSS**, συνιστάται να αρχίσει τη μελέτη των βασικών εντολών της **HTML** και **CSS** για την περίπτωση που θα πρέπει να επεξεργαστεί ή να τροποποιήσει κάποιο στοιχείο ώστε να το προσαρμόσει στις ανάγκες που θα δημιουργηθούν. Στις κοινότητες φόρουμ για Joomla είναι διαθέσιμοι χιλιάδες προγραμματιστές και σχεδιαστές για να βοηθήσουν νέους και έμπειρους χρήστες, εφόσον είναι δυνατό.

**Το Joomla!™ είναι μία δωρεάν εφαρμογή**, ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου, το οποίο έχει συνταχθεί στην γλώσσα προγραμματισμού **PHP** για τη διαχείριση και τη δημοσίευση περιεχομένου στον ιστό (web) και ενδοδικτύων (intranets) χρησιμοποιώντας μια **MySQL** βάση δεδομένων. Το Joomla περιλαμβάνει χαρακτηριστικά όπως η cache των σελίδων για τη βελτίωση των επιδόσεων, web ευρετηρίασης (indexing), **RSS feeds**, εκτυπώσιμες (printable) εκδόσεις των σελίδων, newsflashes, blogs, forums, δημοσκοπήσεις, ημερολόγια, αναζήτηση στην ιστοσελίδα, την ανάλογη γλώσσα αλλά και την δυνατότητα πολυγλωσσικών ιστοσελίδων.

**Το όνομα "Joomla"** είναι μια φωνητική γραφή της γλώσσας **Σουαχίλι (Swahili)** στην οποία η λέξη "**jumla**" σημαίνει "**όλοι μαζί**" ή "**ως σύνολο**". Επέλεξαν αυτό το όνομα για να αντικατοπτρίζει τη δέσμευση της Κοινότητας και την ομάδα ανάπτυξης του έργου. Η πρώτη έκδοση του Joomla (Joomla 1.0.0) ανακοινώθηκε στις 16 Σεπτεμβρίου 2005. Αυτή ήταν μια νέα ονομασία της έκδοσης Mambo 4.5.2.3 σε συνδυασμό με διορθώσεις κάποιων σφαλμάτων (bug) στο περιβάλλον διαχείρισης και στον κώδικα της ασφαλείας. Στη συνέχεια, για την πορεία ανάπτυξης του project, οι προγραμματιστές του πυρήνα προβλέπουν ότι το **Joomla! 2.0** θα είναι ένα εντελώς ανανεωμένο και ξαναγραμμένο σε νέο κώδικα βασισμένο σε **PHP 5.0**.

## 3.2 Χαρακτηριστικά Joomla

Το Joomla! έχει τόσα πολλά χαρακτηριστικά γνωρίσματα που είναι δύσκολο να αναφερθούμε σε όλα. Επιγραμματικά, αναφέρουμε τις πιο δημοφιλείς δυνατότητες :

- ✓ Μπορείτε να προσθέσετε περιεχόμενο στον ιστότοπό σας από οποιονδήποτε υπολογιστή διαθέτει σύνδεση στο διαδίκτυο. Πληκτρολογήστε το κείμενο, ανεβάστε τις φωτογραφίες, δημοσιεύστε!
- ✓ Χρησιμοποιήστε το Joomla! για να συνεργαστείτε με όσους μπορούν να συμβάλλουν στην επεξεργασία του περιεχομένου. Δώστε τη δυνατότητα σε συνεργάτες σας να δημοσιεύσουν περιεχόμενο, στο πλαίσιο ασφαλώς των αρμοδιοτήτων τους. Είναι εύκολο. Μπορούν να το κάνουν από κάθε υπολογιστή με σύνδεση στο διαδίκτυο.
- ✓ Δημοσιεύστε απεριόριστες σελίδες, χωρίς να περιορίζεστε από το Joomla! Χωρίς κανένα πρόβλημα, μπορείτε να κάνετε αναζητήσεις περιεχομένου και να τις αρχειοθετήσετε.
- ✓ Η εφαρμογή υποστηρίζει τα διαφημιστικά banners: προωθήστε δικά σας προϊόντα και υπηρεσίες, ή χρησιμοποιήστε τα ως διαφημιστικό μέσο για τρίτους.
- ✓ Προσθέστε forum, photo galleries, βιβλιοθήκες αρχείων, βιβλία επισκεπτών και φόρμες επικοινωνίας: λίγες από τις εκατοντάδες πρόσθετες εφαρμογές που κυκλοφορούν για το Joomla!

## 3.3 Απαιτήσεις συστήματος για Joomla

Το Joomla! τρέχει σε προ-επεξεργαστή PHP. Το **PHP** βγαίνει σε διάφορες "γεύσεις" για πολλά συστήματα χειρισμού. Εκτός από το PHP θα χρειαστείτε έναν διακομιστή ιστοσελίδων. Το Joomla! βελτιστοποιείται για τον διακομιστή ιστοσελίδων **Apache**, αλλά μπορεί να τρέξει σε διαφορετικούς διακομιστές ιστοσελίδων όπως Microsoft IIS, απλά απαιτείται ένας επιπλέον συντονισμός PHP και MySQL. Το Joomla! βασίζεται επίσης σε μια βάση δεδομένων, γι' αυτό μπορείτε τώρα να χρησιμοποιήσετε μόνο **MySQL**.

Πολλοί άνθρωποι ξέρουν από προσωπική τους εμπειρία ότι δεν είναι τόσο εύκολο να εγκαταστήσεις έναν διακομιστή ιστοσελίδων Apache και γίνεται ακόμη πιο δύσκολο αν θέλετε να προσθέσετε τα MySQL, PHP και Perl. Τα XAMPP, WAMP, και

MAMP εύκολα εγκαθιστούν κατανομές που περιλαμβάνουν Apache, MySQL, PHP και Perl για τα Windows, και τα συστήματα χειρισμού Mac OSX και Linux.

Αυτά τα πακέτα είναι για εγκαταστάσεις τοπικών διακομιστών σε μη δημόσιους servers μόνο.

Οι ελάχιστες απαιτήσεις της έκδοσης είναι:

- ✓ **Apache 1.x ή 2.x**
- ✓ **PHP 4.3 ή και παραπάνω**
- ✓ **MySQL 3.23 ή και παραπάνω**

### 3.4 Apache Web Server



Ο Apache Web Server είναι αυτό ακριβώς που δηλώνει το όνομά του. Πρόκειται δηλαδή για έναν εξυπηρετητή (server) του παγκόσμιου Ιστού (Web). Με τον όρο server το μυαλό μας πηγαίνει ίσως σε ηλεκτρονικούς υπολογιστές που φιλοξενούν ιστοσελίδες και όχι άδικα.

Ο όρος αυτός χρησιμοποιείται και για το μηχάνημα εξυπηρετητή (hardware) αλλά και για το πρόγραμμα (software). Στο άρθρο αυτό θα ασχοληθούμε μόνο με το software και συγκεκριμένα με τον Apache.

Ο Apache εγκαθίσταται σε έναν υπολογιστή ο οποίος μπορεί να χρησιμοποιεί διάφορα λειτουργικά συστήματα όπως Linux, Unix, Microsoft Windows, GNU, FreeBSD, Solaris, Novell NetWare, Mac OS X, OS/2, TPF. Ο ρόλος του Apache είναι να αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες (clients) όπως είναι ένας ο φυλλομετρητής (browser) ενός χρήστη και στη συνέχεια να εξυπηρετεί αυτές τις αιτήσεις “σερβίροντας” τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης (URL), είτε μέσω ενός συνδέσμου (link). Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο HTTP (Hypertext Transfer Protocol).



### **3.4.1 Το ξεκίνημα και η εξέλιξη του Apache**

Η περίοδος έναρξης της δημιουργίας του προγράμματος χρονολογείται στις αρχές του 1990, όταν άρχισε να αναπτύσσεται από τον Robert McCool, ως ένα project του National Center for Supercomputing Applications (NCSA) με το όνομα HTTPd (HTTP daemon). Το 1994 ο Robert McCool αποχώρησε από το NCSA με αποτέλεσμα το NCSA HTTPd να μείνει σχεδόν εγκαταλειμμένο, πέρα από κάποιες διορθώσεις (patches) που ανέπτυσαν και διένειμαν εκτός από τον McCool και άλλοι προγραμματιστές. Το 1995 ανέλαβε το πρόγραμμα το Ίδρυμα Λογισμικού Apache (Apache Software Foundation), το οποίο διατηρεί την εποπτεία του έως και σήμερα.

### **3.4.2 Ο Apache σήμερα**

Ο Apache HTTP αναπτύσσεται από την “Κοινότητα Ανοιχτού Λογισμικού” και η εποπτεία, υποστήριξη, και διάθεση του προγράμματος γίνεται από το Apache Software Foundation. Το πρόγραμμα είναι ανοιχτού κώδικα (open source), κάτι που σημαίνει ότι σύμφωνα με την άδεια χρήσης του (license), διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα από το χρήστη προσθήκες και τροποποιήσεις στον κώδικα του.

### **3.4.3 Η προέλευση του ονόματος του**

Υπάρχουν δύο εκδοχές σχετικά με την προέλευση του ονόματος του. Η πρώτη εντοπίζεται στα πρώτα χρόνια της δημιουργίας του, τότε που ως NCSA HTTPd έπρεπε να αναπτύσσονται συνεχώς διορθώσεις (patches) για να ενσωματωθούν στον αρχικό του κώδικα με αποτέλεσμα να του δοθεί το όνομα a patchy server. Η δεύτερη εκδοχή σύμφωνα με το ίδρυμα Apache, αναφέρει ότι το όνομα αυτό δόθηκε προς τιμήν των ιθαγενών Ινδιάνων της Αμερικής και συμβολίζει το μαχητικό πνεύμα και την αντοχή.

### **3.4.4 Χαρακτηριστικά και λειτουργίες του Apache HTTP**

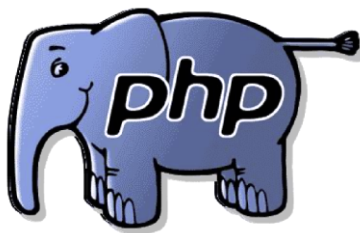
Ο Apache διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Οι δυνατότητες του προγράμματος αυτού καθαυτού και τα χαρακτηριστικά του δεν είναι και τόσο πολλά. Ένα από τα βασικότερα χαρακτηριστικά του όμως, το οποίο και του δίνει μεγάλες δυνατότητες, είναι ότι μπορεί να προσαρμόσει επάνω του πολλές προσθήκες προγραμμάτων (modules), τα οποία με τη σειρά τους παρέχουν διαφορετικές λειτουργίες. Μερικά από τα πιο γνωστά modules του Apache HTTP είναι τα modules πιστοποίησης, όπως για παράδειγμα τα mod\_access, mod\_auth, mod\_digest κ.λπ. Παρέχει επίσης SSL σε

TLS μέσω των (mod\_ssl), και proxy module (mod\_proxy), πραγματοποιεί ανακατευθύνσεις διευθύνσεων (URL rewrites) μέσω του mod\_rewrite, καταγραφές συνδέσεων μέσω του mod\_log\_config, συμπίεση αρχείων μέσω του mod\_gzip και πολλά άλλα modules τα οποία διατίθενται είτε απ'ο το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό – δυνατότητα του Apache HTTP, όπως έχω αναφέρω πιο πάνω, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl, Python κ.λπ.

Αυτά είναι μερικά από τα χαρακτηριστικά και τις λειτουργίες του που κάνουν τον Apache τον πιο δημοφιλή Web Server από το 1996 έως τις μέρες μας. Περισσότερο από το 50% των ιστοχώρων του παγκόσμιου ιστού, χρησιμοποιεί τον Apache ως εξυπηρετητή. Το υπόλοιπο ποσοστό καλύπτουν αντίστοιχα προγράμματα, όπως το Microsoft Internet Information Services (IIS), ο Sun Java System Web Server, ο Zeus Web Server κα.

### 3.5 PHP



Η PHP είναι μια γλώσσα "server-side". Αυτό σημαίνει ότι ο κώδικας PHP που περιέχει μια σελίδα εκτελείται στον server (όπου είναι αποθηκευμένη η σελίδα), ενώ τα αποτελέσματα εμφανίζονται με μορφή HTML στον τελικό χρήστη.

Η PHP αρχικά σήμαινε Personal Home Page ( προσωπική αρχική σελίδα ), αλλά άλλαξε σύμφωνα με την σύμβαση GNU και τώρα σημαίνει PHP Hypertext Preprocessor ( προεπεξεργαστής κειμένου PHP ).

Δημιουργήθηκε το 1994 και ήταν αρχικά η δουλειά ενός ατόμου , του Rasmus Lerdorf. Υιοθετήθηκε και από άλλα ταλαντούχα άτομα και έχει περάσει από τρεις βασικές εκδόσεις . Τον Ιανουάριο του 2001 ήταν σε χρήση σχεδόν σε πέντε εκατομμύρια τομείς παγκόσμια και αυτός ο αριθμός μεγαλώνει γρήγορα. Η PHP είναι ένα προϊόν ανοιχτού κώδικα. Θα έχετε πρόσβαση στον κώδικα προέλευσης. Μπορείτε να τον χρησιμοποιήσετε , να τον αλλάξετε και να τον αναδιανεμίετε , χωρίς χρέωση.

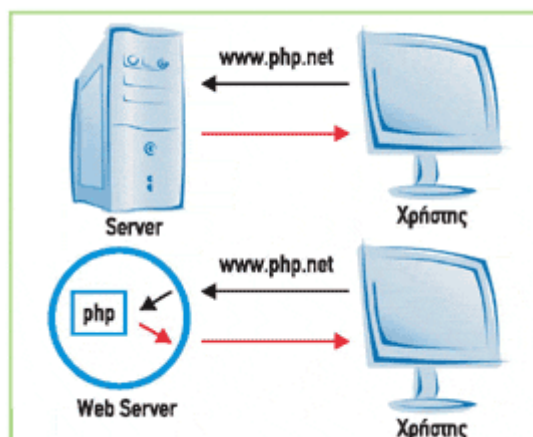
### 3.5.1 Χρήση PHP

Μια σελίδα στο Διαδίκτυο πρέπει συχνά να ανανεώνει το υλικό της, ώστε να διατηρεί την επισκεψιμότητά της. Οι διαδικασίες ανανέωσης αποτελούν ένα καθημερινό πρόβλημα για τους webmasters, μιας και πρέπει να επεξεργάζονται κάθε αρχείο της σελίδας, προσθέτοντας σε αυτό νέες πληροφορίες, και στο τέλος να το αποθηκεύουν πίσω στον server. Αυτή η δουλειά μπορεί να φαίνεται σχετικά εύκολη, αλλά φανταστείτε το site μας να έχει πάνω από 100 σελίδες και να θέλουμε να ανανεώσουμε κάθε μία από αυτές σε εβδομαδιαία βάση.

Με τη βοήθεια της PHP μπορεί να δημιουργηθεί εύκολα ένα σύστημα διαχείρισης δεδομένων (content management system) για την εύχρηστη ανανέωση του περιεχομένου μιας σελίδας. Απλές φόρμες σε HTML μεταφέρουν το πληροφοριακό υλικό στον server, όπου κατόπιν αποθηκεύεται σε αρχειοθετημένους χώρους (databases ή βάσεις δεδομένων), ένα σύστημα ανάγνωσης δεδομένων αναλαμβάνει την εμφάνιση του υλικού στη σελίδα, ενώ μια ειδική εφαρμογή επιτρέπει και την επεξεργασία των ήδη αποθηκευμένων πληροφοριών.

Με ένα φθινό διακομιστή μπορούμε να εξυπηρετήσουμε εκατομμύρια επισκέψεων καθημερινά. Σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες, που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. Μπορείτε να δημιουργήσετε εικόνες GIF δυναμικά, να συνδεθείτε με άλλες υπηρεσίες δικτύων, να στείλετε ηλεκτρονικό ταχυδρομείο, να δουλέψετε με cookies και να δημιουργήσετε PDF έγγραφα: όλα αυτά με λίγες γραμμές κώδικα.

Το σχεδιάγραμμα που ακολουθεί παρουσιάζει αναλυτικά τον τρόπο με τον οποίο δουλεύει ένας web server (απαραίτητο λογισμικό για την επεξεργασία και τη λειτουργία μιας ιστοσελίδας) στον οποίο υπάρχει εγκατεστημένη η PHP. Ο χρήστης "καλεί" μια σελίδα και ο server κάνει τις αντίστοιχες διεργασίες, για να παρουσιάσει το επιθυμητό αποτέλεσμα πίσω στο χρήστη. Μια απλή σελίδα HTML παρακάμπτει το εγκατεστημένο λογισμικό της PHP στον web server και εμφανίζεται όπως ακριβώς είναι στο χρήστη.



### 3.5.2 Πλεονεκτήματα PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Microsoft Silverlight.

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως :

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερσιμότητα
- Διαθεσιμότητα του κώδικα προέλευσης

### 3.6 MySQL



Η MySQL είναι ένα πολύ γρήγορο και δυνατό, σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων σας επιτρέπει να αποθηκεύετε, να αναζητάτε, να ταξινομείτε και να ανακαλείτε τα δεδομένα αποτελεσματικά.

Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα σας, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια. Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινά από το 1979.

### **3.6.1 Πλεονεκτήματα της MySQL**

Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL , Microsoft SQL και Oracle. Η MySQL έχει πολλά πλεονεκτήματα , όπως χαμηλό κόστος , εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος.

#### **Απόδοση**

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Μπορείτε να δείτε την σελίδα δοκιμών <http://web.mysql.com/benchmark.html> . Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

#### **Χαμηλό κόστος**

Η MySQL είναι διαθέσιμη δωρεάν , με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος , αν πάρετε εμπορική άδεια, αν απαιτείται από την εφαρμογή σας.

#### **Ευκολία Χρήσης**

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν έχετε χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων δεν θα έχετε πρόβλημα να προσαρμοστείτε σε αυτό.

#### **Μεταφερσιμότητα**

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα Unix όπως επίσης και στα Microsoft Windows .

#### **Κώδικας Προέλευσης**

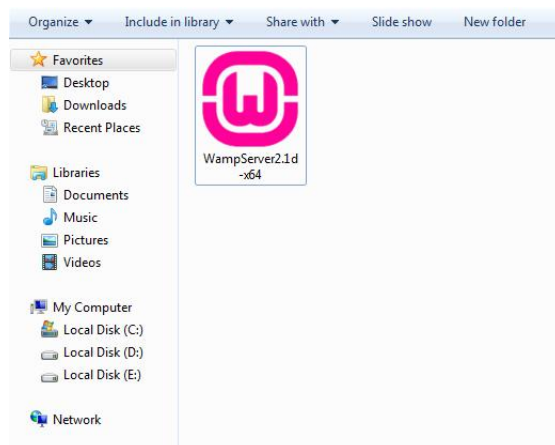
Όπως και με την PHP , μπορείτε να πάρετε και να τροποποιήσετε τον κώδικα προέλευσης της MySQL.

## Κεφάλαιο 4 Εγκατάσταση Joomla

### 4.1 Εγκατάσταση Wamp

Για να μπορέσουμε να μετατρέψουμε τον υπολογιστή μας σε εξυπηρετητή(**server**), πρέπει να εγκαταστήσουμε το **Wamp** με τη βοήθεια του οποίου γίνεται εύκολη η εισαγωγή των **Server Apache**, **PHP** (γλώσσα προγραμματισμού) , **MySQL** (βάση δεδομένων) και **PhpMyadmin**(διαχείριση βάσης δεδομένων) χωρίς να χρειάζεται καμία παραμετροποίηση από εμάς. Η εγκατάσταση αυτή περιγράφεται παρακάτω και είναι απαραίτητη για να μπορέσουμε στη συνέχεια να εγκαταστήσουμε το Joomla.

Κατεβάζουμε το Wamp από <http://www.wampserver.com/en/download.php>



το εκτελούμε και ακολουθούμε τα απλά βήματα του οδηγού που εμφανίζεται.



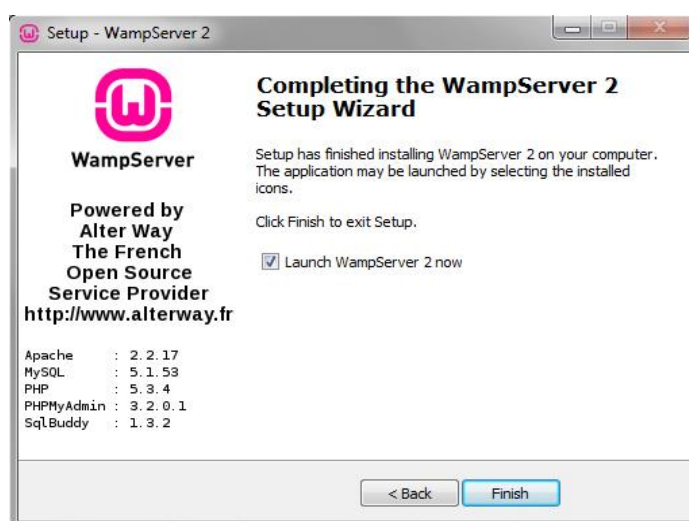
Πριν την ολοκλήρωση της εγκατάστασης μας εμφανίζεται το παράθυρο συμπλήρωσης των **SMTP** και **Email** παραμέτρων το οποίο δεν έχει ιδιαίτερη σημασία σε τοπικό επίπεδο που βρισκόμαστε. Εμείς βέβαια συμπληρώνουμε :


- **SMTP: localhost**
- Email: τη διεύθυνση του ηλεκτρονικού μας ταχυδρομείου.

Και παράλληλα εμφανίζεται και μια ειδοποίηση-ερώτηση από το Firewall των Windows αν θέλουμε να επιτραπεί η πρόσβαση στον Apache HTTP Server στην οποία απαντάμε θετικά.

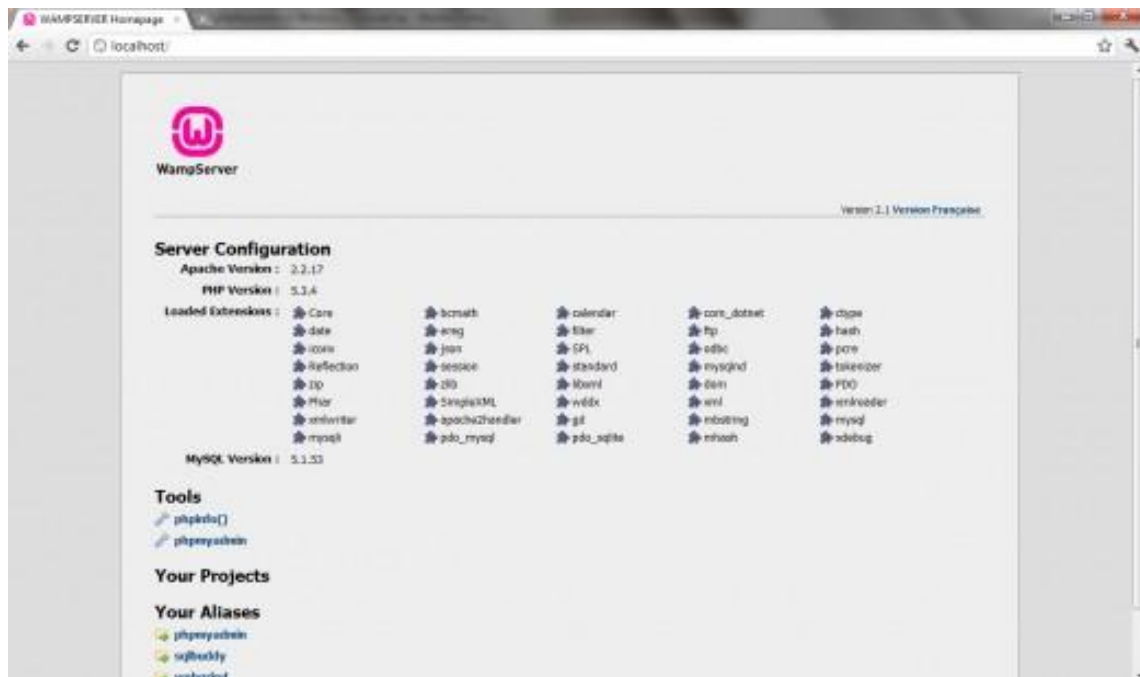


και πατώντας Finish(ολοκλήρωση) στο τελευταίο παράθυρο που μας εμφανίζεται , έχουμε πλέον ολοκληρώσει με επιτυχία την εγκατάσταση του WAMP.



