

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πάτρας  
Τμήμα Εφαρμογών Πληροφορικής στην Διοίκηση και την Οικονομία, Αμαλιάδα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

με θέμα

**«Ανάπτυξη Portal για το Τ.Ε.Ι. Αμαλιάδας»**

**«Portal Development for Higher Technological Education Institute of  
Amaliada»**

**ΚΥΡΙΑΚΑΚΗΣ ΑΔΑΜΑΝΤΙΟΣ**

**ΤΑΞΙΑΡΧΙΩΤΗΣ ΙΩΑΝΝΗΣ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΧΟΧΟΛΗΣ ΔΙΟΝΥΣΙΟΣ**

**ΑΜΑΛΙΑΔΑ 2012**

## Περιεχόμενα

Σύνοψη .....	2
Summary .....	3
<b>ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ .....</b>	<b>4</b>
1.1 Η ιστορία του παγκόσμιου ιστού .....	4
1.2 Τι είναι περιεχόμενο .....	5
1.3 Τι είναι Portal .....	6
1.4 Σύστημα Διαχείρισης Περιεχομένου .....	8
1.5 Δημοφιλή Συστήματα Διαχείρισης Περιεχομένου .....	10
<b>ΚΕΦΑΛΑΙΟ 2 – ΓΝΩΣΤΙΚΟ ΥΠΟΒΑΘΡΟ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ .....</b>	<b>11</b>
2.1 Ιστοσελίδα .....	12
2.2 Ιστότοπος .....	12
2.3 Web Browser .....	12
2.4 Domain Name .....	13
2.5 Στατικές Ιστοσελίδες .....	14
2.6 Δυναμικές Ιστοσελίδες .....	14
2.7 Υλοποίηση Πρωτότυπου Συστήματος Portal σε PHP .....	15
2.8 Το Portal του ΤΕΙ Αμαλιάδας .....	17
<b>ΚΕΦΑΛΑΙΟ 3 – ΥΛΟΠΟΙΗΣΗ PORTAL .....</b>	<b>19</b>
3.1 Σχετικά με την Υλοποίηση .....	20
3.2 Τεχνολογίες που Χρησιμοποιήθηκαν .....	20
3.3 Βήματα Ανάλυσης και Υλοποίησης Portal .....	27
3.3.1 Δομή Αρχείων Συστήματος Portal .....	27
3.3.2 Δομή Βάσης Δεδομένων Συστήματος Portal .....	28
3.3.3 Δομή Πυρήνα Συστήματος Portal .....	37
3.3.4 Template και Blocks .....	41
3.3.5 Γεγονότα που λαμβάνουν χώρα κατά την είσοδο του επισκέπτη στο Portal .....	42
3.3.6 Δομή Κεντρικής Σελίδας .....	43
3.3.7 Δομή Σελίδας article.php .....	44
3.3.8 Δομή Σελίδας announcements.php .....	45
3.3.9 Δομή Σελίδας pages.php .....	46
3.3.10 Δομή Σελίδας registration.php .....	47
3.3.11 Δομή Σελίδας login.php .....	48
3.3.12 Δομή Συστήματος Δημοσκοπήσεων .....	48
<b>ΚΕΦΑΛΑΙΟ 4 – ΔΙΑΧΕΙΡΗΣΤΙΚΑ ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ .....</b>	<b>48</b>
<b>ΚΕΦΑΛΑΙΟ 5 – ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ .....</b>	<b>49</b>
<b>ΚΕΦΑΛΑΙΟ 6 – ΠΑΡΑΔΕΙΓΜΑΤΑ ΤΟΥ PORTAL ΜΕΣΑ ΑΠΟ ΣΤΙΓΜΙΟΤΥΠΑ (SCREENSHOTS) .....</b>	<b>51</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>58</b>
<b>ΠΑΡΑΡΤΗΜΑ .....</b>	<b>59</b>

## Σύνοψη

Η συγκεκριμένη πτυχιακή έχει στόχο την ανάπτυξη portal με χρήση των τεχνολογιών προγραμματισμού HTML, CSS, XML, PHP και MySQL.