Όταν ανοίξουμε το Wamp Server πρέπει να παρατηρήσουμε κάτω δεξιά στο taskbar το χαρακτηριστικό εικονίδιο  να έχει χρώμα πράσινο, κάτι που θα σημαίνει αυτομάτως πως όλες οι υπηρεσίες λειτουργούν σωστά.

Αυτό το πιστοποιούμε τελικά ανοίγοντας ένα browser της επιλογής μας πληκτρολογούμε **http://127.0.0.1/** ή **http://localhost/** και πατάμε Enter. Το αποτέλεσμα θα πρέπει να είναι το παρακάτω :



Αυτή είναι η οθόνη καλωσορίσματος του Wamp , που μας παρέχει πληροφορίες για τις εκδόσεις των Php, MySQL και Apache, μας παρουσιάζει τα πρόσθετα που έχουν εγκατασταθεί όπως επίσης μας δίνει και σύνδεσμο(link) πρόσβασης στο εργαλείο διαχείρισης της βάσης δεδομένων μας PhpMyadmin.

- Apache Version: 2.2.17
- PHP Version: 5.3.4
- Loaded Extensions: Loaded extensions of PHP
- MySql Version: 5.1.53 – It will be Mysql 5.5.8 at 32 bits platform.



## 4.2 Οδηγός Εγκατάστασης Joomla

Αφού τελειώσουμε με τη λήψη, αποσυμπίεση και την αντιγραφή όλων των αρχείων του Joomla μέσα στο φάκελο C:\wamp\www ( του αρχείου Joomla που έχουμε κατεβάσει) είμαστε έτοιμοι να ξεκινήσουμε την εγκατάσταση του Joomla. Η εγκατάσταση του Joomla είναι μία απλή διαδικασία και ολοκληρώνεται μέσα από 7 βήματα.

Ανοίγουμε τον **Internet Explorer**.

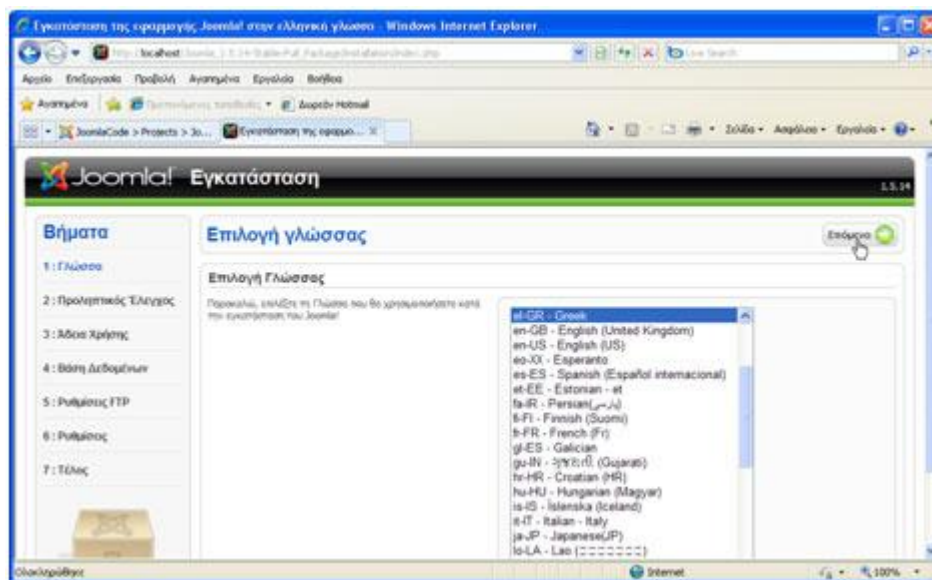
Στη γραμμή διευθύνσεων πληκτρολογούμε **localhost** και πατάμε **Enter**.

Στην οθόνη που εμφανίζεται και στην περιοχή **Your Projects**, κάνουμε κλικ στο σύνδεσμο **Joomla\_1.5.14-Stable-Full\_Package**.



### 4.2.1 Βήμα 1 : Γλώσσα

Στην πρώτη οθόνη εγκατάστασης εμφανίζονται οι διαθέσιμες γλώσσες εγκατάστασης. Επιλέγουμε, εάν δεν είναι επιλεγμένη, την **el-GR- Greek**. Κάνουμε κλικ στο κουμπί **Επόμενο** που βρίσκεται πάνω δεξιά.



## 4.2.2 Βήμα 2 : Προληπτικός Έλεγχος

Στη δεύτερη οθόνη το Joomla κάνει κάποιους ελέγχους σχετικά με τη συμβατότητα του συστήματός μας. Στην παρακάτω ομάδα είναι απαραίτητο να είναι όλες οι ενδείξεις **ΝΑΙ**.

Έκδοση PHP >= 4.3.10	Ναι
- Υποστήριξη Συμπίεσης zip	Ναι
- Υποστήριξη XML	Ναι
- Υποστήριξη MySQL	Ναι
Η γλώσσα για τις συναρτήσεις MB είναι προκαθορισμένη	Ναι
Η υπερφόρτωση αριθμητικών για τις συναρτήσεις MB είναι απενεργοποιημένη	Ναι
configuration.php Εγγράμμο	Ναι



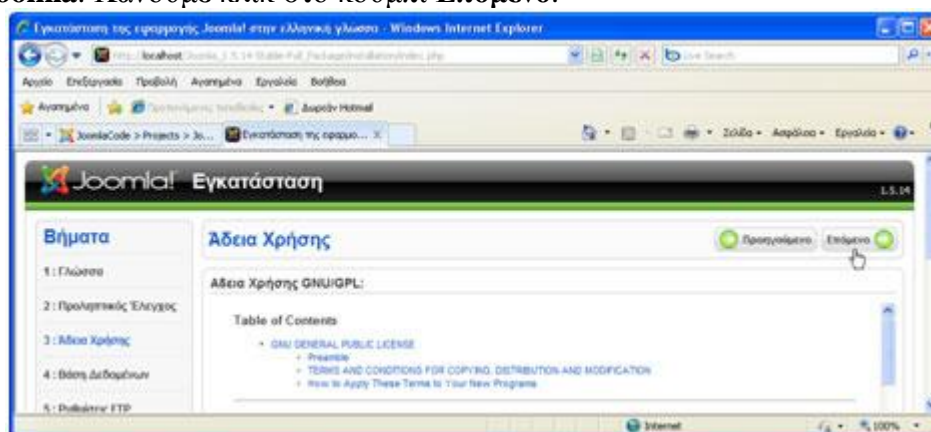
Σε περίπτωση που όλη η επάνω ομάδα είναι **ΝΑΙ**, προχωράμε στο επόμενο βήμα κάνοντας κλικ στο κουμπί **Επόμενο**. Σε περίπτωση που κάποιο από αυτά είναι με κόκκινο χρώμα, το διορθώνουμε και πατάμε **Επανέλεγχος**.

Στην παρακάτω ομάδα μερικές ενδείξεις είναι ενεργές και άλλες απενεργές, ανάλογα με τις ρυθμίσεις του **Server**.

Οδηγία	Συνιστάται	Πραγματικό
Ασφαλής λειτουργία (Safe Mode):	Απενεργό	Απενεργό
Προβολή Σφαλμάτων:	Απενεργό	Ενεργό
Μεταφόρτωση Αρχείων:	Ενεργό	Ενεργό
Χρήση Αυτόματων Εισαγωγικών (Magic Quotes):	Απενεργό	Απενεργό
Γενικός Μεταβλητής (Register Globals):	Απενεργό	Απενεργό
Εκτροπή Εξόδου στη Μνήμη (output buffering):	Απενεργό	Ενεργό
Αυτόματη Ένταξη Συνεδρίας:	Απενεργό	Απενεργό

## 4.2.3 Βήμα 3 : Άδεια Χρήσης

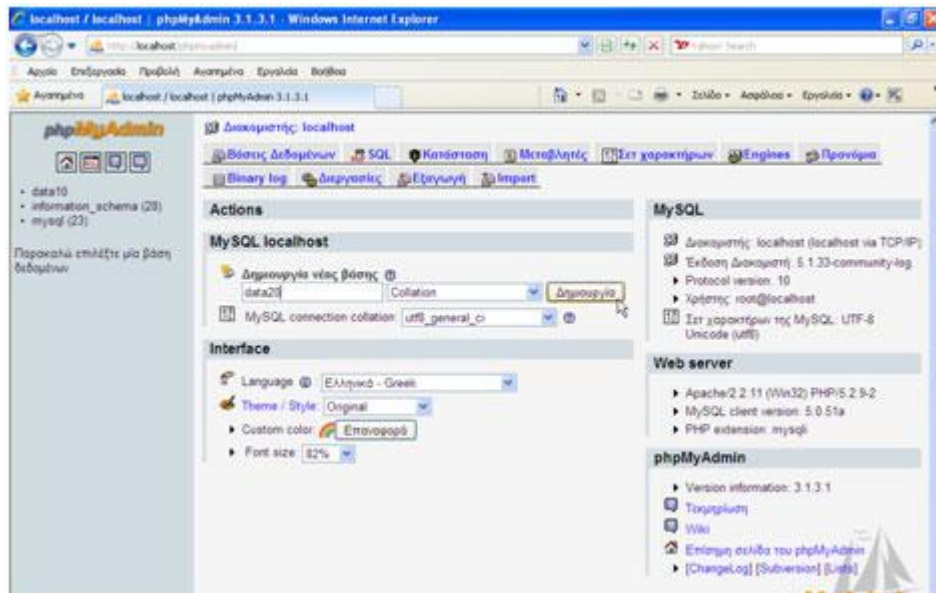
Στην επόμενη οθόνη εμφανίζεται η Άδεια Χρήσης GNU/GPL που χρησιμοποιεί το Joomla. Κάνουμε κλικ στο κουμπί **Επόμενο**.



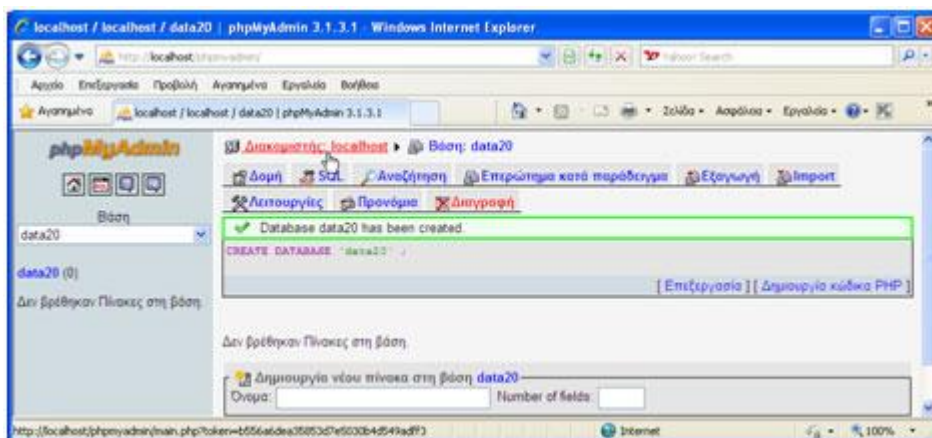
#### 4.2.4 Βήμα 4 : Βάση Δεδομένων

Πριν συνεχίσουμε την εγκατάσταση ανοίγουμε ένα νέο tab στον **Internet Explorer** πληκτρολογούμε **localhost/phpmyadmin** και πατάμε **Enter**.

Εμφανίζεται η παρακάτω οθόνη. Στο πλαίσιο κειμένου **Δημιουργία νέας βάσης** πληκτρολογούμε το όνομα που θα έχει η βάση δεδομένων (π.χ. data20). Στη συνέχεια, κάνουμε κλικ στο κουμπί **Δημιουργία**.

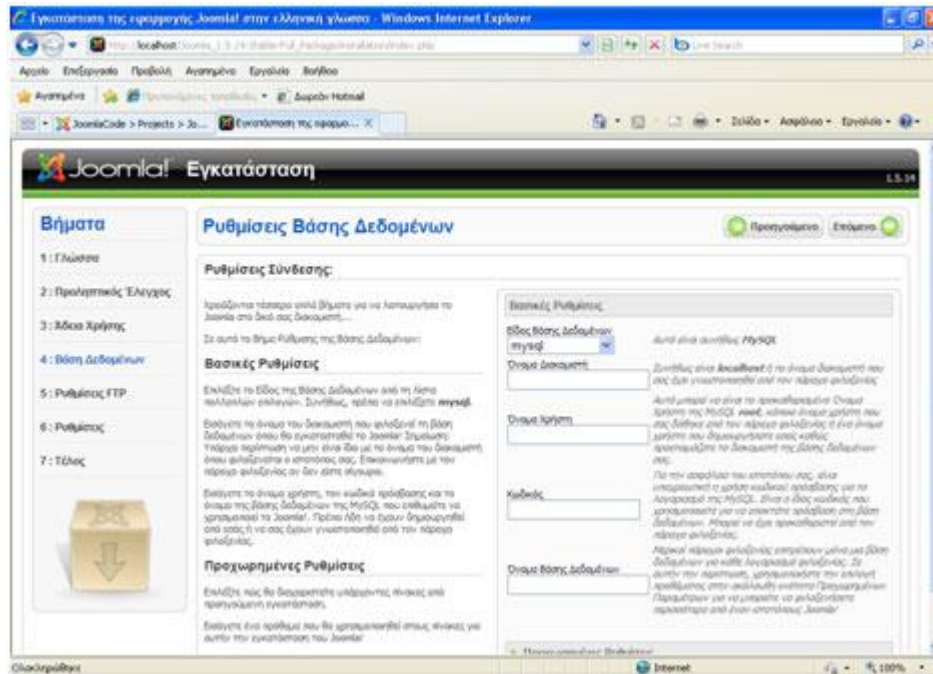


Εμφανίζεται η παρακάτω οθόνη. Στην οθόνη αυτή βλέπουμε ότι δημιουργήθηκε η βάση δεδομένων.



Για τη συγκεκριμένη βάση θα χρησιμοποιήσουμε το προεπιλεγμένο χρήστη **root** με **κενό password**. Αν θέλαμε θα μπορούσαμε να δημιουργήσουμε ένα δικό μας χρήστη από την αρχή.

Είμαστε στο πιο σημαντικό βήμα. Εδώ πρέπει να εισαγάγουμε τα στοιχεία της βάσης δεδομένων με την οποία θα συνεργάζεται το Joomla. Εμφανίζεται η παρακάτω οθόνη. Στην περιοχή **Βασικές Ρυθμίσεις** εισάγουμε στα πλαίσια κειμένου τα αντίστοιχα δεδομένα.



- Στην αναδιπλούμενη λίστα **Είδος Βάσης Δεδομένων** αφήνουμε την προεπιλεγμένη **mysql**.
- Στο πλαίσιο κειμένου **Όνομα Διακομιστή** εισαγάγουμε, συνήθως, το όνομα **localhost**.
- Στο πλαίσιο κειμένου **Όνομα Χρήστη** εισαγάγουμε το όνομα του χρήστη που έχουμε ορίσει στη βάση δεδομένων. ( root )
- Στο πλαίσιο κειμένου **Κωδικός** εισαγάγουμε τον κωδικό του χρήστη που έχουμε ορίσει στη βάση δεδομένων. ( κενό )
- Στο πλαίσιο κειμένου **Όνομα Βάσης Δεδομένων** εισαγάγουμε το όνομα της βάσης δεδομένων που έχουμε δημιουργήσει. ( data20 )

Όταν ολοκληρώσουμε την εισαγωγή δεδομένων, κάνουμε κλικ στο κουμπί **Επόμενο**.

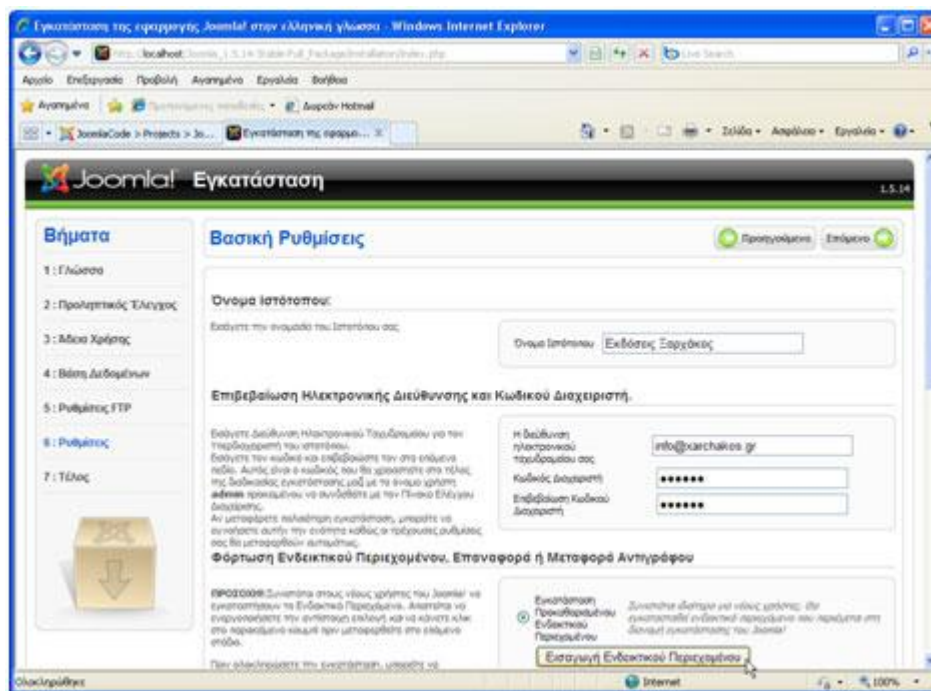
## 4.2.5 Βήμα 5 : Ρυθμίσεις FTP

Για λόγους ασφαλείας και προστασίας των αρχείων που χρησιμοποιεί το Joomla εδώ μας δίνεται η δυνατότητα δημιουργίας ενός FTP (File Transfer Protocol) λογαριασμού. Θα το προσπεράσουμε κάνοντας κλικ στο κουμπί **Επόμενο**.




## 4.2.6 Βήμα 6 : Ρυθμίσεις

Εμφανίζεται η παρακάτω οθόνη.



- Στο πλαίσιο κειμένου **Όνομα Ιστότοπου** εισάγουμε το όνομα της ιστοσελίδας που θα δημιουργήσουμε.

- Στο πλαίσιο κειμένου **Η διεύθυνση ηλεκτρονικού ταχυδρομείου σας** εισάγουμε το **e-mail** του διαχειριστή της ιστοσελίδας το οποίο θα χρησιμεύσει στη μετέπειτα επικοινωνία με τους χρήστες.
- Στο πλαίσιο κειμένου **Κωδικός Διαχειριστή** εισάγουμε τον κωδικό για να μπορούμε να συνδεόμαστε στην Περιοχή Διαχείρισης.
- Στο πλαίσιο κειμένου **Επιβεβαίωση Κωδικού Διαχειριστή** επαναλαμβάνουμε τον ίδιο κωδικό. Στη συνέχεια, κάνουμε κλικ στο κουμπί **Εισαγωγή Ενδεικτικού Περιεχομένου**.

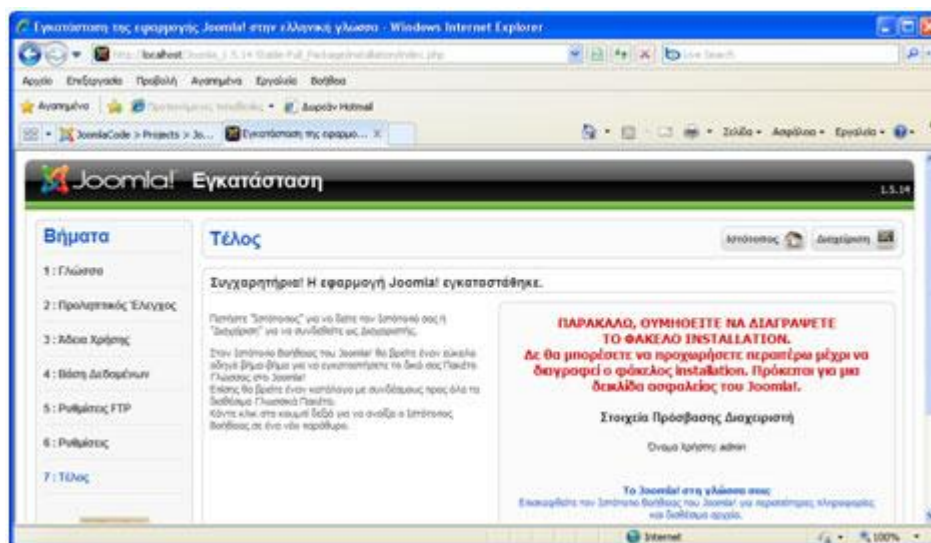


Η εισαγωγή ενδεικτικού περιεχομένου συνιστάται τόσο σε αρχάριους όσο και σε έμπειρους χρήστες, ώστε να μπορέσουν να κατανοήσουν ευκολότερα τη λειτουργία του Joomla.

Κάνουμε κλικ στο κουμπί **Επόμενο**.

#### 4.2.7 Βήμα 7 : Τέλος

Εμφανίζεται η παρακάτω οθόνη.



Στο τελευταίο βήμα το **Joomla** μας ενημερώνει ότι η εγκατάσταση ολοκληρώθηκε με επιτυχία. Πρέπει να διαγράψουμε το φάκελο **installation** για λόγους ασφαλείας.


Ο φάκελος βρίσκεται αποθηκευμένος στη θέση  
**C:\wamp\www\Joomla\_1.5.14-Stable-Full\_Package\installation.**

Εναλλακτικά, αντί να τον διαγράψουμε μπορούμε και να τον μετονομάσουμε (π.χ. installation23). Παρατηρούμε επίσης, ότι μας δίνει το **Username** του διαχειριστή το οποίο είναι προεπιλεγμένα **admin**.

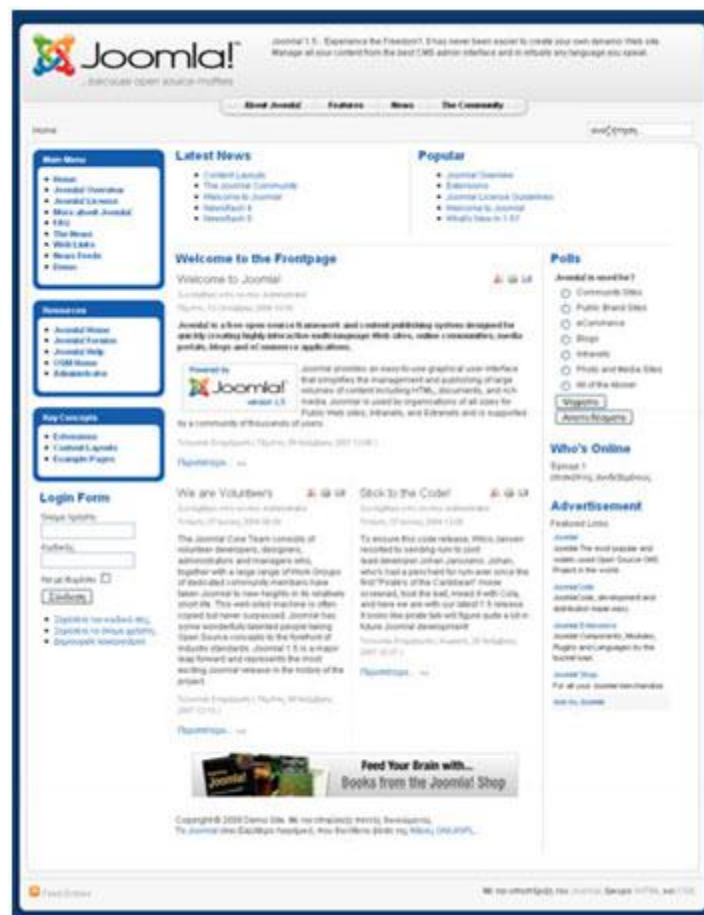
Αφού διαγράψουμε ή μετονομάσουμε το φάκελο **installation** έχουμε δύο επιλογές:

- Να μεταφερθούμε στο **Front End** (Εμφάνιση Ιστοσελίδας) κάνοντας κλικ στο κουμπί **Ιστότοπος**.
- Να μεταφερθούμε στο **Back End** (Περιοχή Διαχείρισης) κάνοντας κλικ στο κουμπί **Διαχείριση**.

Κάνουμε κλικ στο κουμπί **Ιστότοπος** και εμφανίζεται η σελίδα .

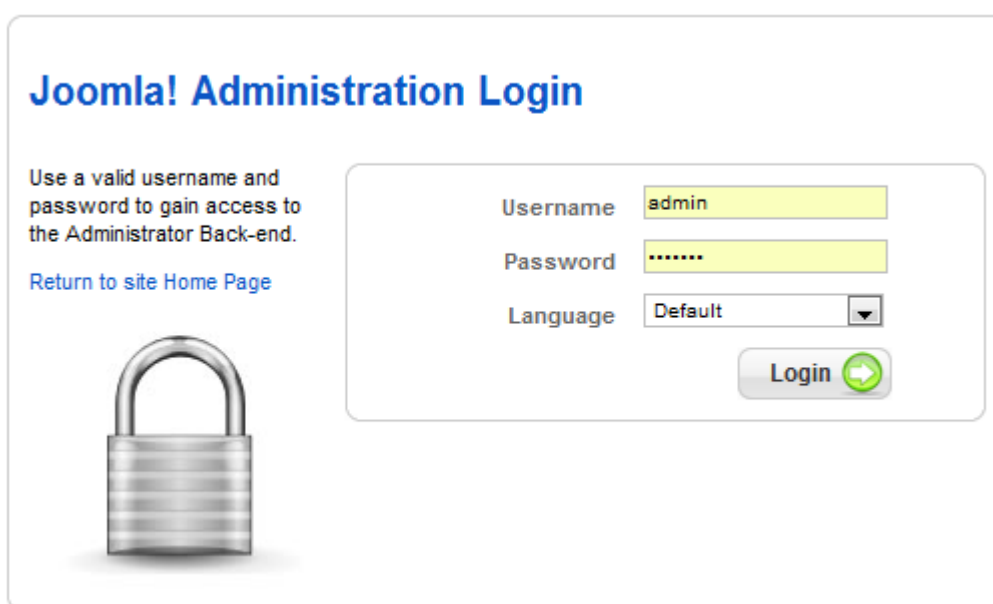


Αν δεν έχουμε διαγράψει ή μετονομάσει το φάκελο **installation** δεν θα εμφανιστεί το **Front End** (Εμφάνιση Ιστοσελίδας).



## 4.3 Περιβάλλον Διαχείρισης Joomla

Στο περιβάλλον διαχείρισης του Joomla, εμφανίζονται όλες οι δυνατότητες και οι διευκολύνσεις που παρέχει ως ένα Σύστημα Διαχείρισης Περιεχομένου. Για να εμφανίσουμε το περιβάλλον διαχείρισης, θα πρέπει να πάμε στον υποφάκελλο “administrator”, π.χ. <http://localhost/workhunter/administrator> (εφόσον έχουμε εγκαταστήσει το Joomla στον φάκελο workhunter). Στην συνέχεια δίνουμε το όνομα διαχειριστή (λογικά admin) και τον κωδικό που έχουμε ορίσει κατά την εγκατάσταση που είδαμε νωρίτερα.



**Joomla! Administration Login**


Use a valid username and password to gain access to the Administrator Back-end.

[Return to site Home Page](#)

Username

Password

Language

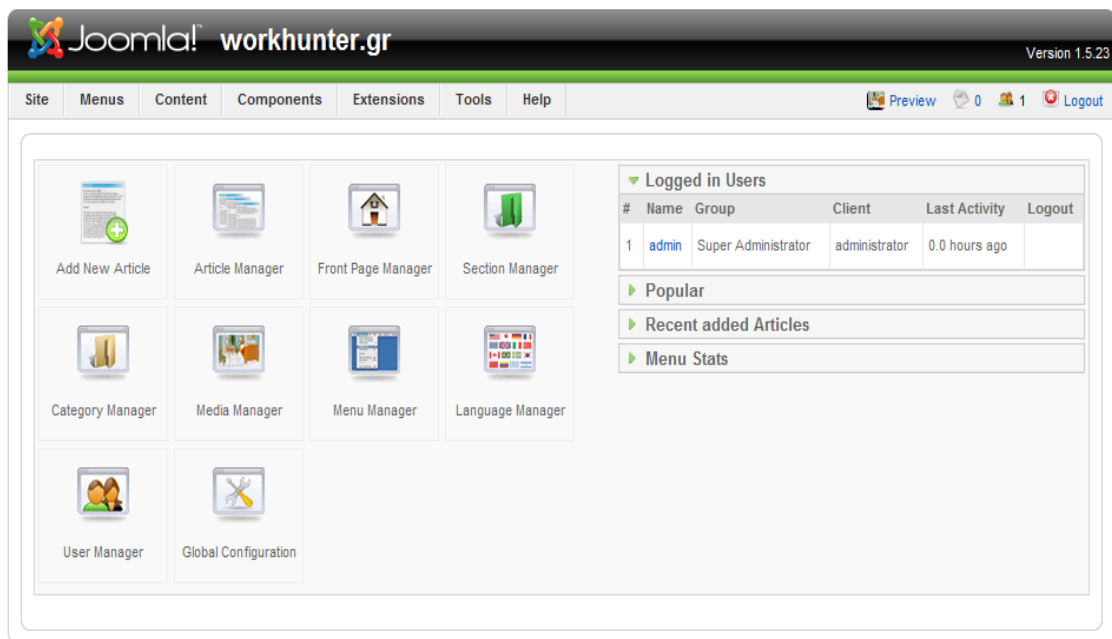
Login 

Όταν ξεκινάμε το περιβάλλον διαχείρισης εμφανίζεται το Control Panel το οποίο χωρίζεται σε δύο μέρη. Το **βασικό μενού** στο επάνω μέρος και στο κάτω μέρος τα μεγάλα χαρακτηριστικά **εικονίδια** που αντιπροσωπεύουν τις βασικότερες λειτουργίες του Joomla.

Το Βασικό μενού μας επιτρέπει να δούμε:

- ✓ Εντολές τοποθεσίας-Site π.χ. το Control Panel
- ✓ Τα υφιστάμενα Μενού και το Menu Manager
- ✓ Το περιεχόμενο (άρθρα, κατηγορίες, ενότητες)
- ✓ Τα εργαλεία – components
- ✓ Τα πρόσθετα και τον διαχειριστή προσθέτων
- ✓ Τα βοηθητικά εργαλεία και την
- ✓ Βοήθεια





Ενώ τα εικονίδια μας εμφανίζουν τις παρακάτω βασικότερες λειτουργίες :

### **Add New Article – Article Manager**

Μας επιτρέπει να προσθέσουμε ένα νέο άρθρο ή να διαχειριστούμε τα υπάρχοντα άρθρα

### **Front Page Manager**

Μας επιτρέπει να διαχειριστούμε το πώς θα εμφανίζεται η πρώτη σελίδα της τοποθεσίας μας.

### **Section – Category Manager**

Μας επιτρέπει να δημιουργήσουμε κατηγορίες και ενότητες, ώστε να εμφανίζονται αυτόματα τα άρθρα που θα δημιουργούμε στο σωστό σημείο.

### **Media Manager**

Μας επιτρέπει να προσθέτουμε εικόνες, video κ.ο.κ.

### **Menu Manager**

Μας επιτρέπει να δημιουργούμε ένα νέο μενού και να καθορίζουμε τον τρόπο εμφάνισής του ή να χειριστούμε υπάρχοντα μενού.

## Language Manager


Επιλέγουμε την προεπιλεγμένη γλώσσα για την τοποθεσία μας και ποιες γλώσσες υποστηρίζει η τοποθεσία μας.

## User Manager

Δημιουργία νέων χρηστών ή αλλαγές για υπάρχοντες.

## Global Configuration

Βασικές Ρυθμίσεις για την δυναμική τοποθεσία μας όπως φαίνεται παρακάτω.



### Global Configuration

[Site](#) | [System](#) | [Server](#)

#### Site Settings

Site Offline	<input checked="" type="radio"/> No <input type="radio"/> Yes
Offline Message	Εκτός λειτουργίας για τεχνικούς λόγους.  Παρακαλώ, προσπαθήστε αργότερα.
Site Name	workhunter.gr
Default WYSIWYG Editor	Editor - TinyMCE
List Length	20
Feed length	10
Feed Email	Author E-mail

#### Metadata Settings

Global Site Meta Description	Joomla! - δυναμική εφαρμογή για θεματικές πύλες και σύστημα διαχείρισης περιεχομένου
Global Site Meta Keywords	joomla, Joomla, myjoomla, ελληνική έκδοση, δυναμικός ιστότοπος, εφαρμογή διαχείρισης περιεχομένου
Show Title Meta Tag	<input type="radio"/> No <input checked="" type="radio"/> Yes
Show Author Meta Tag	<input type="radio"/> No <input checked="" type="radio"/> Yes

Στο Global Configuration μπορούμε να θέσουμε την τοποθεσία μας “Offline”, να ορίσουμε τον τίτλο - όνομα της τοποθεσία μας και τα Metadata Settings. Επίσης επιλέγουμε και κάποιον άλλο επεξεργαστή ιστοσελίδων, εάν αυτό χρειάζεται.

Για λόγους ασφαλείας, μπορεί να θέλουμε να αλλάξουμε τον κωδικό ασφαλείας του διαχειριστή ή να δημιουργήσουμε και δεύτερο διαχειριστή (συστήνεται). Αρκεί να πάμε στο Control Panel και να επιλέξουμε το “User Manager”. Επιλέγουμε τον χρήστη και δίνουμε την εντολή Edit (στα δεξιά). Αλλάζουμε τα στοιχεία του χρήστη όπως π.χ. το password. Μπορούμε να δημιουργήσουμε και νέο χρήστη με την εντολή New, ορίζοντας σε αυτών και ποια δικαιώματα θα έχει.

Τέλος, μια από τις βασικότερες λειτουργίες που πρέπει να αναλύσουμε εκτενέστερα, είναι η εγκατάσταση/απεγκατάσταση προσθέτων. Στο κεντρικό μενού επιλέγουμε Extensions μετά Install/Uninstall και εμφανίζεται η παρακάτω οθόνη :

The screenshot shows the Joomla! Extension Manager interface. At the top, there is a title bar with the text "Extension Manager" and a "Help" icon. Below the title bar is a navigation menu with tabs: "Install", "Components", "Modules", "Plugins", "Languages", and "Templates". The main content area is divided into three sections: "Upload Package File", "Install from Directory", and "Install from URL".

- Upload Package File:** This section has a "Package File:" label, a "Choose File" button, a text field containing "No file chosen", and an "Upload File & Install" button.
- Install from Directory:** This section has an "Install Directory:" label, a text field containing "C:\wamp\www\workhunter\tmp", and an "Install" button.
- Install from URL:** This section has an "Install URL:" label, a text field containing "http://", and an "Install" button.

Επιλέγουμε το αρχείο που έχουμε κατεβάσει, από τη επιλογή Choose file και στη συνέχεια πατάμε το κουμπί **upload file & install**, για να εγκατασταθεί το πρόσθετο μας. Εάν όλα έγιναν σωστά, λαμβάνουμε ένα μήνυμα επιβεβαίωσης.

Συνοψίζοντας μπορούμε να πούμε πως το περιβάλλον διαχείρισης του Joomla, παρέχει όλες τις δυνατότητες ενός σύγχρονου Συστήματος Διαχείρισης Περιεχομένου, έτσι ώστε και ένας χρήστης με ελάχιστες γνώσεις πληροφορικής, να είναι σε θέση να δημιουργήσει και να διαχειριστεί μετέπειτα μια ιστοσελίδα.

## Κεφάλαιο 5 Ανάλυση Απαιτήσεων

### 5.1 Εισαγωγή

Πριν ξεκινήσουμε την ανάλυση των προδιαγραφών και των απαιτήσεων της ιστοσελίδας που δημιουργήσαμε , η πρώτη ανάγκη που γεννήθηκε ήταν η πλήρη κατανόηση του Joomla σε όλα τα επίπεδα του. Αυτό ήταν απαραίτητο να γίνει, αφού είχαμε αποφασίσει εξ' αρχής πως όχι μόνο θα χρησιμοποιήσουμε τις δυνατότητες που μας προσφέρει το Joomla, αλλά θα επεμβαίναμε και σε αυτό, είτε με αλλαγή του κώδικα που ήδη υπάρχει, είτε με δημιουργία δικού μας εξολοκλήρου.

Ξεκινώντας από το πίνακα ελέγχου(control panel) του διαχειριστή, αναλύσαμε και κατανοήσαμε τις έννοιες που είδαμε και προηγουμένως όπως Component , Menu, Category, Section, Content, Module, Article αλλά και πως συνδέονται αυτές τελικά μεταξύ τους.

Έπειτα έπρεπε να αναλύσουμε τη βάση δεδομένων, που αυτόματα δημιουργεί το Joomla. Να κατανοήσουμε τη δομή της και αν αυτή τελικά θα μας καλύψει τις απαιτήσεις που έχουμε θέσει από την αρχή ή χρειάζεται περαιτέρω προσθήκες.

Τέλος έπρεπε να ανακαλύψουμε που εκτελούνται οι διεργασίες που παρέχει το Joomla. Αυτό θα μπορούσε να επιτευχθεί μόνο βρίσκοντας τη διαδρομή των αρχείων που εμπλέκονται σε μια διεργασία και με ποια σειρά αυτά θα εκτελεστούν, πράγμα αρκετά δύσκολο αναλογιζόμενοι το πλήθος των φακέλων και των αρχείων που είχε δημιουργήσει το Joomla.

Προσπαθήσαμε λοιπόν να ανακαλύψουμε όχι μόνο **ΤΙ** μας παρέχει το Joomla σαν σύστημα διαχείρισης περιεχομένου αλλά κυρίως το **ΠΩΣ** αυτό τελικά υλοποιείται. Για να πραγματοποιηθεί όμως αυτό έπρεπε η εργασία μας να στηριχθεί στους παρακάτω βασικούς άξονες :

- ✓ Συχνή επέμβαση και δημιουργία Html και Css κώδικα.
- ✓ Κατανόηση και Εμπλουτισμός της Βάσης Δεδομένων μας
- ✓ Μελέτη του Joomla API για τροποποίηση του υπάρχοντος κώδικα
- ✓ Δημιουργία επαναχρησιμοποιήσιμου Php και Javascript κώδικα.

## 5.2 Βάση Δεδομένων

Αναλύοντας τη βάση δεδομένων διαπιστώσαμε ότι σε μεγαλύτερο ποσοστό η βάση που δημιουργεί το Joomla μας καλύπτει.

Θα χρησιμοποιήσουμε τον πίνακα **jos\_content** για να αποθηκεύονται οι αγγελίες χωρίς να προσθέσουμε παραπάνω πεδία. Το ίδιο ισχύει για τον πίνακα **jos\_categories** όπου δε θα προσθέσουμε τίποτα και θα αποθηκεύσουμε τις κατηγορίες των αγγελιών.

Στη χρήση του πίνακα **jos\_users** υπάρχει η ανάγκη προσθήκης αρκετών πεδίων για να καλύψουμε τις απαιτήσεις μας. Θα προσθέσουμε ένα αναγνωριστικό πεδίο **employer** με το οποίο θα διαχωρίζουμε τον εργοδότη(1) από τον εργαζόμενο(0). Επίσης θα γίνει προσθήκη του πεδίου **email2** στο οποίο ο χρήστης θα επαναπληκτρολογεί το ηλεκτρονικό του ταχυδρομείο για επαλήθευση. Αντίστοιχα θα γίνει η προσθήκη των πεδίων **firstname**, **phone**, **street**, **streetnumber** και **town** στη βάση, γιατί η καταχώρηση αυτών των στοιχείων από κάθε χρήστη, είναι κρίσιμη για εμάς. Τέλος θα γίνει η προσθήκη των πεδίων **logo** και **bio** για να αποθηκεύσουμε τα λογότυπα των εταιριών και τα βιογραφικά των εργαζομένων αντίστοιχα.

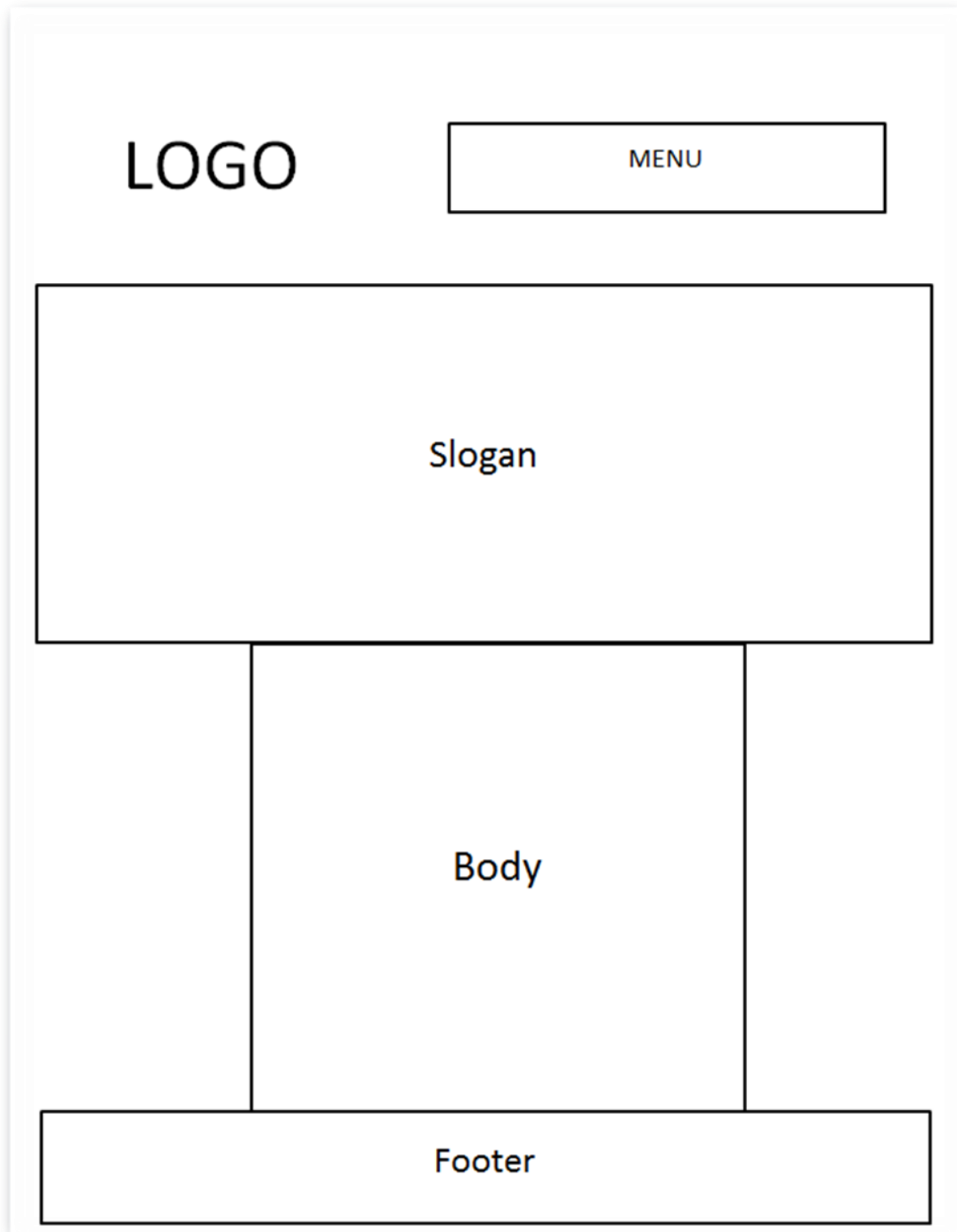
Για να καλύψουμε τις ανάγκες τις ιστοσελίδας μας, έπρεπε να δημιουργήσουμε και δικούς μας πίνακες στη βάση δεδομένων, των οποίων η υλοποίηση θα παρουσιαστεί στο επόμενο κεφάλαιο.

## 5.3 Σχεδίαση Σελίδας (template)

Το σχεδιαστικό κομμάτι της ιστοσελίδας, αποφασίσαμε πως θα το δημιουργούσαμε εμείς και δε θα χρησιμοποιούσαμε κανένα έτοιμο πρότυπο (template). Το προεπιλεγμένο template που φορτώνεται με την εγκατάσταση του Joomla είναι το **rhuk\_milkyway** το οποίο βρίσκεται στη διαδρομή `...\www\workhunter\templates\`.

Στο φάκελο λοιπόν αυτό, υπάρχει το αρχείο **index.php** που εμπεριέχου όλο το κώδικα html και είναι υπεύθυνο για τη μορφή του template. Αντίστοιχα υπάρχει και ο φάκελος **Css** στον οποίο είναι τοποθετημένα όλα τα `.css` αρχεία του template. Σε αυτά ακριβώς τα αρχεία θα επέμβουμε για να αλλάξουμε ολοκληρωτικά την εμφάνιση της ιστοσελίδας μας.

Η ιστοσελίδα μας θα έχει σαν μορφή το παρακάτω mockup :



## 5.4 Γενικό Μενού

Άλλο ένα μέρος της ιστοσελίδας μας που αποφασίσαμε να το υλοποιήσουμε μόνοι μας χωρίς να χρησιμοποιήσουμε το σύστημα που μας προσφέρει το Joomla.

Το μενού θα είναι κεντρικό θα βρίσκεται πάνω δεξιά (όπως φαίνεται και στη παραπάνω εικόνα) και θα είναι απόλυτα δυναμικό. Αν για παράδειγμα ο χρήστης πραγματοποιήσει είσοδο, δε θα εμφανίζεται πλέον η επιλογή εγγραφή και τη θέση της θα πάρει μια άλλη επιλογή (λογαριασμός μου).

Το μενού θα έχει τη παρακάτω επιλογές :

- ✓ **Κεντρική** όπου θα εμφανίζεται η κεντρική μας σελίδα.
- ✓ **Αναζήτηση** για να μπορέσει ο χρήστης να αναζητήσει μια αγγελία.
- ✓ **Επικοινωνία** όπου θα παρουσιάζονται πληροφορίες για την ιστοσελίδα μας.
- ✓ **Εγγραφή** που χωρίζεται σε 2 υπομενού το **Εργοδότης** και **Εργαζόμενος** για να κάνουν την εγγραφή τους στην ιστοσελίδα.
- ✓ **Ο Λογαριασμός μου** και θα εμφανίζεται μόνο όταν έχει γίνει επιτυχημένη είσοδος από το χρήστη, αντικαθιστώντας το μενού Εγγραφή. Θα έχει και αυτό υπομενού ανάλογα με το είδος του χρήστη που θα εισέλθει, όπως φαίνεται παρακάτω.

### Εργαζόμενος

Τα στοιχεία μου

Οι αγγελίες μου

### Εργοδότης

Νέα αγγελία

Τα στοιχεία μου

Οι αγγελίες μου

Το υπομενού **οι αγγελίες μου** εμφανίζεται και τις δυο περιπτώσεις, μόνο που στον εργαζόμενο θα εμφανίζει τις αγγελίες στις οποίες έχει δώσει το βιογραφικό του, ενώ στον εργοδότη θα εμφανίζονται οι αγγελίες που έχει δημιουργήσει ο ίδιος, τις οποίες θα μπορεί και να αλλάξει. Τέλος το υπομενού **τα στοιχεία μου**, θα εμφανίζει τα στοιχεία του εγγεγραμμένου χρήστη, είτε εργαζομένου είτε εργοδότη, δίνοντας παράλληλα και τη δυνατότητα αλλαγής τους.