Το σύστημα διαχείρισης περιεχομένου θα καλύψει τις ανάγκες λειτουργίας του διαδικτυακού τόπου του τμήματος Ε.Π.Δ.Ο. παρέχοντας λειτουργίες όπως: προβολή και διαχείριση άρθρων, υποσέλιδων, ανακοινώσεων καθώς και ένα σύστημα λογαριασμών χρηστών.

Συγκεκριμένα, το σύστημα διαχείρισης περιεχομένου, χωρίζεται σε δύο βασικά μέρη. Το πρώτο μέρος αφορά στους επισκέπτες όπου θα έχουν τη δυνατότητα να πλοηγηθούν στον δικτυακό τόπο, να δουν τα άρθρα που έχουν δημοσιευθεί, τις τελευταίες ανακοινώσεις, καθώς και να συμμετάσχουν στην αναρτημένη ψηφοφορία.

Το δεύτερο μέρος αφορά στον πίνακα διαχείρισης του συστήματος διαχείρισης περιεχομένου, όπου εκεί θα έχουν πρόσβαση οι διαχειριστές του συστήματος. Από τον πίνακα διαχείρισης θα γίνεται και η γενική διαχείριση, συντήρηση, ανανέωση του δικτυακού τόπου. Ο τρόπος ανάπτυξης θα είναι κατάλληλος ώστε να μπορούν εύκολα να προσθέτονται ή να τροποποιούνται όλα τα δεδομένα του συστήματος, χωρίς να χρειάζεται μετατροπή στον κώδικα ή πρόσβαση στη βάση δεδομένων.

## **Summary**

Target of this thesis is to develop a portal using the latest web development technologies including HTML,CSS,XML,PHP and MySQL.

This Content Management System will cover the needs of a website for use by the Highest Technological Education Institute of Amaliada City providing an advanced interface for viewing and managing articles, pages and sub pages, posting and managing announces and polls and a complete user management system.

This Content Management System is divided in two base parts: The first part includes the public view, what all visitors can see and use. All visitors are able to navigate to this site and make use of all provided services like viewing latest articles, voting in polls e.t.c.

The second part includes the control panel of the portal including the content management area (article, announcements, poll management e.t.c.) and user management system.

System is developed in a way that enables everyone to edit or add new content in it without the need of modifying the source code or to directly access the database.

# ΚΕΦΑΛΑΙΟ 1.

## ΕΙΣΑΓΩΓΗ

### 1.1 Η ΙΣΤΟΡΙΑ ΤΟΥ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

Παγκόσμιος ιστός και Διαδίκτυο (Internet) συχνά θεωρείται από πολλούς ως το ίδιο πράγμα. Αυτή η θεώρηση όμως είναι λανθασμένη καθώς ο ιστός αποτελεί μία μόνο εφαρμογή του διαδικτύου, και για να είμαστε ακριβείς, την δημοφιλέστερη. Το διαδίκτυο, έχει και υλική υπόσταση: Υπολογιστές, εκατομμύρια χιλιομέτρων καλώδια και οπτικές ίνες, δρομολογητές, κεντρικοί διακομιστές κ.α. Από την άλλη, ο Παγκόσμιος Ιστός δεν έχει υλική υπόσταση, μιας και αποτελείται από πακέτα πληροφορίας. Η τεχνολογία του Παγκόσμιου Ιστού καθιστά δυνατή την δημιουργία "υπερκειμένων", μία διασύνδεση δηλαδή πάρα πολλών μη ιεραρχημένων στοιχείων που παλαιότερα ήταν απομονωμένα. Τα στοιχεία αυτά μπορούν να πάρουν και άλλες μορφές πέραν της μορφής του γραπτού κειμένου, όπως εικόνας και ήχου. Κάνοντας αναδρομή στο παρελθόν, η τεχνολογία του ιστού δημιουργήθηκε το 1989 από τον Βρετανό Τιμ Μπέρνερς Λι, που εκείνη την εποχή εργαζόταν στον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (το γνωστό σε όλους μας CERN) στην Γενεύη της Ελβετίας.[1.1] Το όνομα που έδωσε στην εφεύρεσή του ο ίδιος ο Lee είναι World Wide Web, όρος γνωστός στους περισσότερους από το "www". Αυτό που οδήγησε τον Lee στην εφεύρεση του Παγκόσμιου ιστού ήταν το όραμά του για ένα κόσμο όπου ο καθένας θα μπορούσε να ανταλλάσσει πληροφορίες και ιδέες άμεσα προσβάσιμες από τους υπολοίπους. Το σημείο στο οποίο έδωσε ιδιαίτερο βάρος ήταν η μη ιεράρχηση των διασυνδεδεμένων στοιχείων. Οραματίστηκε κάθε στοιχείο, κάθε κόμβο του ιστού ίσο ως προς την προσβασιμότητα με τα υπόλοιπα.[1.9] Στην πραγματικότητα βέβαια κάτι τέτοιο δεν συμβαίνει, μιας και αν σκεφτεί κανείς τον βαθμό ιεράρχησης με τον οποίο λειτουργούν οι μηχανές αναζήτησης του ιστού, όπως για παράδειγμα το google, το οποίο καθιστά κόμβους του ιστού δημοφιλέστερους έναντι άλλων, γίνεται εύκολα κατανοητό ότι στην πράξη κάτι τέτοιο δεν συμβαίνει, τουλάχιστον στον βαθμό που το είχε οραματιστεί ο Lee. Σε κάθε περίπτωση κάθε νέος κόμβος έχει την δυνατότητα με χρήση των κατάλληλων τεχνολογιών και μέσων, να ανακτήσει το χαμένο έδαφος (Search Engine Optimization / SEO κτλ).

## 1.2 ΤΙ ΕΙΝΑΙ ΠΕΡΙΕΧΟΜΕΝΟ

Δεδομένα, πληροφορία, περιεχόμενο, γίνονται όλοι οι όροι που συχνά χρησιμοποιούνται ως συνώνυμα, αλλά στην πραγματικότητα ο κάθε ένας έχει μια συγκεκριμένη έννοια που το διαφοροποιεί από τους υπόλοιπους. Κατά την προσέγγιση του Boiko ([http://metatorial.com/downloads/Boiko\\_Wp\\_UnderstandingContentManagement.pdf](http://metatorial.com/downloads/Boiko_Wp_UnderstandingContentManagement.pdf)) αλλά και οι πιο συχνά αποδεκτοί ορισμοί, είναι:

- Δεδομένα: Μικρό κομμάτι πληροφορίας που από μόνο του δεν έχει καμία σημασία για όλους μας (συνάφεια, νόημα κ.ο.κ.). Μπορεί όμως να συλλεχθεί σε αρχείο ή και να αποθηκευθεί σε βάση δεδομένων. Είναι η στοιχειώδης μονάδα χειρισμού πληροφορίας.
- Πληροφορία: Οποιαδήποτε μορφή καταγεγραμμένης επικοινωνίας, δηλαδή οποιοδήποτε είδος κειμένου (άρθρα, βιβλία, κ.τ.λ.), κινούμενων εικόνων (βίντεο, κινούμενα σχέδια κ.τ.λ.), αρχείων υπολογιστή (έγγραφα του Word, παρουσιάσεις του PowerPoint και άλλα) που θα μπορούν να περιέχουν μεγάλη ή καμία απολύτως ανθρώπινη σημασία. Η πληροφορία επιτρέπει στα δεδομένα να ερμηνευτούν και να αποκαλύψουν κρυμμένα νοήματα και αναπάντεχες σχέσεις.
- Περιεχόμενο: Η πληροφορία γίνεται περιεχόμενο όταν χρησιμοποιείται για έναν ή περισσότερους σκοπούς. Η αξία του είναι το άθροισμα της αρχικής του μορφής (πληροφορία), της εφαρμογής, της χρησιμότητας, της σημασίας αλλά και της μοναδικότητάς του. Το περιεχόμενο, είναι πληροφορία συν ένα σύνολο από δεδομένα που την θέτουν σε ένα καλά ορισμένο εννοιολογικό πλαίσιο.
- Γνώση: Η πνευματική κατάσταση του ατόμου που κατέχει πληροφορίες και που γνωρίζει κάτι με αυτοπεποίθηση λόγω εμπειρίας.