## 5.5 Εγγραφή χρήστη

Η εγγραφή ενός χρήστη στην ιστοσελίδα μας είναι ίσως το κομμάτι της ανάλυσης, που μας απασχόλησε περισσότερο από οποιοδήποτε άλλο και θα εξηγήσουμε παρακάτω το γιατί. Το Joomla διαθέτει φόρμα εγγραφής ενός νέου χρήστη, έχοντας και το απαραίτητο έλεγχο(validation) των πεδίων αυτής. Δε παρέχεται όμως η δυνατότητα να προσθέσεις μια νέα φόρμα εγγραφής, παρά μόνο να εμπλουτίσεις τη ήδη υπάρχουσα.

Η ιστοσελίδα μας όμως απαιτεί να υπάρχουν δυο και όχι μία φόρμα εγγραφής ενός επισκέπτη, καθώς απευθύνεται σε δύο είδη χρηστών, τον **εργαζόμενο** και τον **εργοδότη**. Είναι προφανές πως ανάλογα με την είδος του χρήστη που επισκέπτεται τη σελίδα μας θα πρέπει να του παρέχουμε και την αντίστοιχη φόρμα εγγραφής. Ο εργοδότης έχει μια συγκεκριμένη φόρμα εγγραφής όπως αντίστοιχα και ο εργαζόμενος.

Για παράδειγμα, άλλες είναι οι πληροφορίες που χρειάζεται να καταχωρήσουμε στη βάση δεδομένων της ιστοσελίδας, για μια εταιρία (εργοδότη) που θέλει να τοποθετήσει σε εμάς μια αγγελία εργασίας και εντελώς διαφορετικά τα στοιχεία ενός φοιτητή, που λίγο πριν τελειώσει τη σχολή του, προσπαθεί να ψάξει για εργασία (εργαζόμενος). Από την εταιρία σίγουρα ζητάμε την επωνυμία της, ενώ από το φοιτητή χρειαζόμαστε και το επώνυμο αλλά και το όνομα του. Αντίστοιχα τα στοιχεία της διεύθυνσης μιας εταιρίας, είναι **υποχρεωτικά** για εμάς, γιατί θέλουμε να εμφανίζονται οπωσδήποτε στη παρουσίαση μιας μελλοντικής τους αγγελίας, ενώ ο φοιτητής **δεν είναι υποχρεωτικό** να μας συμπληρώσει κανένα στοιχείο της διεύθυνσης του. Αντίθετα το τηλέφωνο, το username, το password και το ηλεκτρονικό ταχυδρομείο, θα ήταν υποχρεωτικό και στις δυο αυτές περιπτώσεις.

Τέλος είναι υποχρεωτικό για εμάς, η εταιρία να μας δώσει ένα αρχείο με το λογότυπο της και επίσης υποχρεώνουμε το φοιτητή, να μας δώσει ένα αρχείο με το βιογραφικό του, που θα αποστείλει μέσω της ιστοσελίδας μετέπειτα στις εταιρίες.

Έτσι λοιπόν, αποφασίσαμε να χρησιμοποιήσουμε τη φόρμα εγγραφής που μας παρείχε το Joomla και να την εμπλουτίσουμε με παραπάνω πεδία, που κρίναμε πως θα ήταν για εμάς απαραίτητα. Το διαχωρισμό όμως των επισκεπτών, στην αντίστοιχη φόρμα εγγραφής σύμφωνα με το είδος τους (εργοδότης-εργαζόμενος), την επιτύχαμε με την υλοποίηση δικού μας κώδικα, που θα διαμόρφωνε **την ίδια φόρμα** ανάλογα με το χρήστη που θα πραγματοποιούσε την εγγραφή.

Αν ο επισκέπτης ήταν για παράδειγμα εργοδότης και επέλεγε από το μενού την εγγραφή για εργοδότες, τότε θα μεταφερόταν αυτόματα σε μια σελίδα όπου η φόρμα εγγραφής, θα εμφάνιζε μόνο τα πεδία που θα απευθύνονται σε εργοδότη, θα διαχώριζε ποια πεδία είναι υποχρεωτικά και ποια όχι και προφανώς θα απέκρυπτε όσα πεδία θα προοριζόταν για το χρήστη εργαζόμενο.



Για τον έλεγχο(validation) των πεδίων της φόρμας, απενεργοποιήσαμε αυτόν που μας παρείχε το Joomla και δημιουργήσαμε δικό μας, κάνοντας χρήση του validation jQuery και ξαναγράφοντας όλα τα μηνύματα ελέγχου που θα εμφανίζονταν σε περίπτωση λάθους.

Τα στοιχεία που επιλέξαμε τελικά να ζητάμε από τους επισκέπτες, ανάλογα με το είδος της εγγραφής που θα πραγματοποιούσαν (εργαζόμενου ή εργοδότη) εμφανίζονται παρακάτω :

#### ***ΕΡΓΑΖΟΜΕΝΟΣ***

Όνομα  
Επώνυμο  
Όνομα Χρήστη  
Κωδικός πρόσβασης  
Έλεγχος Κωδικού  
E-mail  
Έλεγχος E-mail  
Τηλέφωνο  
Πόλη  
Οδός  
Αριθμός Οδού  
Βιογραφικό

#### ***ΕΡΓΟΔΟΤΗΣ***

Επωνυμία  
Όνομα Χρήστη  
Κωδικός πρόσβασης  
Έλεγχος Κωδικού  
E-mail  
Έλεγχος E-mail  
Τηλέφωνο  
Πόλη  
Οδός  
Αριθμός Οδού  
Λογότυπο Εταιρίας

## **5.6 Αλλαγή στοιχείων χρήστη**

Σίγουρα θα δώσουμε τη δυνατότητα στον εγγεγραμμένο χρήστη, να μπορεί να δει τα στοιχεία που έχει δώσει σε εμάς και αν θέλει να προβεί σε αλλαγές αυτών. Τη λειτουργία αυτή μας τη παρέχει το Joomla και όπως στη περίπτωση της εγγραφής θα στηριχτούμε πάνω σε αυτή. Το πρόβλημα της μίας φόρμας εμφανίζεται και εδώ, το οποίο όμως λύνουμε με χρήση δικών μας script, εμφανίζοντας τα στοιχεία που χρειάζονται ανάλογα με το χρήστη ( εργοδότη ή εργαζόμενο).

Αν ο χρήστης επιλέξει τη λειτουργία της προβολής των στοιχείων του, τον εξαναγκάζουμε - για λόγους ασφαλείας - να πληκτρολογήσει ξανά το κωδικό του, ίδιο ή προφανώς και καινούργιο. Επίσης στον εργοδότη εμφανίζεται η φωτογραφία του λογότυπου που μας είχε δώσει, προσθέτοντας όμως τώρα και ένα επιπλέον πεδίο σε περίπτωση που θέλει να το αλλάξει. Το νέο λογότυπο θα ανανεωθεί αυτόματα για όλες τις ενεργές αγγελίες του. Αντίστοιχα και για τον εργαζόμενο, θα φαίνεται το αρχείο του βιογραφικού που μας είχε δώσει. Θα έχει και αυτός τη δυνατότητα αν θελήσει, να μας ανεβάσει ένα νέο του βιογραφικό, το οποίο προφανώς και θα αντικαταστήσει το είδη υπάρχον.

## 5.7 Αναζήτηση

Είναι μια λειτουργία που το Joomla τι παρέχει και ως component και ως module. Εμείς όμως πήραμε την απόφαση πως πρέπει να την δημιουργήσουμε από την αρχή με τη χρήση των τεχνολογιών Php , JavaScript (jQuery), Html και Css.

Η έννοια της αναζήτησης δε θα περιορίζετε μόνο στην αναζήτηση μιας αγγελίας από ένα εγγεγραμμένο χρήστη ή επισκέπτη της ιστοσελίδας μας, αλλά με τον ίδιο περίπου κώδικα θα λύσουμε και άλλα προβλήματα "αναζήτησης". Θα χρησιμοποιηθεί για να εμφανίζει στο χρήστη αποτελέσματα αγγελιών, στις οποίες έχει δώσει το βιογραφικό του και αντίστοιχα θα μπορεί ένας εγγεγραμμένος εργοδότης, να δει πόσες αγγελίες έχει αυτή τη στιγμή εν ενεργεία στην ιστοσελίδα μας.

Για αυτό το λόγο λοιπόν επιλέξαμε την εκ νέου υλοποίηση της έννοιας **Αναζήτηση**.

## 5.8 Εισαγωγή / Ενημέρωση Αγγελίας

Η **καταχώρηση** νέας αγγελίας, θα είναι σίγουρα μια λειτουργία που θα παρέχει η ιστοσελίδα μας σε εγγεγραμμένους χρήστες, μόνο όμως τύπου εργοδότη. Στην πραγματικότητα, θα ξαναδημιουργήσουμε την δυνατότητα καταχώρησης ενός άρθρου, που μας προσφέρει το ίδιο το Joomla από τον πίνακα ελέγχου του διαχειριστή. Τα στοιχεία που θα πρέπει να συμπληρώσει ο εργοδότης, έτσι ώστε να καταχωρηθεί η νέα του αγγελία επιτυχώς, είναι τα ακόλουθα :

- ✓ Τίτλος
- ✓ Κατηγορία
- ✓ Εκπαίδευση
- ✓ Προϋπηρεσία

- ✓ Θέση Προϊσταμένου
- ✓ Μισθός
- ✓ Περιγραφή αγγελίας

Εκτός από την καταχώρηση στον εργοδότη θα παρέχεται η δυνατότητα της **ενημέρωσης**. Θα έχει δηλαδή την επιλογή, εάν το επιθυμεί, να τροποποιήσει όλα τα στοιχεία της αγγελίας που είχε συμπληρώσει κατά την καταχώρηση της. Στόχος μας είναι ο εργοδότης, να έχει την απόλυτη ευθύνη και διαχείριση των αγγελιών του.

## 5.9 Παρουσίαση Αγγελίας

Πολύ σημαντικό θέμα κατά την ανάλυση, ήταν η σελίδα παρουσίασης της αγγελίας, δεδομένου ότι μετά τη κεντρική, θα ήταν η πιο πολυσύχναστη ιστοσελίδα των χρηστών. Θα πρέπει να είναι ευανάγνωστη και φιλική προς το χρήστη, με σκοπό την άμεση κατανόηση των απαιτήσεων της.

Θα πρέπει να δίνει το χώρο στον εργοδότη, να παρουσιάσει όσο το δυνατόν περισσότερα στοιχεία για την ίδια την εταιρία, αλλά και τις απαιτήσεις αυτής κάθε αυτής της αγγελίας. Έτσι αποφασίσαμε να έχουμε κάποιες γρήγορες πληροφορίες, που θα αφορούν την αγγελία πριν καν ο εργαζόμενος διαβάσει το κεντρικό της κείμενο .

Αυτές είναι :

- ✓ Προϋπηρεσία
- ✓ Εκπαίδευση
- ✓ Θέση προϊσταμένου (ναι / όχι)
- ✓ Μισθός

Με μια γρήγορη ματιά λοιπόν, ο εργαζόμενος θα έχει καταλάβει αν αξίζει το κόπο, για τον ίδιο πάντα, να κάτσει και να διαβάσει όλο το κείμενο των απαιτήσεων μιας αγγελίας ή θα πρέπει να συνεχίσει με μια νέα αναζήτηση.

Επίσης παρουσιάζουμε το λογότυπο της εταιρίας, τη διεύθυνση και το τηλέφωνο της, καθώς επίσης και την ακριβή τοποθεσία της στο χάρτη, με τη χρήση του Google Maps.

Αν ο χρήστης θεωρεί, πως η αγγελία εργασίας πληροί τις απαιτήσεις του και θέλει να υποβάλει το βιογραφικό του, έχουμε τοποθετήσει ένα κουμπί για την αποστολή βιογραφικού, που είναι ενεργοποιημένο μόνο για εγγεγραμμένους χρηστές και στέλνει με ηλεκτρονικό ταχυδρομείο, το βιογραφικό του εργαζόμενου στο συγκεκριμένο εργοδότη.

Τέλος κάθε χρήστης της ιστοσελίδας μας, ακόμα και επισκέπτης, έχει τη δυνατότητα να πατήσει το γνωστό σε όλους μας κουμπί **LIKE** του Facebook ή να αποστείλει στο λογαριασμό Facebook ενός φίλου του, μια συγκεκριμένη αγγελία.

## 5.10 Σχόλια

Ένα επιπλέον στοιχείο που θέλαμε να προσθέσουμε στην ιστοσελίδα μας είναι η προσθήκη σχολίων. Σε κάθε αγγελία θα αφιερώσουμε στο τέλος ένα χώρο για σχόλια, που θα επιτρέπονται ακόμα και από ένα απλό επισκέπτη.

Δίνουμε έτσι τη δυνατότητα στους επισκέπτες μας, όχι μόνο να βλέπουν τις πληροφορίες που παρέχει ο κάθε εργοδότης στην αγγελία του, αλλά να κάνουν ερωτήσεις και παρατηρήσεις πάνω σε αυτή, ανοίγοντας έτσι ουσιαστικά ένα διάλογο μεταξύ των εργαζομένων και εργοδοτών.

Σίγουρα αυτό είναι ένα πλεονέκτημα για την ιστοσελίδα και τους χρήστες της, το οποίο όμως χρειάζεται τρομερή προσοχή από τους διαχειριστές αυτής(moderators). Είναι απαραίτητη η συνεχής παρακολούθηση των σχολίων, σε όλο το εύρος της ιστοσελίδας, για πιθανά σχόλια προσβλητικά ή υβριστικά και γενικά για περιεχόμενο που θα πρέπει να μη παρουσιάζεται καθόλου.

## 5.11 Είσοδος στην Ιστοσελίδα ( Log In )

Η δυνατότητα εισόδου ενός εγγεγραμμένου χρήστη στην ιστοσελίδα, εμφανίζεται στο Joomla με τη χρήση της λειτουργίας **module**. Εμείς θα χρησιμοποιήσουμε το συγκεκριμένο module αλλά θα του αλλάξουμε τοποθεσία και εμφάνιση. Αυτό θα υλοποιηθεί με την εγκατάσταση ενός νέου plugin, από τον πίνακα ελέγχου του Joomla.

Η λειτουργία εισόδου στην ιστοσελίδα μας, θα ζητάει από το χρήστη να πληκτρολογήσει επιτυχώς, το όνομα χρήστη και το κωδικό του. Σε περίπτωση που ο εγγεγραμμένος πάντα χρήστης, δε θυμάται κάποιο από τα στοιχεία αυτά, θα δίνεται η δυνατότητα, να αποστέλλονται στο ηλεκτρονικό ταχυδρομείο, που μας είχε δώσει κατά την διαδικασία εγγραφής του.

## Κεφάλαιο 6 Υλοποίηση Ιστοσελίδας

### 6.1 Εισαγωγή

Στην υλοποίηση θα παρουσιάσουμε όσο πιο αναλυτικά γίνεται, όλα τα μέρη που μας απασχόλησαν κατά την ανάλυση της πτυχιακής, τη πορεία που ακολουθήσαμε για να φτάσουμε στο τελικό αποτέλεσμα καθώς και όλα τα τεχνολογικά μέσα που μας χρειάστηκαν.

Θα γίνει λεπτομερής παρουσίαση και επεξήγηση του κώδικα που υλοποιήθηκε από εμάς, των λειτουργιών του Joomla που χρησιμοποιήσαμε και τα ακριβή αρχεία τα οποία χρειάστηκε να εμπλουτίσουμε, έτσι ώστε να πραγματοποιήσουμε τους αρχικούς μας στόχους.

Τέλος, θα αναλύσουμε και θα δείξουμε τις αλλαγές που χρειάστηκαν στη βάση δεδομένων μας, είτε αφορούσαν απλή προσθήκη ενός πεδίου σε ένα πίνακα, είτε την δημιουργία ενός ολόκληρου πίνακα από την αρχή.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι για την εισαγωγή του δικού μας **εξολοκλήρου** κώδικα στο Joomla, χρησιμοποιήσαμε την επέκταση(extension) με ονομασία **Jumi**. Όλοι οι κώδικες που θα παρουσιάσουμε παρακάτω και δεν αφορούν εμπλούτιση αρχείων του Joomla αλλά χτίσιμο από εμάς από την αρχή, θα έχουν εισαχθεί με τη βοήθεια του γνωστού πρόσθετου **Jumi**.

### 6.2 Βάση Δεδομένων

Για να δούμε τοπικά τη βάση δεδομένων, που δημιουργεί αυτόματα το Joomla κατά την εγκατάσταση του, πληκτρολογούμε σε ένα browser της επιλογής μας **http://localhost/phpmyadmin/** και επιλέγουμε τη βάση που δημιουργήσαμε για το Joomla πριν την εγκατάσταση του. Η βάση και όλοι οι πίνακες της, φαίνονται στη παρακάτω φωτογραφία.

phpMyAdmin

localhost ▶ workhunter

Structure SQL Search Query Export Import Operations Privileges Drop

Table	Action	Records <sup>1</sup>	Type	Collation	Size	Overhead
<input type="checkbox"/> jos_banner		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_bannerclient		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_bannertrack		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_categories		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_components		32	MylSAM	utf8_general_ci	7.2 KiB	-
<input type="checkbox"/> jos_contact_details		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_content		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_content_frontpage		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_content_rating		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_core_acl_aro		1	MylSAM	utf8_general_ci	9.0 KiB	-
<input type="checkbox"/> jos_core_acl_aro_groups		11	MylSAM	utf8_general_ci	4.5 KiB	-
<input type="checkbox"/> jos_core_acl_aro_map		0	MylSAM	utf8_general_ci	4.0 KiB	-
<input type="checkbox"/> jos_core_acl_aro_sections		1	MylSAM	utf8_general_ci	9.0 KiB	-
<input type="checkbox"/> jos_core_acl_groups_aro_map		1	MylSAM	utf8_general_ci	7.0 KiB	-
<input type="checkbox"/> jos_core_log_items		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_core_log_searches		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_groups		3	MylSAM	utf8_general_ci	2.1 KiB	-
<input type="checkbox"/> jos_menu		1	MylSAM	utf8_general_ci	4.6 KiB	-
<input type="checkbox"/> jos_menu_types		1	MylSAM	utf8_general_ci	3.1 KiB	-
<input type="checkbox"/> jos_messages		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_messages_cfg		0	MylSAM	utf8_general_ci	2.0 KiB	-
<input type="checkbox"/> jos_migration_backlinks		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_modules		15	MylSAM	utf8_general_ci	4.8 KiB	-
<input type="checkbox"/> jos_modules_menu		1	MylSAM	utf8_general_ci	2.0 KiB	-
<input type="checkbox"/> jos_newsfeeds		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_plugins		33	MylSAM	utf8_general_ci	7.9 KiB	-
<input type="checkbox"/> jos_polls		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_poll_data		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_poll_date		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_poll_menu		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_sections		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_session		1	MylSAM	utf8_general_ci	6.2 KiB	-
<input type="checkbox"/> jos_stats_agents		0	MylSAM	utf8_general_ci	1.0 KiB	-
<input type="checkbox"/> jos_templates_menu		2	MylSAM	utf8_general_ci	8.0 KiB	-
<input type="checkbox"/> jos_users		1	MylSAM	utf8_general_ci	15.1 KiB	-
<input type="checkbox"/> jos_weblinks		0	MylSAM	utf8_general_ci	1.0 KiB	-
36 table(s)	Sum	104	InnoDB	utf8_general_ci	116.5 KiB	0 B

Check All / Uncheck All With selected: ▾

Η βάση μας ονομάζεται **workhunter** και οι πινάκες που εγκαθιστά το Joomla είναι 36 όπως βλέπουμε και παραπάνω. Οι πινάκες αυτοί είναι όλοι απαραίτητοι για την εύρυθμη λειτουργία του Joomla και κατ' επέκταση και της ιστοσελίδας μας. Εμείς κυρίως θα ασχοληθούμε με το πίνακα **jos\_content** όπου θα αποθηκεύονται όλες οι αγγελίες μας και **jos\_users** όπου θα αποθηκεύονται τα στοιχεία όλων των χρηστών και εργαζομένων αλλά και εργοδοτών.

Ο πίνακας **jos\_content** έχει τα παρακάτω πεδία :

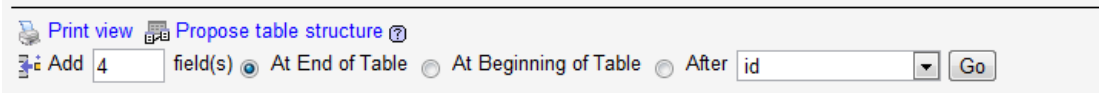
The screenshot shows the phpMyAdmin interface for the 'workhunter' database. The 'Structure' tab is active, displaying the table structure for 'jos\_content'. The table has 30 columns. The columns are:

Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> id	int(11)		UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/> title	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/> alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/> title_alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/> introtext	mediumtext	utf8_general_ci		No	None	
<input type="checkbox"/> fulltext	mediumtext	utf8_general_ci		No	None	
<input type="checkbox"/> state	tinyint(3)			No	0	
<input type="checkbox"/> sectionid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> mask	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> catid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> created	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/> created_by	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> created_by_alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/> modified	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/> modified_by	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> checked_out	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> checked_out_time	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/> publish_up	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/> publish_down	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/> images	text	utf8_general_ci		No	None	
<input type="checkbox"/> urls	text	utf8_general_ci		No	None	
<input type="checkbox"/> attribs	text	utf8_general_ci		No	None	
<input type="checkbox"/> version	int(11)		UNSIGNED	No	1	
<input type="checkbox"/> parentid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> ordering	int(11)			No	0	
<input type="checkbox"/> metakey	text	utf8_general_ci		No	None	
<input type="checkbox"/> metadesc	text	utf8_general_ci		No	None	
<input type="checkbox"/> access	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> hits	int(11)		UNSIGNED	No	0	
<input type="checkbox"/> metadata	text	utf8_general_ci		No	None	

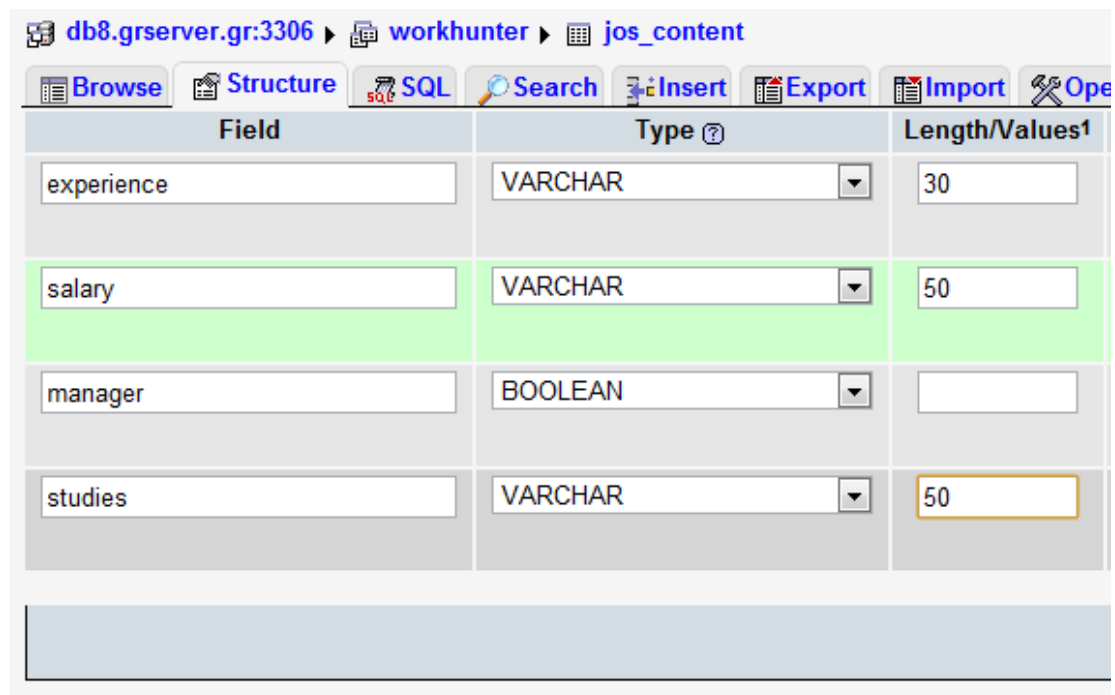
Από το παραπάνω πίνακα θα κάνουμε χρήση των περισσότερων πεδίων. Για παράδειγμα το πεδίο title θα είναι ο τίτλος της κάθε αγγελίας, το introtext θα είναι το κεντρικό περιεχόμενο της κάθε αγγελίας, το catid θα μας υποδεικνύει σε ποια κατηγορία ανήκει η αγγελία, το πεδίο hits θα μας ενημερώνει πόσες φορές έχει εμφανιστεί αυτή η αγγελία στους χρήστες, το πεδίο crated\_by από ποιον εργοδότη δημιουργήθηκε η αγγελία, το πεδίο created την ημερομηνία και την ώρα που δημιουργήθηκε η αγγελία και τα υπόλοιπα πεδία αντίστοιχα.