Αν οι τέσσερις αυτές έννοιες ήταν μέρη του λόγου, τα δεδομένα θα ήταν μεμονωμένες λέξεις και η πληροφορία μια πρόταση που αποτελείται από αυτές τις λέξεις με συγκεκριμένη σειρά. Αν μια πληροφορία εισαχθεί σε ένα συγκεκριμένο εννοιολογικό πλαίσιο, γίνεται περισσότερο αντιληπτή η έννοια του περιεχομένου. Συνεχίζοντας τον προηγούμενο παραλληλισμό με τα μέρη του λόγου, θα μπορεί να θεωρηθεί ότι η εισαγωγή μιας πληροφορίας σε ένα πλαίσιο είναι το αντίστοιχο της χρησιμοποίησης μιας πρότασης σε ένα κείμενο: το εννοιολογικό πλαίσιο που δημιουργείται από τα συμφραζόμενα, δίνει στην πρόταση ένα συγκεκριμένο νόημα που γίνεται αντιληπτό από τον άνθρωπο που την διαβάζει. Είναι ακριβώς η στιγμή που η πρόταση γίνεται περιεχόμενο. Φυσικά, ο παραλληλισμός δεν είναι τελείως άστοχος αφού μιλώντας, μια πρόταση μπορεί να θεωρηθεί πληροφορία και τοποθετημένη σε ένα εννοιολογικό πλαίσιο (πχ μια παράγραφο με συμφραζόμενα ή

κάποια λεζάντα) μπορεί να αποτελέσει περιεχόμενο. Τέλος εάν όλα αυτά επεκταθούν λίγο και διαβαστούν από κάποιον άνθρωπο, τότε μπορούν να θεωρηθούν γνώση.

Η διαφορά μεταξύ των όρων δεδομένα, πληροφορία, περιεχόμενο και γνώση, είναι ιδιαίτερα σημαντική για την κατανόηση των προκλήσεων της διαχείριση περιεχομένου με εργαλεία της Πληροφορικής. Οι υπολογιστές αρχικά δημιουργήθηκαν για να εκτελούν πολύπλοκους και χρονοβόρους υπολογισμούς. Σε αυτό το σημείο έρχεται ο Βοϊκο [3] ο οποίος περιγράφει το υπολογιστικό μοντέλο ως ακολούθως: «Αν μπορέσεις να αναγάγεις ένα πρόβλημα σε μια σειρά απλών μηχανικών με αριθμούς και λογικές συνθήκες (true / false) τότε το πρόβλημα μπορεί να επιλυθεί από τον υπολογιστή.

Τα δεδομένα βέβαια μπορεί να μην είναι άμεσα αναγνώσιμα ή κατανοητά γιατί η μορφή τους έχει επιλεγεί ώστε να είναι αναγνώσιμα και κατανοητά από τους ίδιους τους υπολογιστές. Αυτό είναι το πρόβλημα με την διαχείριση περιεχομένου: Οι υπολογιστές είναι μηχανές που έχουν ως αποστολή να επεξεργάζονται δεδομένα, αλλά το περιεχόμενο δεν είναι απλά και σκόρπια δεδομένα. Όπως αναφέραμε προηγουμένως, το περιεχόμενο σαν έννοια, αντιπροσωπεύει την εισαγωγή πληροφοριών σε ένα συγκεκριμένο εννοιολογικό πλαίσιο. Όταν τα δεδομένα αποκτούν μια χρήσιμη μορφή, τότε αποκτούν αξία. Όπως αναφέραμε, η αξία αυτών των δεδομένων, καθορίζεται από την μορφή, την εφαρμογή, την αναγνωσιμότητα αλλά και την μοναδικότητά τους. Για παράδειγμα, δεδομένα αποθηκευμένα στην κεντρική βάση δεδομένων (Database) μιας επιχείρησης δεν μπορούν να χρησιμοποιηθούν απευθείας, και για αυτό δεν έχουν κάποια ιδιαίτερη αξία για την επιχείρηση. Όταν όμως εφαρμοστούν τεχνικές εξόρυξης δεδομένων πάνω σε αυτά, μπορεί να συλλεχθεί σχετικό και πολύτιμο περιεχόμενο.

Όμως μέχρι οι υπολογιστές να είναι σε θέση να διαχειρισθούν απευθείας το περιεχόμενο, οι άνθρωποι καλούνται να βρουν τρόπους να χρησιμοποιήσουν τεχνολογίες διαχείρισης και εξόρυξης δεδομένων για να μπορούν να συλλέγουν και να διανέμουν το περιεχόμενο.

### **1.3 ΤΙ ΕΙΝΑΙ PORTAL ;**

Η λέξη Portal χρησιμοποιήθηκε για πρώτη φορά στο χώρο της επιστημονικής φαντασίας όπου υποδήλωνε την πύλη χάρη στην οποία οι ήρωες ενός έργου (ή οι παίκτες ενός παιχνιδιού) μπορούσαν να περάσουν άμεσα σε άλλους κόσμους και να βρεθούν αυτοστιγμεί οπουδήποτε επιθυμούσαν.[1.2 ]

Ήταν λοιπόν πολύ φυσικό για τους χρήστες του Internet (μεγάλο ποσοστό των οποίων αγαπά την επιστημονική φαντασία) να χρησιμοποιήσουν το ίδιο όρο για να περιγράψουν όσα sites τους άνοιγαν τις πύλες του δικτύου και τους παρείχαν τη δυνατότητα να «μεταφερθούν» σε όποιον «κόσμο» επιθυμούσαν. Και τα sites αυτά δεν ήταν άλλα από τα «Εργαλεία Αναζήτησης του Internet» (συνήθως κατάλογοι όπως το Yahoo!) τα οποία χρησιμεύουν πάντα ως σημείο εκκίνησης για τους χρήστες του δικτύου οπουδήποτε και αν θέλουν να βρεθούν. (Όταν δεν γνωρίζουμε πού θα βρούμε ό,τι αναζητούμε μέσα στο δίκτυο, επισκεπτόμαστε πάντοτε ένα εργαλείο αναζήτησης και εκείνο μας κατευθύνει στο σχετικό site.)

Έτσι, τα Portals λειτουργούσαν αρχικά ως απλές «πινακίδες» οι οποίες οδηγούσαν τους χρήστες στα άλλα sites του δικτύου. Σύντομα όμως οι διαχειριστές τους συνειδητοποίησαν ότι με τον τρόπο αυτό είχαν αποκτήσει τη δυνατότητα να κατευθύνουν τους χρήστες του Internet όπου οι ίδιοι επιθυμούσαν.[1.2]

Χαρακτηριστικό παράδειγμα της πορείας που πήραν τα πράγματα όταν έγινε κατανοητή η ισχύς των Portals είναι ο καιρός. Τα πρώτα χρόνια του δικτύου οι μετεωρολογικές πληροφορίες παρέχονταν από εξειδικευμένα sites όπως το [www.intellicast.com](http://www.intellicast.com). Οι υπεύθυνοι του Yahoo! όμως σκέφθηκαν ότι θα αποκτούσαν πολύ περισσότερα διαφημιστικά έσοδα αν παρείχαν οι ίδιοι αυτές τις πληροφορίες, αντί να «στέλνουν» απλώς τους χρήστες κάπου αλλού. Έτσι, δημιούργησαν μια αντίστοιχη υπηρεσία και αντί οι χρήστες να επισκέπτονται μόνο μια φορά το Yahoo!, να μαθαίνουν από εκεί για την ύπαρξη του [www.intellicast.com](http://www.intellicast.com) και στη συνέχεια να πηγαίνουν συνεχώς στο Intellicast, έγιναν, στην πλειοψηφία τους, τακτικοί επισκέπτες του Yahoo! Weather. Με τον τρόπο αυτό γεννήθηκε η σύγχρονη έννοια του Portal.[1.2] Πρόκειται για ένα site βασισμένο πάνω σε έναν κατάλογο πληροφοριών Internet το οποίο προσπαθεί να κάνει τους χρήστες να το επισκέπτονται κάθε φορά που ζητούν κάτι μέσα από το δίκτυο και δεν γνωρίζουν πού θα το βρουν. Αν ζητούν κάτι εξειδικευμένο τότε το Portal τους κατευθύνει στο ανάλογο site (αν υπάρχει). Αν όμως ζητούν κάτι κοινότυπο ή δημοφιλές (π.χ. ειδήσεις κτλ) τότε το Portal προσπαθεί να τους οδηγήσει στις υπηρεσίες που προσφέρει **το ίδιο**.