Για να αποθηκεύσουμε όμως τις πληροφορίες **Προϋπηρεσία, Σπουδές, Θέση προϊσταμένου** και **Μισθός** που θέλουμε να παρουσιάζονται στην αρχή κάθε αγγελίας για τη διευκόλυνση του χρήστη, έπρεπε να προσθέσουμε επιπλέον πεδία.

Για να προσθέσουμε λοιπόν αυτά τα 4 νέα στοιχεία, πηγαίνουμε στη λειτουργία προσθήκης νέων πεδίων, που βρίσκεται ακριβώς κάτω από τον πίνακα και συμπληρώνουμε τον αριθμό των πεδίων που θέλουμε να προσθέσουμε, όπως φαίνεται και παρακάτω.



Στην οθόνη που εμφανίζεται τη συμπληρώνουμε ως εξής και πατάμε Save.



Field	Type	Length/Values <sup>1</sup>
experience	VARCHAR	30
salary	VARCHAR	50
manager	BOOLEAN	
studies	VARCHAR	50

Βλέπουμε ότι συμπληρώσαμε τα πεδία με μια αντίστοιχη περιγραφή, για να κατανοούμε και αργότερα που απευθύνεται το κάθε πεδίο και επίσης έπρεπε να συμπληρώσουμε το τύπο που θα ήταν το κάθε πεδίο. Η τελική μορφή του πίνακα **jos\_content** ολοκληρώθηκε και φαίνεται παρακάτω.



	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)		UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/>	title	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	title_alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	introtext	mediumtext	utf8_general_ci		No	None	
<input type="checkbox"/>	fulltext	mediumtext	utf8_general_ci		No	None	
<input type="checkbox"/>	state	tinyint(3)			No	0	
<input type="checkbox"/>	sectionid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	mask	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	catid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	created	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	created_by	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	created_by_alias	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	modified	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	modified_by	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	checked_out	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	checked_out_time	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	publish_up	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	publish_down	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	images	text	utf8_general_ci		No	None	
<input type="checkbox"/>	urls	text	utf8_general_ci		No	None	
<input type="checkbox"/>	attribs	text	utf8_general_ci		No	None	
<input type="checkbox"/>	version	int(11)		UNSIGNED	No	1	
<input type="checkbox"/>	parentid	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	ordering	int(11)			No	0	
<input type="checkbox"/>	metakey	text	utf8_general_ci		No	None	
<input type="checkbox"/>	metadesc	text	utf8_general_ci		No	None	
<input type="checkbox"/>	access	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	hits	int(11)		UNSIGNED	No	0	
<input type="checkbox"/>	metadata	text	utf8_general_ci		No	None	
<input type="checkbox"/>	experience	varchar(30)	utf8_general_ci		No	None	
<input type="checkbox"/>	salary	varchar(50)	utf8_general_ci		No	None	
<input type="checkbox"/>	manager	tinyint(1)			No	None	
<input type="checkbox"/>	studies	varchar(50)	utf8_general_ci		No	None	

Για τον πίνακα **jos\_users** αντίστοιχα για να καλύψουμε τις απαιτήσεις που είχαμε θέσει, έπρεπε και εκεί να κάνουμε τις απαραίτητες προσθήκες πεδίων. Τα παρακάτω στοιχεία (πεδία) δεν υπήρχαν και έπρεπε να τα δημιουργήσουμε εμείς.

- ✓ τηλέφωνο
- ✓ πόλη
- ✓ οδός και αριθμός
- ✓ μικρό όνομα ενός χρήστη
- ✓ βιογραφικό
- ✓ λογότυπο

Έτσι ο πίνακας των χρηστών της ιστοσελίδας μας **jos\_users**, διαμορφώνεται τελικά ως εξής :

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	<b>name</b>	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	<b>username</b>	varchar(150)	utf8_general_ci		No		
<input type="checkbox"/>	<b>email</b>	varchar(100)	utf8_general_ci		No		
<input type="checkbox"/>	<b>password</b>	varchar(100)	utf8_general_ci		No		
<input type="checkbox"/>	<b>usertype</b>	varchar(25)	utf8_general_ci		No		
<input type="checkbox"/>	<b>block</b>	tinyint(4)			No	0	
<input type="checkbox"/>	<b>sendEmail</b>	tinyint(4)			Yes	0	
<input type="checkbox"/>	<b>gid</b>	tinyint(3)		UNSIGNED	No	1	
<input type="checkbox"/>	<b>registerDate</b>	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	<b>lastvisitDate</b>	datetime			No	0000-00-00 00:00:00	
<input type="checkbox"/>	<b>activation</b>	varchar(100)	utf8_general_ci		No		
<input type="checkbox"/>	<b>params</b>	text	utf8_general_ci		No	None	
<input type="checkbox"/>	<b>firstname</b>	varchar(30)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/>	<b>phone</b>	int(15)			Yes	NULL	
<input type="checkbox"/>	<b>town</b>	varchar(50)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/>	<b>street</b>	varchar(50)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/>	<b>streetnumber</b>	int(20)			Yes	NULL	
<input type="checkbox"/>	<b>logo</b>	varchar(100)	utf8_general_ci		Yes	NULL	
<input type="checkbox"/>	<b>bio</b>	varchar(100)	utf8_general_ci		Yes	NULL	

Τέλος, δημιουργήθηκε η ανάγκη κατασκευής δύο εξολοκλήρου δικών μας πινάκων, που έπρεπε να προσθέσουμε στη βάση δεδομένων μας. Ο πρώτος πίνακας, θα συσχετίζει τους χρήστες με τις αγγελίες, στις οποίες έχουν δώσει βιογραφικό. Με τη βοήθεια αυτού του πίνακα και με ένα σύνθετο κάλεσμα στη βάση δεδομένων που θα παρουσιάσουμε αργότερα, μπορούμε να παρουσιάσουμε στο χρήστη όλες τις αγγελίες όπου έχει δώσει το βιογραφικό του.

Για να δημιουργήσουμε αυτό το πίνακα πηγαίνουμε στη λειτουργία προσθήκης νέου πίνακα, που βρίσκεται στη κεντρική οθόνη της βάσης δεδομένων μας, όπου βρίσκονται όλοι οι πίνακες της. Συμπληρώνουμε το όνομα του πίνακα και των αριθμό των πεδίων, όπως φαίνεται παρακάτω και πατάμε Go.

 Create new table on database **workhunter**

Name:  Number of fields:

Η ονομασία του πίνακα είναι **jos\_userbio** και θα αποτελείται από 3 πεδία τα οποία παρουσιάζονται παρακάτω :

Field	id	id_user	id_advert
Type	INT	INT	INT
Length/Values1			
Default2	None	None	None
Collation			
Attributes			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	UNIQUE	---	---
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments			

Table comments:

Storage Engine: MyISAM

Collation:

PARTITION definition:

Save Or Add 1 field(s) Go

Το **id** θα είναι ένας μοναδικός αριθμός του πίνακα που θα αυξάνεται αυτόματα κατά ένα για κάθε νέα εγγραφή. Το **id\_user** θα αντιστοιχεί στο id του χρήστη, ενώ το **id\_advert** θα αντιστοιχεί στην αγγελία που έδωσε το βιογραφικό του. Κάθε φορά που ένας εργαζόμενος θα αποστέλλει το βιογραφικό του, αυτός ο πίνακας θα συμπληρώνεται αναλόγως.

Ο δεύτερος πίνακας που δημιουργήσαμε, αποσκοπεί στην αποθήκευση των σχολίων που θα πραγματοποιούν οι χρήστες μας για κάθε αγγελία. Αντίστοιχα λοιπόν κάναμε τις παρακάτω εργασίες.

Ορίσαμε το όνομα του πίνακα ως **jos\_comments**, ο οποίος θα αποτελείτε από 4 πεδία.

Create new table on database **workhunter**

Name:  Number of fields:

Go

Τον οποίων η ονομασία και ο τύπος φαίνεται αναλυτικά εδώ :

Field	Type ?	Length/Values <sup>1</sup>
id	INT	11
id_advert	INT	11
name	VARCHAR	20
comment	VARCHAR	255

Το **id** θα είναι ένας μοναδικός αριθμός του πίνακα **jcs\_comments**, που θα αυξάνεται αυτόματα κατά ένα για κάθε νέο σχόλιο. Το πεδίο **id\_advert** θα αναφέρεται στην αγγελία όπου πραγματοποιήθηκε το σχόλιο και τέλος το **name**, θα είναι το όνομα αυτού που έκανε το σχόλιο, ενώ το πεδίο **comment**, θα είναι το σχόλιο αυτό κάθε αυτό. Με αυτόν τον πίνακα και με μόνο τέσσερα πεδία, έχουμε καλύψει τη λειτουργία των σχολίων, για όλες τις αγγελίες της ιστοσελίδας μας.

Έχοντας κάνει όλες τις παραπάνω αλλαγές, η βάση δεδομένων μας είναι πλέον έτοιμη, να εξυπηρετήσει κάθε μας ανάγκη.

### 6.3 Σχεδίαση Σελίδας (template)

Αποφασίσαμε να μη χρησιμοποιήσουμε κανένα έτοιμο πρότυπο (template), αλλά να τροποποιήσουμε αυτό που εγκαθίσταται αυτόματα. Το πρότυπο αυτό είναι το **rhuk\_milkyway** και ο φάκελος του βρίσκεται στη διαδρομή ...\\www\workhunter\templates\, όπως και όλων των template άλλωστε. Το αρχείο που δομεί το κάθε πρότυπο είναι το index.php και βρίσκετε μέσα στο φάκελο του προτύπου. Έχοντας στο μυαλό μας πως θέλουμε να διαμορφωθεί η εμφάνιση της ιστοσελίδας μας, όπως είδαμε προηγούμενος στην ανάλυση, κάναμε και τις ανάλογες αλλαγές.

Στη κορυφή του index.php, έχουμε τοποθετήσει όλα τα script και css αρχεία που θα χρησιμοποιεί η ιστοσελίδα μας, για την εύρυθμη λειτουργία της. Εν συνεχεία γίνονται πάντα οι έλεγχοι για το ποια στοιχεία του κεντρικού μενού μας θα εμφανίσουμε, κάτι το οποίο θα δούμε αναλυτικότερα σε επόμενη ενότητα. Μετά το μενού εμφανίζουμε ένα μεγάλο div με id = slogan, που θα εκτείνεται σε όλο το μήκος της ιστοσελίδας μας και του οποίου το περιεχόμενο θα αλλάζει ανάλογα με τη σελίδα που βρισκόμαστε. Στη κεντρική μας σελίδα θα εμφανίζει μια εικόνα με ένα κείμενο καλωσορίσματος του επισκέπτη, ενώ σε οποιαδήποτε άλλη σελίδα θα έχει περιγραφικό χαρακτήρα, εμφανίζοντας την επικεφαλίδα χρήσης της εκάστοτε σελίδας. Αν για παράδειγμα βρισκόμαστε στην σελίδα αναζήτησης θα αναγράφει *Αναζήτηση* ενώ αν βρισκόμαστε σε μια σελίδα εγγραφής θα αναγράφει *Εγγραφή*. Ακριβώς από κάτω δημιουργήσαμε το div στο οποίο θα εμφανίζονται τα περιεχόμενα της κάθε σελίδας. Αυτό δε θα εκτείνεται σε ολόκληρο το μήκος της, αλλά θα εμφανίζεται στο κέντρο αυτής. Τέλος υλοποιήσαμε το div με id = myfooter το οποίο θα είναι κοινό για όλες τις σελίδες και θα εμπεριέχει γενικές πληροφορίες όπως το σύνολο των αγγελιών μας, τις σελίδες κοινωνικής δικτύωσης (facebook, twitter) και σύντομη περιγραφή του κεντρικού μας μενού.

Για την υλοποίηση της νέας αυτής μορφής της σελίδας μας, ήταν απαραίτητη η δημιουργία αρκετού html αλλά κυρίως css κώδικα που είναι και υπεύθυνος για την σωστή τοποθέτηση και παρουσίαση όλων των στοιχείων (html document elements) της ιστοσελίδας μας.

## 6.4 Γενικό Μενού

Η κατασκευή του μενού περιήγησης της ιστοσελίδας, κατασκευάστηκε εξολοκλήρου από εμάς και δε κάναμε χρήση κανενός module, για τη παρουσίαση του. Είναι προσβάσιμο από όλες τις σελίδες μας και αποτελείται από τις επιλογές **Κεντρική**, **Αναζήτηση**, **Επικοινωνία**, **Εγγραφή** ή **Ο Λογαριασμός μου** αν ο χρήστης έχει κάνει επιτυχημένη είσοδο στην σελίδα. Το μενού δημιουργείται από το παρακάτω κώδικα :

```
<div class="menu">
  <div id="mainNav" class="clearfix">
    <div class="menu-main-nav-container">

      <ul id="menu-main-nav" class="sf-menu sf-js-enabled sf-shadow">
        <li id="menuhome" ><a href="index.php">Κεντρική</a></li>
        <li id="menusearch" class="current_page_item"><a
href="index.php/search">Αναζήτηση</a></li>
        <li id="menucontact" ><a
href="index.php/contact">Επικοινωνία</a></li>

        <li id="menuregister" ><a >Εγγραφή</a>
```

```

<ul class="sub-menu" style="float: none; width: 8em; display: none;
visibility: hidden; ">

<li id="mploee" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="index.php/component/user/register"
style="float: none; width: auto; " class="sf-with-
ul">Εργαζόμενος</a></li>
<li id="mplayer" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="index.php/component/user/register?empl=1"
style="float: none; width: auto; " class="sf-with-
ul">Εργοδότης</a></li>

</ul>

</li>

<li id="menumyaccount" style="display:none" ><a>Λογαριασμος μου</a>
  <ul class="sub-menu" style="float: none; width: 8em; display:
none; visibility: hidden; ">

<li id="newadvert" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="http://www.contra.gr" style="float: none;
width: auto; " class="sf-with-ul">Νέα Αγγελία</a></li>

<li id="personal" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="index.php/myinfo" style="float: none;
width: auto; " class="sf-with-ul">Τα Στοιχεία μου</a></li>

<li id="myadverts" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="index.php/myadverts" style="float: none;
width: auto; " class="sf-with-ul">Οι Αγγελίες μου</a></li>

<li id="mybio" class=" " style="width: 100%; float: none; white-
space: normal; "><a href="index.php/mybio" style="float: none; width:
auto; " class="sf-with-ul">Οι Αγγελίες μου</a></li>

</ul>

</li>

</ul>

  </div>
</div>
</div>

```

Σίγουρα δε θα φαίνονται τα ίδια στοιχεία του μενού σε ένα επισκέπτη και σε ένα εγγεγραμμένο χρήστη. Ομοίως άλλες πληροφορίες παρουσιάζονται σε ένα εργοδότη και άλλες σε ένα εργαζόμενο. Η απόφαση για το ποια στοιχεία του μενού θα εμφανιστούν, γίνεται με τους παρακάτω ελέγχους :

```

<?php
    $user_id = $user->get('id');
    $user_empl = $user->get('employer');

    //Περίπτωση Επισκέπτη
    if (!$user_id)
    {
?>

<script>
    jQuery(document).ready(function() {

        jQuery('#menuregister').show();
        jQuery('#menumyaccount').hide();
    });

</script>

<?php
    }
    //Περίπτωση Εγγεγραμμένου Χρήστη
    else
    {
?>
<script>
    jQuery(document).ready(function() {

        jQuery('#menuregister').hide();
        jQuery('#menumyaccount').show();
    });

</script>
<?php
    }
    //Περίπτωση Εργαζομένου
    if ( $user_empl != 1 )
    {
?>

<script>
    jQuery(document).ready(function() {

        jQuery('#newadvert').hide();
        jQuery('#mybio').show();
        jQuery('#myadverts').hide();
    });
</script>

<?php
    }

    //Περίπτωση Εργοδότη
    if ( $user_empl == 1 )
    {
?>
<script>
    jQuery(document).ready(function() {

```

```
jQuery('#newadvert').show();
jQuery('#mybio').hide();
jQuery('#myadverts').show();
});

</script>

<?php

    }

?>
```

Είναι συνδυασμός JavaScript και Php κώδικα, με τη βοήθεια του οποίου κρύβουμε ή εμφανίζουμε τα στοιχεία του μενού, που απαιτούνται ανά περίπτωση.

## 6.5 Εγγραφή χρήστη

Όταν ένας επισκέπτης εισέλθει στην ιστοσελίδα μας, η μόνη δυνατότητα που του παρέχεται είναι η προβολή και η σχολίαση των αγγελιών. Στη περίπτωση που θέλει να τοποθετήσει μια αγγελία στη σελίδα μας, θα πρέπει να εγγραφεί ως **εργοδότης**. Αντίστοιχα αν κάποιος επισκέπτης μας, ενδιαφέρεται για μια αγγελία και θέλει να δώσει το βιογραφικό του σε αυτή, είναι υποχρεωμένος πρώτα να έχει εγγραφεί ως **εργαζόμενος**.

Είναι προφανές, όπως είχαμε αναφέρει και στην ανάλυση μας, πως χρειαζόμαστε δυο φόρμες εγγραφής για τους χρήστες μας. Η μία θα απευθύνεται στους εργοδότες και η άλλη στους εργαζομένους. Το πρόβλημα που είχαμε στη περίπτωση αυτή, είναι πως το Joomla δίνει τη δυνατότητα χρήσης, μόνο μίας φόρμας αγγελίας. Έτσι σκεφτήκαμε να μετατρέψουμε τη φόρμα αυτή δυναμικά, ανάλογα με την εγγραφή που θα πραγματοποιούσε ο χρήστης.

Το πρώτο πράγμα που έπρεπε να κάνουμε, είναι να εντοπίσουμε τη τοποθεσία της φόρμας, μέσα στους φακέλους που αυτόματα δημιούργησε το Joomla. Αυτή είναι η εξής :

**...\workhunter\components\com\_user\views\register\tmpl\default.php**

Το αρχείο αυτό λοιπόν, έπρεπε να το τροποποιήσουμε έτσι ώστε να εξυπηρετεί τελικά και τον εργαζόμενο και τον εργοδότη. Για να το πετύχουμε αυτό κάναμε



χρήση κώδικα PHP , JavaScript και jQuery Validation για τους ελέγχους των πεδίων της φόρμας μας. Επειδή ο κώδικας που υλοποιήσαμε είναι αρκετά μεγάλος, για την ευκολότερη κατανόηση του, θα χωρίσουμε τη περιγραφή σε τρεις κεντρικές ενότητες.

- ✓ Την υλοποίηση φόρμας
- ✓ Τον έλεγχο της φόρμας
- ✓ Την υποβολή των στοιχείων της φόρμας

### 6.5.1 Υλοποίηση φόρμας

Στην ενότητα αυτή θα παρουσιάσουμε, το σύνολο των πεδίων της φόρμας μας, είτε αυτά αφορούν εργοδότη είτε εργαζόμενο. Η φόρμα με όλα της τα πεδία είναι η παρακάτω :

```
<form action="" enctype="multipart/form-data" method="post"
id="register" name="register" class="myform">

<?php if ( $this->params->def( 'show_page_title', 1 ) ) : ?>
<div class="componentheading"<?php echo $this->escape($this->params-
>get('pageclass_sfx')); ?>"></div>
<?php endif; ?>

<p id='fname'>
    <label id="firstnamelbl" for="firstname">
        <?php echo JText::_ ( '* Όνομα' ); ?>:
        <span class="small">Το όνομα σου</span>
    </label>
    <input type="text" name="firstname" id="firstname" size="40"
maxlength="50" />
</p >

<p id='nme'>
    <label id="namelbl" for="name">
        <?php echo JText::_ ( '* Επώνυμο' ); ?>:
        <span id="nmsmall" class="small">Το επώνυμο σου</span>
    </label>
    <input type="text" name="name" id="name" size="40"
maxlength="50" />
</p>

<p>
    <label id="usernamelbl" for="username">
        <?php echo JText::_ ( '* Όνομα Χρήστη' ); ?>:
        <span class="small">4 έως 8 χαρακτήρες</span>
    </label>
    <input type="text" id="username" name="username" size="40"
maxlength="25" />
</p>

<p>
    <label id="pwlbl" for="password">
        <?php echo JText::_ ( '* Κωδικός Πρόσβασης ' ); ?>:
```

```

        <span class="small">4 έως 10 χαρακτήρες</span>
    </label>
<input type="password" id="password" name="password" size="40" />
</p>

<p>
    <label id="pw2lbl" for="password2">
        <?php echo JText::_ ( '* Έλεγχος Κωδικού ' ); ?>:
        <span class="small">Γράψε ξανά τον κωδικό</span>
    </label>
    <input type="password" id="password2" name="password2"
    size="40" />
</p>

<p>
    <label id="emailmsg" for="email">
        <?php echo JText::_ ( '* E-mail' ); ?>:
        <span class="small">Δώσε έγκυρο e-mail</span>
    </label>
    <input type="text" id="email" name="email" size="40"
    maxlength="100" />
</p>

<p>
    <label id="emailblcheck" for="email2">
        <?php echo JText::_ ( '* Έλεγχος E-mail' ); ?>:
        <span class="small">Γράψε ξανά το e-mail</span>
    </label>
    <input type="text" id="email2" name="email2" size="40"
    maxlength="100" />
</p>

<p>
    <label id="phonelbl" for="phone">
        <?php echo JText::_ ( 'Τηλέφωνο' ); ?>:
        <span class="small">Σταθερό ή κινητό</span>
    </label>
    <input type="text" id="phone" name="phone" size="40" />
</p>

<p>
    <label id="townlbl" for="town">
        <?php echo JText::_ ( 'Πόλη' ); ?>:
        <span class="small">Η πόλη διαμονής σου</span>
    </label>
    <input type="text" id="town" name="town" size="40" />
</p>

<p>
    <label id="streetlbl" for="street">
        <?php echo JText::_ ( 'Οδός' ); ?>:
        <span class="small">Η οδός διευθύνσεως</span>
    </label>
    <input type="text" id="street" name="street" size="40" />
</p>

<p>
    <label id="numberlbl" for="streetnumber">
        <?php echo JText::_ ( 'Αριθμός Οδού' ); ?>:

```

```

        <span class="small">Ο αριθμός διεύθυνσης</span>
    </label>
    <input type="text" id="streetnumber" name="streetnumber"
    size="40" />
</p>

<p id='lg'>
    <label id="logobl" for="logo">
        <?php echo JText::_ ( '* Λογότυπο Εταιρείας' ); ?>:
        <span class="small">Επιλέξτε φωτογραφία</span>
    </label>
    <input type="file" id="logo" name="logo"
    onChange="validatelogo()" />
</p>

<p id='bi'>
    <label id="biobl" for="bio">
        <?php echo JText::_ ( '* Το Βιογραφικό σου' ); ?>:
        <span class="small">Επιλέξτε αρχείο</span>
    </label>
    <input type="file" id="bio" name="bio"
    onChange="validateform()" />
</p>

```

Σε κάθε πεδίο υπάρχει η ονομασία του και από κάτω μια μικρή περιγραφή, η οποία τις περισσότερες φορές υποδηλώνει κάποιους περιορισμούς. Τα κοινά πεδία εργαζομένου και εργοδότη είναι τα εξής :

- ✓ Επώνυμο
- ✓ Όνομα Χρήστη
- ✓ Κωδικός Πρόσβασης
- ✓ Έλεγχος Κωδικού
- ✓ E-mail
- ✓ Έλεγχος
- ✓ Τηλέφωνο
- ✓ Πόλη
- ✓ Οδός
- ✓ Αριθμός Οδού

Το πεδίο **όνομα** και **βιογραφικό** βρίσκεται μόνο στον εργαζόμενο ενώ το πεδίο **λογότυπο** μόνο στον εργοδότη αντίστοιχα. Το πεδίο επώνυμο ( με id = name ) παρόλο που είναι κοινό μετονομάζεται. Στη περίπτωση του εργαζομένου μένει όπως είναι, ενώ για τον εργοδότη μετατρέπεται αυτόματα σε **επωνυμία**.