ENA PORTAL, στην ουσία είναι ένα **σύστημα διαχείρισης περιεχομένου (C.M.S.)**. Το σύστημα διαχείρισης περιεχομένου, δίνει την δυνατότητα στους διαχειριστές και τους συντάκτες τους, να προσθέτουν, αφαιρούν, επεξεργάζονται και γενικά να ανανεώνουν το περιεχόμενο (κείμενα, εικόνες, συνδέσμους, ανακοινώσεις κτλ) ενός δικτυακού τόπου / portal. Η διαδικασία αυτή, αποκαλείται ΣΥΝΤΗΡΗΣΗ.



### **Γιατί η συντήρηση είναι απαραίτητη;**

Η συντήρηση ενός εταιρικού δικτυακού τόπου είναι μια σημαντική παράμετρος για τη δημιουργία μιας επιτυχημένης παρουσίας στο Internet.

Τα sites δεν δημιουργούνται για να μένουν στατικά . Μία εταιρική ιστοσελίδα έχει μεγαλύτερη αξία για τον επισκέπτη όταν μπορεί να διατηρεί την πληροφορία της επίκαιρη. Συνεπώς, η ανανέωση του περιεχομένου καθώς και η προσθήκη νέων σελίδων, σύμφωνα με τις αυξανόμενες ανάγκες της μικρομεσαίας επιχείρησης, αποτελούν απαραίτητες διαδικασίες. Με τη συντήρηση, η επιχείρηση εξασφαλίζει ότι μπορεί να παρουσιάζει πάντοτε νέες πληροφορίες και να διατηρεί το επίπεδο λειτουργικότητας του δικτυακού της τόπου σε υψηλά επίπεδα.

Επιπλέον, η συντήρηση σχετίζεται άμεσα και με την αλληλεπίδραση με τον επισκέπτη. Είναι σημαντικό να προσφέρεται στο χρήστη η δυνατότητα να πλοηγείται σε ένα απλό και ευχάριστο περιβάλλον και να μπορεί να ανακαλύπτει νέα ενδιαφέροντα θέματα και πληροφορίες κάθε φορά που επισκέπτεται το site. Πολλές εταιρίες ανανεώνουν συχνά ακόμη και το γραφικό περιβάλλον του δικτυακού τους τόπου, έτσι ώστε να μη γίνεται ποτέ "βαρετό" στο χρήστη.

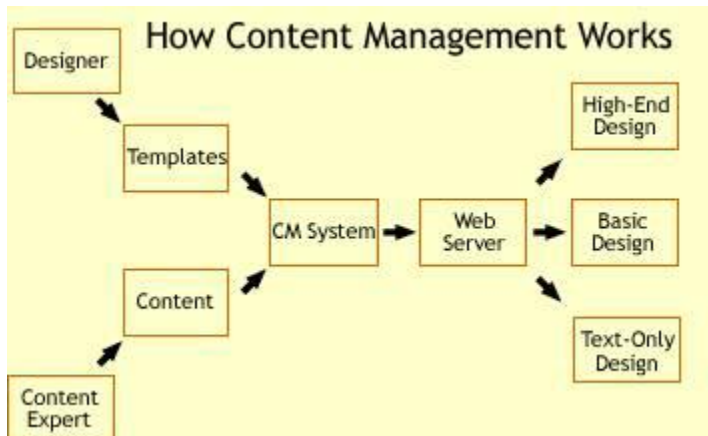
### **1.4 ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ**