Τα υποχρεωτικά πεδία είναι διαφορετικά σε κάθε εγγραφή. Έτσι παρόλο που ένα πεδίο είναι κοινό, στη μία περίπτωση μπορεί να είναι υποχρεωτικό και την άλλη όχι. Πως υλοποιούνται αυτοί οι έλεγχοι και όχι μόνο θα παρουσιάσουμε στην επόμενη ενότητα.

## 6.5.2 Έλεγχος Φόρμας

Στην ενότητα αυτή θα παρουσιάσουμε, πως γίνεται τελικά ο διαχωρισμός της εγγραφής του εργαζόμενου και του εργοδότη, με τη χρήση μίας και μόνο φόρμας.

Ο διαχωρισμός αυτός υλοποιείται, με τη χρήση μιας μεταβλητής **empl** στο query string της σελίδας εγγραφής μας . Αν ο χρήστης επιλέξει να κάνει εγγραφή σαν εργοδότη, θα εισέλθει στη σελίδα **/index.php/component/user/register?empl=1** ενώ αν επιλέξει εγγραφή για εργαζόμενο, θα δρομολογηθεί στη σελίδα **/index.php/component/user/register**. Με τον τρόπο αυτό γνωρίζουμε ο χρήστης τι εγγραφή θέλει να πραγματοποιήσει, εμφανίζοντας του και τα αντίστοιχα πεδία . Πώς υλοποιείται αυτό φαίνεται παρακάτω :

```
<script>

    jQuery(document).ready(function() {

        jQuery.validator.addMethod("lettersonly", function(value,
element) {
            return this.optional(element) || /^[^\x00-\x80]|[a-zA-Z\
\'']+$/.test(value);}, "Μόνο γράμματα επιτρέπονται");

        jQuery("#register").validate({
            <?php if ($_GET['empl'] == 1) : ?>
            rules:
            {
                name:{ required: true , lettersonly: true },
                username:{ required: true , minlength: 4 , maxlength: 8
            },
                password:{ required: true , minlength: 4 , maxlength: 8
            },
                password2:{ required: true, equalTo: "#password" },
                email:{ required: true , email: true },
                email2:{ required: true , email: true , equalTo: "#email"
            },
                phone:{ required: true , number: true , minlength: 10},
                streetnumber:{ required: true , number: true},
                street:{ required: true , lettersonly: true},
                town:{ required: true , lettersonly: true},
                logo:{ required: true , accept: "jpg|png" }

            },<?php endif; ?>
            <?php if ($_GET['empl'] != 1) : ?>
            rules:
            {
                name:{ required: true , lettersonly: true },
                firstname:{ required: true , lettersonly: true },
                username:{ required: true , minlength: 4 , maxlength: 8
            },
                password:{ required: true , minlength: 4 , maxlength: 8
            },
                password2:{ required: true, equalTo: "#password" },
                email:{ required: true , email: true },
                email2:{ required: true , email: true , equalTo: "#email"
            },
                phone:{number: true , minlength: 10},
```

```

        town:{ letteronly: true},
        streetnumber:{number: true},
        street:{letteronly: true},
        bio:{ required: true , accept: "pdf"}
    },
<?php endif; ?>

messages:{
    firstname:{
        required: "Το πεδίο είναι απαραίτητο" ,
    },

    name:{
        required: "Το πεδίο είναι απαραίτητο" ,
    },

    username:{
        required: "Το πεδίο είναι απαραίτητο" ,
        minlength : "Εισάγετε τουλάχιστον 4 χαρακτήρες",
        maxlength : "Εισάγετε έως 8 χαρακτήρες"
    },

    password:{
        required: "Το πεδίο είναι απαραίτητο" ,
        minlength : "Εισάγετε τουλάχιστον 4 χαρακτήρες",
        maxlength : "Εισάγετε έως 8 χαρακτήρες"
    },

    password2:{
        required: "Το πεδίο είναι απαραίτητο" ,
        equalTo : "Ο κωδικός δεν είναι ο ίδιος"
    },

    email:{
        email : "Πληκτρολογήστε έγκυρο E-mail.",
        required: "Το πεδίο είναι απαραίτητο"
    },

    email2:{
        email : "Πληκτρολογήστε έγκυρο E-mail.",
        required: "Το πεδίο είναι απαραίτητο" ,
        equalTo : "Το E-mail δεν είναι το ίδιο"
    },

    phone:{
        required: "Το πεδίο είναι απαραίτητο" ,
        number : "Μόνο αριθμοί επιτρέπονται",
        minlength : "Εισάγετε τουλάχιστον 10 χαρακτήρες",
    },

    streetnumber:{
        required: "Το πεδίο είναι απαραίτητο" ,
        number : "Μόνο αριθμοί επιτρέπονται"
    },

    street:{
        required: "Το πεδίο είναι απαραίτητο" ,
    },

```

```

        town:{
        required: "Το πεδίο είναι απαραίτητο" ,
        letteronly: true
        },
        logo:{
        required: " Το πεδίο είναι απαραίτητο "
        },
        bio:{
        required: " Το πεδίο είναι απαραίτητο "
        }
    }
    });
});
</script>

```

Το πρώτο πράγμα που ελέγχεται πριν την εμφάνιση της φόρμας, είναι αν η μεταβλητή **empl** ισούται με ένα ή όχι. Με βάση την τιμή αυτής της μεταβλητής, αποφασίζουμε ποια πεδία θα εμφανίσουμε και ποια από αυτά θα είναι υποχρεωτικά. Επίσης για κάθε πεδίο μπορούμε να ορίσουμε το πεδίο τιμών του και το είδος του (αν για παράδειγμα πρέπει να είναι μόνο αριθμητικό ή αν αντιπροσωπεύει ηλεκτρονικό ταχυδρομείο κ.α.). Τέλος σε περίπτωση λάθους, έχουμε ορίσει και το αντίστοιχο μήνυμα που θα εμφανίζεται, έτσι ώστε να ενημερώνεται για το λάθος του και εν συνεχεία να το διορθώνει. Όλα τα παραπάνω υπολήφθηκαν με τη χρήση της **JavaScript** βιβλιοθήκης **jQuery**.

### 6.5.3 Υποβολή στοιχείων της φόρμας

Αφού πλέον ο χρήστης έχει συμπληρώσει σωστά όλα τα υποχρεωτικά πεδία της φόρμας μας, θα πρέπει να πατήσει το κουμπί **εγγραφή** το οποίο και θα αποθηκεύσει τα δεδομένα στη βάση μας. Αυτό υλοποιείτε στο τέλος της φόρμας μας.

```

<button class="button" type="submit"
<?php echo JText::_('Register'); ?>
</button>

```

Τέλος, έχουμε ένα κρυφό πεδίο με ονομασία **employer** και στο οποίο εκχωρείτε αυτόματα η τιμή ένα, μόνο στη περίπτωση που ο χρήστης πραγματοποιήσει εγγραφή εργοδότη.

Εν συνεχεία έπρεπε να βρούμε το αρχείο στο οποίο πραγματοποιείται η υποβολή των στοιχείων της φόρμας και η αποθήκευση τους τελικά στη βάση δεδομένων μας. Το αρχείο αυτό είναι :

**...\\www\\workhunter\\components\\com\_user\\controller.php**

Στο αρχείο αυτό, η μέθοδος **register\_save( )** εκτελείται αμέσως μετά την υποβολή των στοιχείων από εμάς. Δρομολογούμαστε εκεί, μέσω του κρυφού πεδίου με ονομασία **task** και τιμή **register\_save** που βρίσκεται στο τέλος της φόρμας (όπως φαίνεται και παραπάνω).

Στο αρχείο αυτό έχουμε τοποθετήσει δύο μεθόδους την **save\_bio** και την **save\_logo** για την αποθήκευση του βιογραφικού και του λογότυπου αντίστοιχα. Ποια από τις δυο θα εκτελείται κάθε φορά, αποφασίζεται σύμφωνα με τον έλεγχο της μεταβλητής **employer** της φόρμας μας.

```
$empl= JRequest::getVar('employer', null, 'post', 'string');

if ( $empl == 1)
{
    $this->save_logo($user);
}
else
{
    $this->save_bio($user);
}
```

Με τη χρήση των μεθόδων αυτών (**save\_logo** και **save\_bio**) το λογότυπο ή το βιογραφικό αντίστοιχα, τοποθετείτε σε ένα συγκεκριμένο φάκελο που έχουμε ορίσει εμείς, ενώ παράλληλα αποθηκεύεται η ονομασία του αρχείου, στο αντίστοιχο πεδίο της βάσης μας(**logo** ή **bio**). Οι μέθοδοι παρουσιάζονται εδώ :

```
function save_logo($user)
{

    $db          =& JFactory::getDBO();
    $file = JRequest::getVar('logo', null, 'files', 'array');

    jimport('joomla.filesystem.file');

    $filename = $file['name'];

    $src = $file['tmp_name'];
    //Ορισμός τοποθεσίας αποθήκευσης του αρχείου
    $dest = JPATH_COMPONENT . DS . "logos" . DS . $filename;

    if ( $file['size'] < 20000000 )

    {

        if ( JFile::upload($src, $dest) ) {
            //Αποθήκευση ονομασίας του αρχείου στη βάση μας τοποθεσίας
            $user->set('logo', $filename);
        } else {
            //Redirect and throw an error message
        }
    }

}
```

```

function save_bio($user)
{
    $db          =& JFactory::getDBO();
    $file = JRequest::getVar('bio', null, 'files', 'array');

    jimport('joomla.filesystem.file');

    $filename = $file['name'];

    $src = $file['tmp_name'];
    //Ορισμός τοποθεσίας αποθήκευσης του αρχείου
    $dest = JPATH_COMPONENT . DS . "bio" . DS . $filename;

    if ( $file['size'] < 20000000 )
    {
        if ( JFile::upload($src, $dest) ) {
            //Αποθήκευση ονομασίας του αρχείου στη βάση μας τοποθεσίας
            $user->set('bio', $filename);
        }
        else {
            //Redirect and throw an error message
        }
    }
}

```

Όταν πλέον η ολοκλήρωση της εγγραφής έχει στεφθεί με επιτυχία, γίνεται ανακατεύθυνση του χρήστη στην κεντρική μας σελίδα, έχοντας εμφανιστεί παράλληλα και ένα πληροφοριακό μήνυμα. Το μήνυμα αυτό ενημερώνει το χρήστη, πως ένας **Σύνδεσμος Ενεργοποίησης** έχει αποσταλεί στη διεύθυνση ηλεκτρονικού ταχυδρομείου που δήλωσε, με τη χρήση του οποίου θα ενεργοποιηθεί ο λογαριασμός του.

## 6.6 Αλλαγή στοιχείων χρήστη

Ο εγγεγραμμένος πλέον χρήστης της σελίδας μας, είτε εργαζόμενος είτε εργοδότης έχει τη δυνατότητα της τροποποίηση των στοιχείων του. Αυτή βρίσκεται στο μενού **Ο Λογαριασμός μου** στην επιλογή **Τα στοιχεία μου**. Η λογική της υλοποίηση και εδώ είναι παρόμοια με αυτή της εγγραφής. Υπάρχουν όμως και αρκετές διαφορές που και θα παρουσιάσουμε στη συνέχεια.



Το αρχείο όπου πραγματοποιείται η αλλαγή των στοιχείων του χρήστη και το οποίο εμείς τροποποιήσαμε, βρίσκεται στη παρακάτω τοποθεσία :

... `www\workhunter\components\com_user\views\user\templ\form.php`

Σχεδόν όλα τα πεδία της φόρμας μας πρέπει να εμφανίζουν τις τιμές που είχε καταχωρήσει ο χρήστης κατά την εγγραφή του. Αυτό επιτυγχάνεται με τη χρήση του `echo $this->escape( $this->user->get( 'όνομα βάσης του πεδίου' ) )` ως τιμή (value) στα πεδία που χρειάζεται. Για παράδειγμα το πεδίο Όνομα Χρήστη θα έχει πλέον την εξής μορφή :

```
<p>
    <label id="usernamelbl" for="username">
        <?php echo JText::_ ( '* Όνομα Χρήστη' ); ?>:
        <span class="small">4 έως 8 χαρακτήρες</span>
    </label>
    <input type="text" id="username" name="username" size="40"
value="<?php echo $this->escape( $this->user->get ( 'username' ) ); ?>"
maxlength="25" />
</p>
```

Τα πεδία που δε χρειάζεται να έχουν καμία τιμή είναι ο κωδικός, ο έλεγχος κωδικού και ο έλεγχος του ηλεκτρονικού ταχυδρομείου. Αυτά τα πεδία πρέπει να συμπληρωθούν ξανά για λόγους ασφαλείας και για να διασφαλίσουμε την ορθότητα των στοιχείων που θα καταχωρήσουμε στη βάση δεδομένων μας. Το πεδίο του ονόματος χρήστη απαγορεύεται να αλλαχθεί και ακλουθεί το χρήστη για πάντα.

Το μόνο νέο πεδίο που προστίθεται στο τέλος της φόρμας μας, προορίζεται για την υποβολή νέου βιογραφικού ή λογότυπου αντίστοιχα. Στη θέση που γινόταν η υποβολή τους κατά την εγγραφή, πλέον εμφανίζονται στο χρήστη τα αρχεία που μας είχε καταχωρήσει.

Όλοι οι έλεγχοι (validation) των πεδίων της φόρμας μας παραμένουν οι ίδιοι με αυτούς της εγγραφής. Ο έλεγχος όμως για το **ποια** πεδία θα εμφανίσουμε, γίνεται πλέον με το έλεγχο της μεταβλητής **employer**. Αν αυτή έχει τιμή ένα, τότε αναφερόμαστε σε εργοδότη ενώ σε οποιαδήποτε άλλη περίπτωση σε εργαζόμενο

Επίσης μια άλλη σημαντική διαφορά είναι πως το κρυφό πεδίο με ονομασία **task** έχει τιμή **save**. Αυτό σημαίνει πως κατά την υποβολή ( submit ) των στοιχείων της φόρμας μας, στο αρχείο **controller.php** (που είναι υπεύθυνο για την αποθήκευση των στοιχείων αυτών) θα εκτελεστεί η μέθοδος `save( )`. Εδώ έχουμε χρησιμοποιήσει ξανά τις μεθόδους **save\_bio** και **save\_logo** για την ορθή αποθήκευση του βιογραφικού και του λογότυπου εάν αυτά άλλαξαν.

Και εδώ όταν η ενημέρωση ολοκληρωθεί επιτυχώς, γίνεται ανακατεύθυνση του χρήστη στην κεντρική μας σελίδα, έχοντας εμφανιστεί παράλληλα και ένα μήνυμα επιτυχίας αποθήκευσης των δεδομένων.

## 6.7 Αναζήτηση

Η αναζήτηση στην ιστοσελίδα μας αποφασίστηκε να υλοποιηθεί από εμάς εξολοκλήρου. Η έννοια της αναζήτησης εμφανίζεται σε παραπάνω από ένα σημεία της σελίδας μας, όπως θα δούμε και παρακάτω και αυτό ήταν τελικά που μας οδήγησε στην απόφαση της υλοποίησης. Για την κατασκευή της αναζήτησης, χρησιμοποιήσαμε κώδικα Php και JavaScript.

Ξεκινώντας θα παρουσιάσουμε την υλοποίηση **Αναζήτησης Αγγελίας** από ένα επισκέπτη της ιστοσελίδας μας. Επιλέγοντας από το κύριο μενού την επιλογή αναζήτηση, εμφανίζεται η αντίστοιχη φόρμα.

Ο κώδικας της φόρμας είναι ο παρακάτω :

```
<form style="border: 1px #DDDDDE solid; padding:35px" id="category"
name="category" action = "" method = "POST">

<input style="margin-right:10px" type = "text" name = "keywords" />

<select style="margin-right:30px" name="category" id ="ad_category" >
  <option value="" selected> Επιλέξτε Κατηγορία</option>
</select>

<select style="margin-right:30px" name="short">
  <option value="" selected> Επιλέξτε Ταξινόμηση</option>
  <option value="modified">Νεότερο</option>
  <option value="hits">Δημοφιλέστερο</option>
  <option value="title">Αλφαβητικά</option>
</select>

<input style="margin-left:40px" type = "submit" Value = "Search" />

</form>
```

Παρατηρούμε πως ο χρήστης, πρέπει να δώσει μία **λέξη κλειδί** ( keyword ) με βάση την οποία θα υλοποιηθεί η αναζήτηση. Ο χρήστης επίσης, έχει τη δυνατότητα να βελτιστοποιήσει τα αποτελέσματά του, επιλέγοντας τα πεδία **κατηγορίας** και **ταξινόμησης**.

Η **ταξινόμηση** των αποτελεσμάτων όπως φαίνεται και στο κώδικα μπορεί να γίνει με τρεις τρόπους.

- ✓ Το πιο νέο
- ✓ Το πιο δημοφιλές (έχει τις περισσότερες εμφανίσεις)
- ✓ Αλφαβητικά

Η επιλογή **κατηγορίας** είναι το επόμενο κριτήριο, με βάση το οποίο ο χρήστης κάνει τα αποτελέσματα του πιο συγκεκριμένα, το οποίο όμως στο παραπάνω κώδικα παραμένει κενό. Αν είχαμε τοποθετήσει στατικά εδώ τις κατηγορίες, μια πιθανή προσθήκη νέας, θα μας ήταν αρκετά πολύπλοκη. Δε θα έπρεπε μόνο να την προσθέσουμε από το διαχειριστικό μέρος του Joomla, αλλά θα έπρεπε να ψάξουμε σε όλη την ιστοσελίδα μας για να τη προσθέσουμε και όπου αλλού χρειαστεί.

Για το λόγο αυτό λοιπόν, υλοποιήσαμε το παρακάτω κώδικα ο οποίος γεμίζει τις κατηγορίες δυναμικά. Αν για παράδειγμα, παρουσιαστεί η ανάγκη προσθήκης μιας νέας κατηγορίας για την ιστοσελίδα μας, αυτή θα περαστεί μία μόνο φορά από το διαχειριστικό του Joomla και θα έχει εμφανιστεί αυτόματα στη επιλογή μας (select).

```
<?php

$sql = "SELECT * FROM jos_categories";
$results_number = ($results = mysql_query($sql)) ?
mysql_num_rows($results) : 0 ;

    if ($results_number == 0)
    {
        echo "Δεν υπάρχει κατηγορία";
    }
    else
    {
        while ($results_row = mysql_fetch_assoc($results))
        {
            ?>

<script>
jQuery(document).ready(function()
{
    jQuery("#ad_category").append("<option value='<?php echo
$results_row['id'];?>'><?php echo
$results_row['title'];?></option>");

});
</script>
<?php
    }
}

?>
```

Είναι λίγο σύνθετος κώδικας και περιέχει script γραμμένο εσωτερικά της Php. Αναλυτικότερα στη συγκεκριμένη περίπτωση επιλέγουμε όλα τα στοιχεία από το πίνακα **jos\_categories** όπου υπάρχουν όλες οι κατηγορίες. Σε περίπτωση που το SQL ερώτημα μας δεν έχει αποτελέσματα, τυπώνουμε ότι δεν υπάρχει κατηγορία.

Στη περίπτωση όμως που βρούμε αποτελέσματα, τα αποθηκεύουμε σε ένα πίνακα και τα τοποθετούμε με τη βοήθεια της jQuery ως επιλογές, στο **select element** της φόρμας μας με `id = ad_category`. Ως τιμή της κάθε επιλογής, ορίζουμε το `id` της κατηγορίας ενώ για περιγραφή το τίτλο της.

Το τελικό αποτέλεσμα φαίνεται παρακάτω. Και το ανακαλύπτουμε βλέποντας το πηγαίο κώδικα (source code) της ιστοσελίδας μας, με τη βοήθεια ενός περιηγητή (browser).

```
<select style="margin-right: 30px; opacity: 0;" name="category"
id="ad_category">

    <option value="" selected=""> Επιλέξτε Κατηγορία</option>
    <option value="1">Πληροφορική / Τηλεπικοινωνίες</option>
    <option value="2">Πρακτική Άσκηση</option>
    <option value="3">ΜΜΕ / Δημοσιογράφοι</option>
    <option value="4">Ανθρώπινο Δυναμικό</option>
    <option value="5">Λογιστήριο</option>
    <option value="6">Ασφάλειες</option>
    .
    .
    .
    .
    .

</select>
```

Αφού πλέον ο χρήστης έχει γράψει τη λέξη κλειδί που θέλει να αναζητήσει, έχει επιλέξει και όποια από τα κρατήρα επιθυμεί, πατάει το κουμπί **αναζήτηση** και αναμένει για τα αποτελέσματα. Ας δούμε λοιπόν, πως υλοποιήσαμε το ζωτικότερο κομμάτι της αναζήτησής μας και να το αναλύσουμε στη συνέχεια.

```
<?php

include "search_results.php";
include "categories.php";

if (isset($_POST['keywords']))
{
    $keywords =
mysql_real_escape_string(htmlspecialchars(trim($_POST['keywords']),
ENT_QUOTES , "UTF-8"));

    $errors = array();

    if (empty($keywords))
    {
        $errors[] = "Παρακαλώ εισάγετε μία φράση αναζήτησης.";
    }
    else if (strlen($keywords)<3)
    {
```

```

        $errors[] = "Η αναζήτηση σας πρέπει να αποτελείτε απο 3 ή
περισσότερους χαρακτήρες.";
    }
    else if (search_results($keywords) === false)
    {
        $errors[] = "Η αναζήτηση σου για
<strong>".$keywords."</strong> δεν επέστρεψε κανένα αποτέλεσμα.";
    }

    if (empty($errors))
    {
        $results = search_results($keywords);
        $results_num = count($results);

        echo "<p><br> Η αναζήτηση για <strong> $keywords </strong>
επέστρεψε <strong> $results_num </strong> αποτελέσματα.</p><br><br>";

        foreach ($results as $result)
        {
            $userid = $result['created_by'];
            $sql1="SELECT logo FROM jos_users WHERE
id=$userid";

            if ($resultlogo=mysql_query($sql1))
            {
                if ($row=mysql_fetch_assoc($resultlogo))
                {
                    $logo = $row[logo];
                }
            }

            echo '<div class = "searchresults"> <table> <tr> <td valign="top"
class="searchcontent" width="100%">', '<a style=" font-size:2em;
color:gray;"
href="/index.php?option=com_content&view=article&id=', $result['id'], '
">', $result['title'] , '</a>', '<span style="margin-left:60px"> Hits:
' , $result['hits'] , '<a style=" font-size:2em; color:coral; font-
family: Segoe UI Light"
href="/index.php?option=com_jumi&fileid=4&id=', $result['id'], '">',
"EDIT" , '</a>', ' </span> <p class="createdate"> test</p></td> <td>
 </td> </tr> </table> <br><br></div>  ';
        }
    }
    else
    {
        foreach ($errors as $error)
        {
            echo $error, '</br>';
        }
    }
}

?>

```

Στην αρχή του κώδικα κάνουμε τους απαραίτητους ελέγχους στη φράση αναζήτησης (keyword) που πληκτρολογεί ο χρήστης. Αυτό είναι απαραίτητο γιατί η συγκεκριμένη φράση, τοποθετείτε στο ερώτημα(sql query) που κάνουμε στην βάση δεδομένων μας, έτσι ώστε να έχουμε τα αποτελέσματα.