Το σύστημα διαχείρισης περιεχομένου (CMS) είναι μια εφαρμογή που χρησιμοποιείται για να δημοσιεύσει ιστοσελίδες στο διαδίκτυο.[1.14] Τα Συστήματα Διαχείρισης Περιεχομένου, μπορούν να χρησιμοποιηθούν για να κατασκευάσουν Web Sites όπως:

- Εταιρικούς
- Προσωπικούς
- Εκπαιδευτικούς
- Ενημερωτικούς

Και γενικά web sites που μπορούν να καλύψουν σχεδόν όλη την γκάμα των ενδιαφερομένων. Το περιεχόμενο που μπορεί να χρησιμοποιηθεί περιλαμβάνει κείμενα, εικόνα, ήχους, video, ηλεκτρονικά αρχεία και γενικά οτιδήποτε μπορεί να διανεμηθεί μέσω του διαδικτύου. Δημιουργείται και εγκαθίσταται από τους σχεδιαστές ιστοσελίδων, αλλά προορίζεται για χρήση από τελικούς χρήστες.

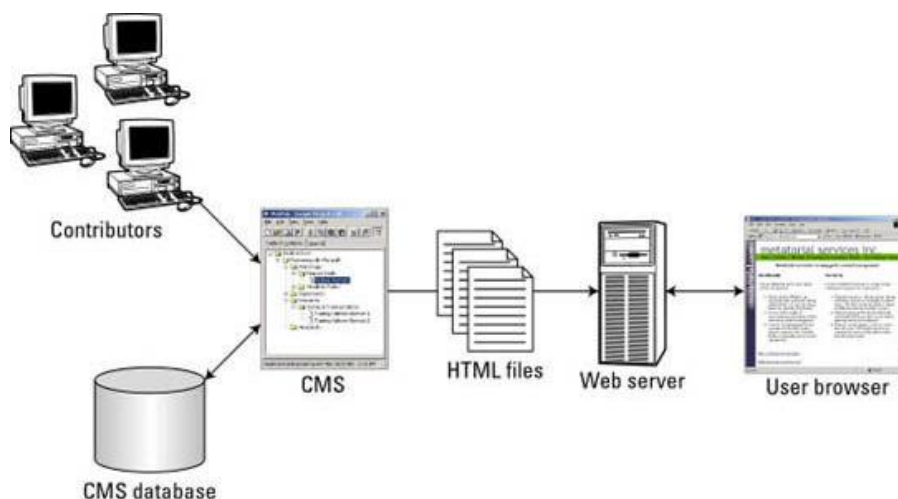
Αρχικά, προσφέρει έναν εύκολο, εύχρηστο τρόπο ενημέρωσης περιεχομένου. Αυτό συνήθως γίνεται με τη χρήση ενός συστήματος πλοήγησης (browser). Ο χρήστης απλά εισάγει το νέο κείμενο και το αποθηκεύει. Η ιστοσελίδα ενημερώνεται αμέσως! Το ίδιο απλό είναι να προστεθούν νέες σελίδες, να διαγραφούν παλαιές, ή να αναδιαμορφωθεί μια ιστοσελίδα ώστε να συμβαδίζει με νέες απαιτήσεις ή προδιαγραφές, και το σημαντικότερο: Χωρίς να απαιτείται επέμβαση στον πηγαίο κώδικα.



Το Σύστημα Διαχείρισης Περιεχομένου αυτοματοποιεί διάφορες διαδικασίες όπως η διατήρηση της εμφάνισης των σελίδων σε όλο το Web Site καθώς και η δημιουργία των σχετικών μενού, συνδέσμων κλπ. Επίσης η ύπαρξη και αρκετών άλλων εργαλείων διαχείρισης, επιτρέπει την εστίαση στις λέξεις και όχι στην τεχνολογία.