Με την εντολή **mysql\_real\_escape\_string( )** παραβλέπουμε τους ειδικούς χαρακτήρες (\x00, \n, \r, \, ', " και \x1a) εάν φυσικά αυτοί υπάρχουν. Έπειτα με τη μέθοδο **htmlspecialchars( )** ελέγχουμε, εάν η φράση εμπεριέχει κώδικα html. Στη περίπτωση που υπάρχει, τότε αυτός μεταφέρεται αυτούσιος στη φράση αναζήτησης μας, χωρίς να μεταφραστεί η λειτουργία του. Τέλος με τη μέθοδο **trim( )**, ελέγχουμε αν η φράση αναζήτησης εμπεριέχει κενά(whitespace) στην αρχή της. Αν έχει, τότε όσα και αν είναι αυτά απαλείφονται.

Στη περίπτωση που δεν έχει τοποθετήσει ο χρήστης καμία φράση αναζήτησης ή αυτή είναι μικρότερη από τρεις χαρακτήρες, εμφανίζεται και το αντίστοιχο μήνυμα λάθους. Εάν ο χρήστης έχει τοποθετήσει πάνω από 3 χαρακτήρες στη φράση αναζήτησης του και πατήσει το κουμπί **αναζήτηση**, θα εκτελεστεί ο παρακάτω κώδικας :

```
<?php
```

```
function search_results($keywords)
{

    $returned_results = array();
    $where = "";

    $keywords = preg_split('/[\s]+/', $keywords);
    $total_keywords = count($keywords);

    foreach ($keywords as $key=>$keyword)
    {

        $where .= "title LIKE '%$keyword%'";
        if ($key != ($total_keywords - 1))
        {
            $where .= " AND ";
        }
    }

    if ($_POST['category'] != null) { $cid = $_POST['category']; $where
    .= " AND catid = $cid "; }

    if ($_POST['short'] != null) { $order = $_POST['short']; } else {
    $order = "modified";}

    $sql = "SELECT * FROM jos_content WHERE $where ORDER BY $order
    DESC ";
```

```

    $results_number = ($results = mysql_query($sql)) ?
mysql_num_rows($results) : 0 ;

    if ($results_number == 0)
    {
        return false;
    }
    else
    {
        while ($results_row = mysql_fetch_assoc($results))
        {
            $returned_results[] =
array('title'=>$results_row['title'],

'hits'=>$results_row['hits'],

'id' =>
$results_row['id'],

'introtext'=>$results_row['introtext'],

'created_by'=>$results_row['created_by']);
        }

        return $returned_results;
    }
}
?>

```

Εδώ με την εντολή **preg\_split('/[s]+/', \$keywords)**; χωρίζουμε τη φράση σε λέξεις. Αν στη φράση υπάρχουν παραπάνω από μία λέξεις και αυτές χωρίζονται με κόμμα ή κενό τότε τοποθετούνται σε ένα πίνακα. Χρησιμοποιούμε στη συνέχεια το πίνακα αυτόν, δημιουργώντας μια νέα φράση με τις ίδιες λέξεις κλειδιά, έχοντας όμως τώρα τοποθετήσει το **AND** ανάμεσά τους. Έτσι η φράση μας, που έχει αποθηκευτεί στη μεταβλητή **where**, είναι έτοιμη να τοποθετηθεί στο ερώτημα μας. Το ερώτημα θα αναζητήσει τη φράση, στον τίτλο όλων των ενεργών μας αγγελιών. Αν έχουμε επιλέξει συγκεκριμένη κατηγορία, τότε τα αποτελέσματα μας θα αφορούν μόνο αυτή, ενώ αν δεν επιλέξαμε καμία, τα αποτελέσματα θα αφορούν όλες τις κατηγορίες. Η σειρά με την οποία θα εμφανιστούν τα αποτελέσματα, επίσης επιλέγεται από εμάς. Σε περίπτωση που δεν θέσαμε κριτήριο ταξινόμησης, το προεπιλεγμένο κριτήριο είναι η χρονολογική σειρά. Τέλος, σε περίπτωση που δεν βρεθούν αποτελέσματα ο χρήστης ενημερώνεται με το αντίστοιχο μήνυμα. Αν όμως βρεθούν, τότε αυτά αποθηκεύονται σε ένα νέο πίνακα και επιστρέφουν για να τα εμφανίσουμε με τον τρόπο που εμείς επιθυμούμε.

Η έννοια της αναζήτησης εμφανίζεται και στους εγγεγραμμένους χρήστες, διαφορετικά στον εργαζόμενο και διαφορετικά στον εργοδότη. Από το μενού *Ο Λογαριασμός μου* και στην επιλογή **Οι αγγελίες μου** ο εργοδότης έχει τη δυνατότητα να δει ποιες αγγελίες έχει καταχωρήσει, ενώ αντίστοιχα ο χρήστης τύπου εργαζόμενος βλέπει σε ποιες αγγελίες έχει δώσει το βιογραφικό του.

Ας δούμε τον κώδικα που εκτελείται στη περίπτωση του εργοδότη και πως αυτός διαφοροποιείται σε σχέση με του εργαζόμενου :

```
<?php

function search_results( )
{
    $user =& JFactory::getUser();

    $id_employer = $user->get('id');

    $order = "created";

    $sql = "SELECT * FROM jos_content WHERE created_by = $id_employer
ORDER BY $order DESC";

    $results_number = ($results = mysql_query($sql)) ?
mysql_num_rows($results) : 0 ;

    if ($results_number == 0)
    {
        return false;
    }
    else
    {
        while ($results_row = mysql_fetch_assoc($results))
        {
            $returned_results[] =
array('title'=>$results_row['title'],
'hits'=>$results_row['hits'],
'id' =>
$results_row['id'],
'introtext'=>$results_row['introtext']);
        }

        return $returned_results;
    }
}

if ( $user->get('id') == 0 || $user->get('employer') == 0)
{
    $errors = "Δέν υπάρχουν αποτελέσματα είτε γιατί δεν είσαι
εργοδότης είτε γιατί δεν έχεις πραγματοποιήσει Είσοδο (Log In) ";
    echo $errors;
}
```



```

}

else
{
    if (empty($errors))
    {
        $results = search_results ( );

        if (empty($results))
        {
            echo "Δέν έχετε καμία ενεργή Αγγελία";
        }

        else
        {
            $results_num = count($results);
            echo "<p><br> Οι ενεργές σας αγγελίες είναι <strong>
$results_num </strong> .</p><br><br>";

            foreach ($results as $result)
            {
                echo '<div style="border: 1px #DDDDDE solid; margin-
left:50px; padding-left:40px; line-height:60px; height:65px;
width:800px; -moz-box-shadow: 1px 1px 1px #DDDDDE;-webkit-box-shadow:
1px 1px 1px #DDDDDE;box-shadow: 1px 1px 1px #DDDDDE;">','<a style="
font-size:2em; color:coral; font-family: Segoe UI Light"
href="joomla15/index.php?option=com_content&view=article&id=', $result
['id'],'">', $result['title'] , '</a>', '<span> Hits: ',
$result['hits'] , '<a style=" font-size:2em; color:coral; font-family:
Segoe UI Light"
href="/index.php?option=com_jumi&fileid=5&id=', $result['id'],'">',
"EDIT" , '</a>', ' </span>', ' </div> <br><br>';
            }
        }
    }
}

?>

```

Ελέγχουμε αν υπάρχει συνδεδεμένος χρήστης στη σελίδα μας και αν αυτός υπάρχει να είναι τύπου εργοδότη. Αν ο έλεγχος είναι επιτυχής, εκτελείται η μέθοδος **search\_results( )** με την οποία αναζητάμε αγγελίες στη βάση **jos\_content**, με το **id** του συγκεκριμένου χρήστη να είναι ίδιο με εκείνου που δημιούργησε την αγγελία(**created\_by**). Αν το αποτέλεσμα που επιστρέφει είναι κενό, τότε εμφανίζουμε πως δεν έχει καμία αγγελία καταχωρημένη σε εμάς. Αν η μέθοδος επιστρέψει αποτελέσματα τότε τα εμφανίζουμε στο χρήστη.

Για το χρήστη τύπου εργαζόμενος κάνουμε τους αντίστοιχους ελέγχους και το μόνο που αλλάζει είναι το ερώτημα στη βάση δεδομένων μας. Εδώ ελέγχουμε από τον πίνακα **jos\_userbio** το πεδίο **id\_advert** να ισούται με το πεδίο **id** του πίνακα

**jos\_content**, μόνο όμως για τις περιπτώσεις όπου το πεδίο **id\_user** του πίνακα **jos\_userbio** είναι ίσο με το **id** του χρήστη που έχει **συνδεθεί**. Αυτό το πολύπλοκο ερώτημα στη βάση δεδομένων μας εμφανίζεται παρακάτω :

```
$sql = "SELECT * FROM jos_userbio INNER JOIN jos_content ON
jos_userbio.id_advert = jos_content.id WHERE jos_userbio.id_user =
$id_employee ORDER BY $order DESC";
```

Η συχνή παρουσία και η πολυπλοκότητα της **αναζήτησης**, όπως φάνηκε και στα παραπάνω παραδείγματα, ήταν ο λόγος που μας οδήγησε στην εκ νέου δημιουργία της συγκεκριμένης λειτουργίας, παρόλο που το Joomla τη διαθέτει.

## 6.8 Εισαγωγή / Ενημέρωση Αγγελίας

Το προνόμιο της καταχώρησης και ενημέρωσης μιας αγγελίας παρέχεται μόνο στους χρήστες τύπου εργοδότη. Στο κεντρικό μενού τους *Ο Λογαριασμός μου* υπάρχει η επιλογή **Νέα Αγγελία**, μέσω της οποίας πραγματοποιείται η εισαγωγή της αγγελίας στην ιστοσελίδα μας. Στο ίδιο μενού βρίσκεται και η επιλογή **Οι Αγγελίες μου**, όπου πατώντας τη, εμφανίζονται αυτόματα όλες οι ενεργές αγγελίες του συγκεκριμένου εργοδότη. Σε κάθε αγγελία από αυτές, παρουσιάζεται η επιλογή επεξεργασία, μέσω της οποίας, ο εργοδότης έχει την δυνατότητα ενημέρωσης όλων των πληροφοριών της. Ας δούμε αναλυτικά τον κώδικα υλοποίησης, για την **ενημέρωση** μιας νέας αγγελίας και πως αυτός διαφοροποιείται σε σχέση με την καταχώρηση της.

Παρακάτω φαίνεται η φόρμα που πρέπει να συμπληρώσει ο χρήστης :

```
<form id="advert-form" action= "" name="advert-form" class="myform">

  <p>
  <label for="ad_title">*Τίτλος
  <span class="small">Τίτλος αγγελίας</span>
  </label>
  <input type="text" id="ad_title" name="title" />
  </p>

  <p>
  <label for="ad_category">* Κατηγορία
  <span class="small">Επιλέξτε κατηγορία</span>
  </label>
  <select id = "ad_category" name="category">
  <option value="" >Επιλέξτε Κατηγορία</option>
  </select>
  </p>
```

```
<p>

<label for="ad_edu">Εκπαίδευση
<span class="small">Ελάχιστη εκπαίδευση</span>
</label>

<select id="ad_edu" name="edu" >
<option value="" > Επιλέξτε Εκπαίδευση</option>
<option value="Τριτοβάθμια " > Τριτοβάθμια </option>
<option value="Ανώτερη " > Ανώτερη </option>
<option value="Ανώτατη" > Ανώτατη</option>
<option value="Μεταπτυχιακή" > Μεταπτυχιακή</option>
</select>

</p>

<p>

<label for="ad_xp">Προϋπηρεσία
<span class="small">Ελάχιστη προϋπηρεσία</span>
</label>

<select id="ad_xp" name="xp" >
<option value="" > Επιλέξτε Προϋπηρεσία</option>
<option value="1 εως 2 χρόνια " > 1 εως 2 χρόνια </option>
<option value="3 εως 5 χρόνια" > 3 εως 5 χρόνια </option>
<option value="6 χρόνια και πάνω" > 6 χρόνια και πάνω</option>
</select>

</p>

<p>

<label for="ad_chief">Θέση Προϊσταμένου
<span class="small">Επιλέξτε ναι/οχι</span>
</label>

<select id="ad_chief" name="chief" >
<option value="" > Προϊστάμενος?</option>
<option value="Ναι" > Ναι </option>
<option value="Οχι" > Οχι </option>
</select>

</p>

<p>

<label for="ad_salary">Μισθός
<span class="small">Προτεινόμενος μισθός</span>
</label>

<input type="text" id="ad_salary" name="salary" />

</p>
```

```

<p>
<label for="ad_description">* Περιγραφή Αγγελίας
<span class="small">Αντιγράψτε εδώ την αγγελία σας </span>
</label>

<?php

$id_advert = $_GET['id'];

    $sql = "SELECT * FROM jos_content WHERE id=$id_advert ";
    $results = mysql_query($sql) ;

        while ($results_row = mysql_fetch_assoc($results))
        {
            $intxt = $results_row['introtxt'];
        }

$editor =& JFactory::getEditor( );
echo $editor->display('content',$intxt, '550', '400', '60', '20',
false);

?>

</p>

```

Στο τέλος βλέπουμε πως για την περιγραφή της αγγελίας, χρησιμοποιήσαμε τον επεξεργαστή κειμένου(**editor**) του Joomla. Έτσι ο εργοδότης έχει την απόλυτη ελευθερία στη δημιουργία της περιγραφής του και κυρίως ότι ακριβώς θα βλέπει στον editor θα είναι αυτό ακριβώς που θα εμφανιστεί και στην μετέπειτα παρουσίαση της αγγελίας. Επίσης παρατηρούμε και το ερώτημα στο πίνακα jos\_content της βάσης μας, με το οποίο εμφανίζουμε τη περιγραφή που είχε δοθεί από τον εργοδότη κατά την καταχώρηση. Ας δούμε πώς εμφανίζουμε τα δεδομένα μας και στα υπόλοιπα πεδία :

```

if (isset ($_GET['id']))
{
    $id_advert = $_GET['id'];

        $sql = "SELECT * FROM jos_content WHERE id=$id_advert ";
    $results_number = ($results = mysql_query($sql)) ?
mysql_num_rows($results) : 0 ;

    if ($results_number == 0)
    {
        echo "Δεν υπάρχει αυτή η αγγελία";
    }
    else
    {
        while ($results_row = mysql_fetch_assoc($results))
        {
            ?>

```



```

<?php

if( isset($_POST['id']) && $_POST['id'] != null )

{

$title = $_POST['title'];

$category = $_POST['category'];

$id = $_POST['id'];

$usrid = $_POST['usrid'];

$xp = $_POST['xp'];

$salary = $_POST['salary'];

$manager = $_POST['chief'];

$edu = $_POST['edu'];

$con = $_POST['cont'];

$sql="UPDATE jos_content SET title = '$title' , catid = '$category' ,
experience = '$xp' , salary = '$salary' , studies = '$edu' , manager
= '$manager' , introtext = '$con', modified_by = '$usrid' , modified
= now() WHERE id = '$id' ";

$results = mysql_query($sql);

        echo "success";

}

else
{

        echo "error";

}

?>

```

Στην περίπτωση της **καταχώρησης** μιας αγγελίας από τον εργοδότη, οι αλλαγές σε σχέση με την ενημέρωση είναι ελάχιστες. Καταρχήν αφαιρείται ο κώδικας που εμφάνιζε τις τιμές των πεδίων, τις οποίες καλείται τώρα να συμπληρώσει ο εργοδότης, για πρώτη φορά. Επίσης αλλάζει το ερώτημα στη βάση δεδομένων μας, κατά την καταχώρηση πλέον των πεδίων (και όχι ενημέρωση), όπως φαίνεται παρακάτω :

```

<?php
if ( $user->id != 0 && $user->employer == 1)
{
$title = $_POST['title'];
$category = $_POST['category'];
$id = $_POST['usrid'];
$xp = $_POST['xp'];
$salary = $_POST['salary'];
$manager = $_POST['chief'];
$edu = $_POST['edu'];
$con = $_POST['cont'];

$sql = "INSERT INTO
jos_content(title,catid,salary,experience,studies,manager,introtext,c
reated_by,sectionid,created,state) VALUES
('$title','$category','$salary','$xp','$edu','$manager','$con','$id',
'1', now(), '1')";

$result=mysql_query($sql);

if (!$result) {
$message = 'Invalid query: ' . mysql_error() . "n";
$message .= 'Whole query: ' . $query;
die($message);
}

echo "success";

}

else
{

echo "error!!";

}

?>

```

Στην αρχή του κώδικα ελέγχουμε αν έχει πραγματοποιήσει είσοδο κάποιος χρήστης, ο οποίος θα πρέπει να είναι τύπου εργοδότη. Τέλος, εισάγουμε τα στοιχεία της αγγελίας στο πίνακα jos\_content της βάσης δεδομένων μας, κάνοντας και τους απαραίτητους ελέγχους.

## 6.9 Παρουσίαση Αγγελίας

Κάθε ενεργή αγγελία που έχει δημιουργηθεί από έναν εργοδότη, είναι διαθέσιμη για αναζήτηση από όλους του χρήστες. Όταν ο χρήστης επιλέξει από την αναζήτηση την αγγελία που τον ενδιαφέρει, αυτόματα θα μεταφερθεί στη σελίδα παρουσίασης, της συγκεκριμένης αγγελίας. Αυτό είναι και το μέρος της ιστοσελίδας μας που θα περιγράψουμε παρακάτω.

Σκοπός της συγκεκριμένης σελίδας είναι να παρουσιάσει, όσο περισσότερες πληροφορίες γίνεται για την αγγελία και τον εργοδότη της, με το πιο απλό και κατανοητό τρόπο για το χρήστη. Όταν ο χρήστης διαβάσει όλο το περιεχόμενο της σελίδας, θα πρέπει να έχει κατανοήσει αν αυτή η αγγελία τον ενδιαφέρει πραγματικά για να στείλει το βιογραφικό του ή όχι. Σε περίπτωση αναζήτησης παραπάνω πληροφοριών για μια συγκεκριμένη αγγελία, του δίνεται η δυνατότητα να γράψει κάποιο σχόλιο-διευκρίνηση, στο οποίο θα πρέπει να απαντήσει ο εργοδότης της αντίστοιχης αγγελίας. Τα σχόλια είναι κάτι που θα αναλύσουμε στην επόμενη ενότητα μας.

Για τη δημιουργία της σελίδας παρουσίασης, χρησιμοποιήσαμε την σελίδα όπου το Joomla διαθέτει για τη παρουσίαση κάθε άρθρου. Έτσι το πρώτο πράγμα που έπρεπε να βρούμε, είναι το που βρίσκεται ο κώδικας, που υλοποιεί τη παρουσίαση άρθρων. Το συγκεκριμένο Php αρχείο βρίσκεται στην εξής τοποθεσία :

```
...\\www\\workhunter\\components\\com_content\\views\\article\\tmpl\\default.php
```

Με τη βοήθεια αυτού του αρχείου, το Joomla υλοποιεί τη παρουσίαση οποιουδήποτε άρθρου δημιουργείτε από αυτό. Εμείς το μετατρέψαμε σχεδόν εξολοκλήρου, όπως θα δείξουμε και στη συνέχεια, για να εξυπηρετεί το σκοπό παρουσίασης όλων των αγγελιών στους επισκέπτες μας.

Η αγγελία, εμπεριέχει όχι μόνο τα στοιχεία αυτής, αλλά και τα στοιχεία του εργοδότη που τοποθέτησε την αγγελία σε εμάς. Όλα τα στοιχεία αυτά συμπεριλήφθησαν σε ένα Html πίνακα (table), του οποίου το αριστερό μέρος, αποτελείται από τα **στοιχεία της αγγελίας** ενώ το δεξί από **τις πληροφορίες του εργοδότη**.

Ας ξεκινήσουμε με το πώς υλοποιήσαμε τη παρουσίαση **των στοιχείων της αγγελίας**. Δημιουργήσαμε δύο πίνακες εκ των οποίων ο πρώτος εμπεριέχει το τίτλο της αγγελίας, την ημερομηνία δημιουργίας της, την κατηγορία της και διάφορα άλλα στοιχεία, όπως αριθμό εμφανίσεων και σχολίων αυτής. Στη συνέχεια εμφανίζονται τα εισαγωγικά στοιχεία της κάθε αγγελίας Προϋπηρεσία, Σπουδές, Θέση Προϊσταμένου και Μισθός. Τέλος στον δεύτερο πίνακα, εμφανίζουμε το κεντρικό κείμενο περιγραφής της κάθε αγγελίας, καθώς επίσης και την ημερομηνία ενημέρωσης της.

Ας δούμε αναλυτικά, το κώδικα υλοποίηση του συγκεκριμένου μέρους της ιστοσελίδας μας :



```

<?php

$adid= $this->article->id;
$catid = $this->article->catid;

        $comresult = mysql_query("SELECT * FROM jos_comments
WHERE id_advert=$adid");
        $comment_rows = mysql_num_rows($comresult);

        $catresult = mysql_query("SELECT title FROM
jos_categories WHERE id=$catid")

while ($catrow=mysql_fetch_assoc($catresult ))
    {
        $cat = $catrow[ 'title'];
    }

?>

<table class="contentpaneopen">
<tr valign="top">

<td>

<table class="contentpaneopen">
<tr>

    <?php if ($this->params->get('show_title')) : ?>
<td valign="middle" class="contentheading<?php echo $this-
>escape($this->params->get('pageclass_sfx')); ?>" width="100%">

        <?php echo $this->escape($this->article->title); ?>

        <p class="createdate">
        <?php echo JHTML::_('date', $this->article->created,
JText::_('DATE_FORMAT_LC2')); ?>
        <?php echo ' | '; ?>
        <?php echo 'Σχόλια' ?>
        <b><?php echo $comment_rows ?></b>
        <?php echo ' | '; ?>
        <?php echo 'Εμφανίσεις'; ?>
        <b><?php echo $this->article->hits; ?></b>
        <?php echo ' | '; ?>
        <?php echo 'Κατηγορία'; ?>
        <b><?php echo $cat ?></b>
        </p>
    </td>

    <?php endif; ?>

<td valign="top" class="contentheading<?php echo $this->escape($this-
>params->get('pageclass_sfx')); ?>" width="100%">

<div style="position:relative; right:723px;">

</div>

</td>
</tr>
</table>

```

Στην αρχή του παραπάνω κώδικα, εκτελούμε δυο **SQL query** στη βάση δεδομένων μας, για να ανακτήσουμε τον αριθμό των **σχολίων** που έχει η κάθε αγγελία (αν έχει), καθώς και τη **κατηγορία** στην οποία αυτή ανήκει.

Στη συνέχεια χρησιμοποιώντας το API του Joomla, τυπώνουμε το τίτλο της αγγελίας μας, στη κορυφή της σελίδας με την εντολή :

```
<?php echo $this->escape($this->article->title); ?>
```

στη συνέχεια εμφανίζουμε την ημερομηνία δημιουργίας, τον αριθμό των σχολίων, την κατηγορία και τον αριθμό εμφανίσεων της αγγελίας, χωρισμένα με κάθετες γραμμές. Έπειτα και λίγο πριν την εμφάνιση του κύριου κειμένου, παρουσιάζουμε τα περιληπτικά στοιχεία που έχει δώσει ο εργοδότης για τη συγκεκριμένη αγγελία.

Όπως φαίνεται στο τέλος του παραπάνω κώδικα, έχουμε τοποθετήσει στα αριστερά του πίνακα, μια φωτογραφία με το λογότυπο του εργοδότη, έτσι ώστε να γίνει η ιστοσελίδα μας πιο ευπαρουσίαστη και άμεσα κατανοητό στο χρήστη, σε ποια εταιρία αναφέρεται η αγγελία.

Ο δεύτερος πίνακας φαίνεται παρακάτω :

```
<table class="contentpaneopen"<?php echo $this->escape($this->params->get('pageclass_sfx')); ?>">
<br>
<label>Περιγραφή Αγγελίας</label>
<br>
<br>
<p>
<?php echo $this->article->text; ?></p>
</td>
</tr>

<?php if ( intval($this->article->modified) !=0 && $this->params->get('show_modify_date')) : ?>
<tr>
<td class="modifydate">
<?php echo JText::sprintf('LAST_UPDATED2', JHTML::_('date',
$this->article->modified, JText::_('DATE_FORMAT_LC2'))); ?>
</td>
</tr>
<?php endif; ?>
</table>
```

Στη συγκεκριμένη περίπτωση όπως φαίνεται και στον κώδικα χρησιμοποιήσαμε τις κλάσεις του Joomla για την εμφάνιση του κεντρικού κειμένου, αλλά και για την ημερομηνία ανανέωσης της αγγελίας.

Τα **στοιχεία του εργοδότη** βρίσκονται και αυτά με τη σειρά τους, στο εσωτερικό ενός πίνακα, ο οποίος είναι χωρισμένος από τους προαναφερθέντες πίνακες με μια μεγάλη κάθετη γραμμή. Στην κορυφή του, φαίνεται η τοποθεσία του εργοδότη στο χάρτη, με τη βοήθεια του Google Maps.

Εν συνεχεία παρουσιάζουμε την επωνυμία της εταιρίας, το ηλεκτρονικό ταχυδρομείο, το τηλέφωνο, την πόλη και την διεύθυνση της (με οδό και αριθμό).

Στο τέλος, έχουμε τοποθετήσει το κουμπί υποβολής βιογραφικού, το οποίο ο εργαζόμενος θα πρέπει να πατάει, στις αγγελίες που εκείνος επιθυμεί.

Πριν την παρουσίαση οποιοδήποτε κώδικα, πρέπει να πούμε ότι κάναμε χρήση ενός **Sql Query**, με τη βοήθεια του οποίου παίρνουμε για κάθε αγγελία όλα τα στοιχεία του εργοδότη της. Αυτό το αναφέρουμε τώρα, γιατί χρησιμοποιείται σε όλα τα σημεία της υλοποίησης που θα αναφέρουμε παρακάτω και είναι το εξής

```
$id = $this->article->created_by ;
$sql = "SELECT * FROM jos_users WHERE id=$id";

if ($result=mysql_query($sql))
{
    $row=mysql_fetch_array($result)
}
}
```

Ας δούμε τώρα βήμα προς βήμα, την υλοποίηση του πίνακα.

```
<td class="greyline">&nbsp;</td>
<td>
<table class="contentpaneopen">
<tr>
<td width="160">
<label>Που Βρισκόμαστε...</label>
<br>
<div id = "map_canvas" style="margin-top:10px; width:150;
height:180" />
<td>
</tr>
<tr>
<td>
```

```

<label>Τα στοιχεία μας</label>
<br>
<br>

<p><label for='uploaded_file'><b>Επωνυμία </b>
<br> <?php echo $row['name']; ?></span><br></label></p>

<p><label for='uploaded_file'><b>Email </b>
<br> <?php echo $row['email']; ?><br></label></p>

<p><label for='uploaded_file'><b>Τηλέφωνο</b>
<br> <?php echo $row['phone']; ?><br></label></p>

<p><label for='uploaded_file'><b>Πόλη</b>
<br> <?php echo $row['town']; ?><br></label></p>

<p><label for='uploaded_file'><b>Διεύθυνση</b>
<br> <?php echo $row['street'];?>
<?php echo $row['streetnumber'];?><br></label></p>

<br>
<br>
<br>
<form id = 'mail'
action="index.php?option=com_content&view=article&id=<?php echo
$this->article->id; ?>" method="post">

<input type="hidden" name="sendmail" value="send" />

</form>

<input type="submit" name="biobutton" id="biobutton"
class="button green" value="Βιογραφικό" onclick="disableMail( ) ">

<br>
<br>
<br>

</td>
</tr>
</table>
</td>

```

Στην αρχή του κώδικα φαίνεται η δημιουργία της οριζόντιας γραμμής, που χωρίζει ολόκληρο το πίνακα σε δυο διακριτά μέρη. Η κορυφή του νέου μας πίνακα, έχει τη περιγραφή “Που Βρισκόμαστε” και ακριβώς από κάτω, βρίσκεται ένα **html div element με id = map\_canvas**. Στο συγκεκριμένο div θα το τοποθετηθεί ο χάρτης που θα εμφανίζει την ακριβή τοποθεσία του εργοδότη, κάνοντας χρήση του Google Maps API που παρουσιάζεται παρακάτω.

```

<script type="text/javascript">

var geocoder;
var map;
function initialize() {

geocoder = new google.maps.Geocoder();

var latlng = new google.maps.LatLng(-34.397, 150.644);

var myOptions = { zoom: 16, center: latlng, mapTypeId:
google.maps.MapTypeId.ROADMAP }

map = new google.maps.Map(document.getElementById("map_canvas"),
myOptions);

codeAddress();}

function codeAddress() {

var address = "<?php echo $row['street'];?> <?php echo
$row['streetnumber']; ?> <?php echo $row['town']; ?> ";

geocoder.geocode( { 'address': address}, function(results, status) {

if (status == google.maps.GeocoderStatus.OK) {

map.setCenter(results[0].geometry.location);

var marker = new google.maps.Marker({

map: map,

position: results[0].geometry.location });

} else {

alert("Geocode was not successful for the following reason: " +
status); } }); }

</script>

```

Ο παραπάνω κώδικας παίρνει ως παράμετρο από εμάς, τη διεύθυνση (οδό - αριθμό) και τη πόλη, την οποία μας έχει δώσει ο εργοδότης κατά την εγγραφή του. Με τον τρόπο αυτό έχουμε στη παρουσίαση της αγγελίας μας, τον χάρτη με την ακριβή τοποθεσία του εργοδότη που προσφέρει την εργασία.

Επιστρέφοντας ξανά στον αρχικό μας κώδικα, παρατηρούμε πως στη συνέχεια υπάρχει ένα label με την ένδειξη “Τα Στοιχεία μας” το οποίο ουσιαστικά έχει το ρόλο επικεφαλίδας για τα στοιχεία που ακολουθούν. Εδώ δίνουμε όλα τα στοιχεία της εταιρίας που έχουμε στη διάθεση μας, από τη χρήση του Sql Query που αναφέραμε στην αρχή.

Τα στοιχεία λοιπόν αυτά είναι η **επωνυμία** της εταιρίας, το **ηλεκτρονικό ταχυδρομείο**, το **τηλέφωνο**, η **πόλη** και η **διεύθυνση** (με οδό και αριθμό).

Στο τέλος, βλέπουμε μια φόρμα που εμπεριέχει ένα κρυμμένο πεδίο και το κουμπί για την αποστολή βιογραφικού του εργαζομένου, που προαναφέραμε. Σε αυτό το κομμάτι, ουσιαστικά υλοποιείται η αποστολή του βιογραφικού μέσω ηλεκτρονικού ταχυδρομείου την οποία και θα αναλύσουμε στη συνέχεια ενδελεχώς.

Όταν ο χρήστης πατήσει το κουμπί της αποστολής βιογραφικού, τότε ενεργοποιείται το onclick event και εκτελείται το παρακάτω script.

```
<script>

function disableMail( )

{
    jQuery('#biobutton').attr("disabled", "true");
    jQuery('#mail').submit();
}

</script>
```

Το πρώτο πράγμα που γίνεται είναι η απενεργοποίηση του κουμπιού, έτσι ώστε να διασφαλίσουμε ότι ο χρήστης δε θα μπορεί να το ξαναπατήσει και ότι ο παραλήπτης θα λάβει μόνο ένα ηλεκτρονικό ταχυδρομείο. Αμέσως μετά υποβάλουμε(submit) χειροκίνητα τα στοιχεία της φόρμας με id=mail. Με αυτό το τρόπο ενεργοποιούμε το παρακάτω κομμάτι κώδικα, που εκτελείτε μόνο εάν έχει γίνει η υποβολή των στοιχείων της φόρμας.

```
<?php

$bio = $this->user->bio;
$from = $this->user->email;

$fromname = $this->user->name;
$fromname .= " ";
$fromname .= $this->user->firstname;

$titlemail = $this->article->title;
$path = "components/com_user/bio/$bio";

$bio = $this->user->firstname;
$body = " Σας αποστέλω το βιογραφικό μου για την αγγελία με τε τίτλο
'$titlemail' " ;

$email = $row['email'];
```

```

if(isset($_POST['sendmail']))
{
    if ( $this->user->id != 0)
    {
        $mail =& JFactory::getMailer();
        $mail->addRecipient($email) ;
        $mail->setSender(array($from, $fromname));
        $mail->setSubject( 'Βιογραφικό' );
        $mail->setBody($body);
        $mail->addAttachment('/components/com_user/logos/$logo');

        if ($mail->Send())
        {
            $idadv = $this->article->id;
            $idusr = $this->user->id;

            mysql_query("INSERT INTO jos_userbio (id_user, id_advert) VALUES
($idusr,$idadv)");

            ?>

            <script>

                jQuery(document).ready(function()
                {
                    jQuery('#biobutton').attr("disabled", "true");

                });

            </script>

            <?php

                JFactory::raiseNotice( 100, 'Η αποστολή του βιογραφικού σας
ήταν επιτυχής!!!' );

            }

            else
            {
                JFactory::raiseWarning( 100, 'Η αποστολή του βιογραφικού σας
απέτυχε' );
            }

            }

            else
            {
                JFactory::raiseWarning( 100, 'Η αποστολή του βιογραφικού σας
απέτυχε' );
            }

            }

            ?>

```

Για την υλοποίηση της αποστολής του ηλεκτρονικού ταχυδρομείου, χρησιμοποιήσαμε τη μέθοδο του Joomla **getMailer()**. Δημιουργούμε όλα τα στοιχεία που είναι απαραίτητα για την αποστολή του ηλεκτρονικού ταχυδρομείου και τα οποία είναι :

- ο παραλήπτης ( εργοδότης )
- ο αποστολέας ( εργαζόμενος )
- το θέμα ( βιογραφικό )
- το σώμα ( περιγραφικό κείμενο που καταλήγει με το τίτλο της αγγελίας )
- το συνημμένο ( το βιογραφικό του εργαζομένου )

Τέλος είναι πολύ βασικό, να ορίσουμε πότε το κουμπί θα είναι ενεργό και πότε όχι. Εμείς επιτρέπουμε μόνο σε εγγεγραμμένους χρήστες, να μπορούν να στείλουν το βιογραφικό τους και οι οποίοι μάλιστα θα πρέπει να είναι μόνο εργαζόμενοι και όχι εργοδότες. Σε όποιες αγγελίες ένας εργαζόμενος έχει στείλει το βιογραφικό του, θα πρέπει για το συγκεκριμένο χρήστη, το κουμπί των αγγελιών αυτών να απενεργοποιείται. Αυτούς τους ελέγχους τους πραγματοποιήσαμε με το παρακάτω κώδικα :

```
//disable button

$userid = $this->user->id;
$advertid = $this->article->id;
$sqlcheck="SELECT id_user, id_advert FROM jos_userbio WHERE
id_user = $userid and id_advert = $advertid ";
$check=mysql_query($sqlcheck);

if (mysql_num_rows($check) == false)

{ }

else
{

?>

<script>
    jQuery(document).ready(function() {
        jQuery('#biobutton').attr("disabled", "true");

    });
</script>

<?php
}
```



```

if ( $this->user->id == 0 || $this->user->employer ==1)
    {
        ?>
<script>
    jQuery(document).ready(function() {
        jQuery('#biobutton').attr("disabled", "true");

        });
</script>
<?php
}

```

Σε όλες τις περιπτώσεις ελέγχου που κάνουμε

- ✓ Αν ο χρήστης έχει κάνει είσοδο
- ✓ Αν ο χρήστη είναι εργαζόμενος
- ✓ Αν ο εργαζόμενος έχει δώσει ήδη το βιογραφικό του

Χρησιμοποιούμε τη JavaScript βιβλιοθήκη jQuery για την απενεργοποίηση της αποστολής του βιογραφικού.

Τέλος, έχουμε τοποθετήσει στην αγγελία μας τη δυνατότητα σε κάθε επισκέπτη, να κάνει χρήση των γνωστών κουμπιών **Like** και **Send** του Facebook. Έτσι ο επισκέπτης έχει τη δυνατότητα να δει πόσα 'like' έχουν γίνει στη συγκεκριμένη αγγελία και να προσθέσει και αυτός άλλο ένα, εάν το επιθυμεί. Μπορεί επίσης, με τη βοήθεια του κουμπιού Send να αποστείλει σε όποιο 'φίλο' του στο Facebook επιθυμεί, την αγγελία που τον ενδιαφέρει.

Αυτός είναι ο πιο εύκολος και χωρίς κανένα κόστος τρόπος διαφήμισης της ιστοσελίδας μας, σε ολόκληρο τον κόσμο. Η υλοποίηση των κουμπιών έγινε με τη χρήση του API που μας παρέχει το Facebook, όπως φαίνεται παρακάτω.

```

<div id="fb-root"></div>

<script src="http://connect.facebook.net/en_US/all.js#xfbml=1">
</script>

<fb:like
href="http://www.workhunter.gr/index.php?option=com_content&view=article&id=<?php echo $this->article->id; ?>" send="true" width="450"
show_faces="false" font="verdana">
</fb:like>

```

## 6.10 Σχόλια

Κάτω ακριβώς από την εμφάνιση των αγγελιών, έχουμε τοποθετήσει την λειτουργία των σχολίων. Κάθε εγγεγραμμένος χρήστης αλλά και επισκέπτης της σελίδας μας, έχει την δυνατότητα να σχολιάσει την αγγελία που επιθυμεί.

Η φόρμα καταχώρησης των σχολίων, αποτελείται από δύο πεδία, το όνομα και το κείμενο σχολίου, τα οποία και πρέπει να συμπληρώσει ο χρήστης για να υποβάλει επιτυχώς το σχόλιο του.

Τα σχόλια τοποθετούνται κατακόρυφα, το ένα κάτω από το άλλο, με χρονολογική σειρά (το τελευταίο είναι το νεότερο). Ας δούμε τον κώδικα υλοποίησης της φόρμας υποβολής σχολίου :

```
<div id="comments">
  <h2 class = "comment">Σχόλια</h2>
  <ul id="commentList">
    <?php

      $thisid = $this->article->id;

      $sqlcom = "SELECT * FROM jos_comments WHERE id_advert = $thisid ";

      $resultscom = mysql_query($sqlcom);

      while ($rowcom = mysql_fetch_assoc($resultscom))
      {
        echo "<li>" , $rowcom['name'],           $rowcom['date'],
          $rowcom['comment'], "</li>";

      }

    ?>
  </ul>

  <form id="commentForm" method="post" action="" >

    <div>
      <label for="comname">Το Όνομα σας</label>
      <input id="comname" type="text" name="comname" /><br>
      <label for="comment">Το Σχόλιο σας</label>
      <textarea id="comment" name="comment" cols="40"
rows="5"></textarea>
      <input type="hidden" value="
      <?php
        $greekMonths =
array('Ιανουαρίου', 'Φεβρουαρίου', 'Μαρτίου', 'Απριλίου', 'Μαΐου', 'Ιουνί
ου', 'Ιουλίου', 'Αυγούστου', 'Σεπτεμβρίου', 'Οκτωβρίου', 'Νοεμβρίου', 'Δεκε
μβρίου');

        $greekDate = "στις ";
        $greekDate .= date('d') . ' ' .
        $greekMonths[intval(date('m'))-1] . ' ' . date('Y') . ' ' .
date("H:i:s");
        echo $greekDate;    ?>" id="comdate" name="comdate" />
      <input type="submit" value="Σχολίασε" /><div id="comerror" >
    </div>
  </div></form>
</div>
```

Παρατηρούμε πως πριν την φόρμα υποβολής σχολίου, υπάρχει ένα ερώτημα στον πίνακα **jos\_comments** με τη βοήθεια του οποίου εμφανίζουμε όλα τα προηγούμενα

σχόλια μιας αγγελίας, εάν αυτά φυσικά υπάρχουν. Κατά την υποβολή του σχολίου εκτελείτε ο παρακάτω κώδικας, με τον οποίο λαμβάνουμε τα δεδομένα από τη φόρμα που συμπλήρωσε ο χρήστης και στη συνέχεια τα παρουσιάζουμε. Ο χρήστη είναι υποχρεωμένος να συμπληρώσει και τα δυο πεδία (όνομα, σχόλιο) για να καταχωρηθεί επιτυχώς το σχόλιο του.

```
<script>
```

```
jQuery(document).ready(function() {  
  
    jQuery('#commentForm').submit(function() {  
  
        jQuery.post(  
            'newcomment.php',  
  
            {  
                comname:jQuery("#comname").val(),  
                comment:jQuery("#comment").val(),  
                date :jQuery("#comdate").val(),  
                articleid: '<?php echo $this->article->id ?>'  
            },  
  
            function(response)  
            {  
  
                if (response == "error")  
                {  
  
                    jQuery('#comerror').html("Συμπληρώστε το <strong>Όνομα και το Σχόλιο σας</strong>");  
                }  
  
                else  
                {  
                    jQuery('#commentList').append(response);  
                    jQuery('#comerror').html("");    jQuery('#comname').val('');  
                    jQuery('#comment').val(''); }  
                }  
  
            )  
  
            return false;  
  
        });  
  
    });  
  
</script>
```

Η αποθήκευση τελικά του σχολίου στη βάση δεδομένων μας, γίνεται στο αρχείο **newcomment.php** το οποίο και παρουσιάζουμε στη συνέχεια :

```
<?php
if ( $_POST['comment'] && $_POST['comname'] )
{
    if ( $_POST['comname'] ) {
        $comname = '<h3>'.strip_tags($_POST['comname']).'</h3>';
    }
    $comment = '<p>'.strip_tags($_POST['comment']).'</p>';

    $date = $_POST['date'];
    echo "<li>$comname$date$comment</li>";

    $id = $_POST['articleid'];

    $sql = "INSERT INTO jos_comments (name,comment,id_advert,date)
VALUES ('$comname','$comment','$id','$date')";
    $result=mysql_query($sql);

    if (!$result) {
        $message = 'Invalid query: ' . mysql_error() . "n";
        $message .= 'Whole query: ' . $query;
        die($message);
    }
}
else
{ echo "error"; }
?>
```

Στην αρχή του κώδικα ελέγχουμε αν τα πεδία του σχολίου και του ονόματος έχουν συμπληρωθεί. Αν είναι κενά, τότε επιστρέφουμε *error* και αυτόματα εμφανίζεται το αντίστοιχο μήνυμα λάθους ενώ αν τα πεδία έχουν συμπληρωθεί, τότε πρώτα αποθηκεύουμε το όνομα, την ημερομηνία και το σχόλιο που έκανε ο χρήστης και εν συνεχεία τα εμφανίζουμε.

## 6.11 Είσοδος στην Ιστοσελίδα ( Log In )

Για την είσοδο ενός χρήστη στην ιστοσελίδα μας χρησιμοποιήσαμε εξολοκλήρου τη λειτουργία(module) εισόδου που μας παρέχει το Joomla. Με την εγκατάσταση όμως ενός plugin της αλλάξαμε τοποθεσία και εμφάνιση. Πλέον βρίσκεται επάνω δεξιά, διπλά από το κεντρικό μενού και παρουσιάζεται ως ένα κουμπί με ονομασία **Είσοδος**. Πατώντας το, εμφανίζονται τα πεδία ονόματος χρήστη και κωδικού που πρέπει να συμπληρώσει ο χρήστης, για να εισέλθει τελικά στην σελίδα.

Αν ο χρήστης πραγματοποιήσει επιτυχημένη είσοδο, το κουμπί μετονομάζεται σε Έξοδο και με το οποίο ο χρήστης έχει την δυνατότητα της αποσύνδεσης. Τέλος, στη περίπτωση που ο χρήστης έχει ξεχάσει κάποιο από τα στοιχεία εισόδου του, προσφέρεται η επιλογή αποστολής τους στο ηλεκτρονικό ταχυδρομείο του.

## Κεφάλαιο 7 Ιστοσελίδα στο Διαδίκτυο

Αποφασίσαμε πως για την ολοκλήρωση της πτυχιακής, θα έπρεπε να ανεβάσουμε την ιστοσελίδα μας στο διαδίκτυο. Έτσι μόνο, θα μπορούσε να γίνει μια ζωντανή και πραγματική αναπαράσταση της σελίδας που δημιουργήσαμε. Το πρώτο βήμα ξεκινώντας, ήταν η αναζήτηση του ονόματος που θα δίνουμε στην ιστοσελίδα. Καταλήξαμε στο **workhunter.gr**, πιστεύοντας πως αυτή η ονομασία θα αντιπροσωπεύει το περιεχόμενό και τον σκοπό της. Αγοράσαμε λοιπόν το domain name **workhunter.gr** για δυο χρόνια και από τον ίδιο πάροχο, αγοράσαμε το χώρο όπου θα γινόταν η φιλοξενία της ιστοσελίδας. Κάναμε χειροκίνητη εγκατάσταση του Joomla1.5 στο χώρο φιλοξενίας της σελίδας τοποθετώντας παράλληλα, όλα μας τα αρχεία και την νέα βάση δεδομένων. Με αυτό τον τρόπο η πτυχιακή μας εργασία είναι διαθέσιμη σε όλους, πληκτρολογώντας απλά σε ένα περιηγητή, την ηλεκτρονική διεύθυνση [www.workhunter.gr](http://www.workhunter.gr).

## Συμπεράσματα – Επίλογος

Η πραγματοποίηση της πτυχιακής μας εργασίας, αποτελεί σίγουρα την πιο ολοκληρωμένη προσπάθεια που έχουμε υλοποιήσει από την εισαγωγή μας στη σχολή. Έπρεπε ουσιαστικά να χρησιμοποιήσουμε τις γνώσεις που αποκτήσαμε από τα χρόνια της φοίτησης μας, σε συνδυασμό με τις προσωπικές μας εμπειρίες και εφόδια έτσι ώστε να υλοποιήσουμε την ιστοσελίδα. Σίγουρα ήταν μια δύσκολη αλλά συνάμα απόλυτα ενδιαφέρουσα εμπειρία και εφόδιο για την επαγγελματική μας πορεία.

Η ενασχόληση μας με το Joomla, πραγματικά μας έδωσε να καταλάβουμε γιατί θεωρείται ένα από τα καλύτερα συστήματα διαχείρισης περιεχομένου. Εύχρηστο, κατανοητό και με απίστευτες δυνατότητες κάνει τη δημιουργία αλλά κυρίως τη διαχείριση μιας ιστοσελίδας, πιο απλή από όσο θα φανταζόταν κάποιος. Η κοινότητα επίσης του Joomla έχει γιγαντωθεί, βρίσκοντας ατελείωτα πρόσθετα και συμβουλές για οποιοδήποτε πρόβλημα και αν μας παρουσιαστεί.

# Βιβλιογραφία

## Βιβλία

### **Joomla! 1.5 Template Design**

Συγγραφέας Tessa Blakeley Silver

### **Joomla! Bible**

Συγγραφέας Ric Shreves

### **Beginning Joomla! Web Site Development**

Συγγραφέας Cory Webb

### **Learning Joomla! 1.5 Extension Development**

Συγγραφέας Joseph LeBlanc

## Ιστοσελίδες

[www.joomla.org](http://www.joomla.org)

[www.joomla.gr](http://www.joomla.gr)

[extensions.joomla.org](http://extensions.joomla.org)

[www.wampserver.com](http://www.wampserver.com)

[css-tricks.com](http://css-tricks.com)

[www.w3schools.com](http://www.w3schools.com)

[tutorialzine.com](http://tutorialzine.com)

[www.mysql.com](http://www.mysql.com)

[www.php.net](http://www.php.net)

[www.jquery.com](http://www.jquery.com)

[docs.jquery.com/Plugins/Validation](http://docs.jquery.com/Plugins/Validation)

[code.google.com/intl/el-GR/apis/maps/index.html](http://code.google.com/intl/el-GR/apis/maps/index.html)

[developers.facebook.com/docs/reference/plugins/like](http://developers.facebook.com/docs/reference/plugins/like)