Ένα CMS πρέπει να υποστηρίζει τις παρακάτω δυνατότητες:[1.14]

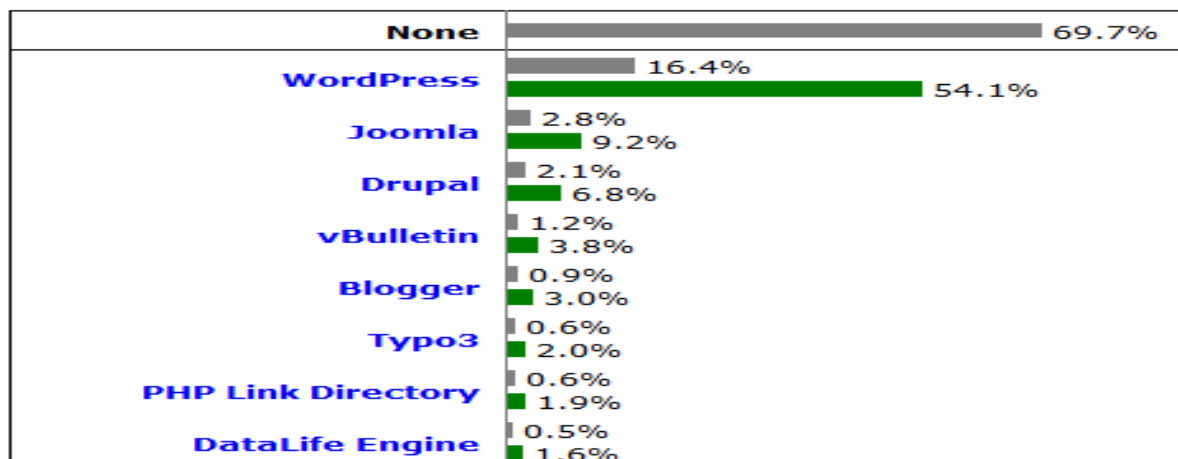
- Εύκολη διαχείριση περιεχομένου μέσω ενός browser.
- Διαχωρισμός περιεχομένου και εμφάνισης (για παράδειγμα να μπορούμε οποιαδήποτε στιγμή να αλλάξουμε το φόντο ή το στυλ της γραμματοσειράς μια φορά και να εφαρμοστεί σε όλες τις σελίδες).
- Διαφορετικούς ρόλους και επίπεδα για τους χρήστες του.
- Δυνατότητα δημοσίευσης περιεχομένου από χρήστες έπειτα από έγκριση του διαχειριστή.
- Δυνατότητα κατηγοριοποίησης του περιεχομένου ώστε να είναι ευκολότερη η διαχείρισή του.



### 1.5 ΔΗΜΟΦΙΛΗ ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

WordPress, Joomla! και Drupal είναι τα τρία δημοφιλέστερα συστήματα διαχείρισης περιεχομένου. Τα δύο τελευταία ανταγωνίζονταν εδώ και καιρό για την πρώτη θέση στα CMS αφού το WordPress μέχρι πρόσφατα θεωρούταν ως μία πλατφόρμα blogging και μόνο τα τελευταία δύο χρόνια άρχισε να αναπτύσσεται ως ένα ολοκληρωμένο σύστημα διαχείρισης περιεχομένου, φτάνοντας μάλιστα στην πρώτη θέση στη χρήση από δικτυακού τύπους. Και τα τρία συστήματα είναι ανοιχτού κώδικα και έχουν αναπτυχθεί με βοήθεια από χιλιάδες ή εκατοντάδες μέλη από τις αντίστοιχες κοινότητες. Καθένα από αυτά έχει πλεονεκτήματα και μειονεκτήματα, αλλά ουσιαστικά μπορεί να χρησιμοποιηθεί οποιοδήποτε από τα τρία για την δημιουργία αποτελεσματικών και επαγγελματικής εμφάνισης ιστοτόπων, με μεγάλη γκάμα δυνατοτήτων και για οποιαδήποτε χρήση, από ένα απλό προσωπικό blog έως πλήρες ecommerce site.[1.14]

Ακολουθεί διάγραμμα που δείχνει την χρήση των γνωστότερων CMS αυτή τη στιγμή.



## ΚΕΦΑΛΑΙΟ 2

### ΓΝΩΣΤΙΚΟ ΥΠΟΒΑΘΡΟ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ



## 2.1 ΙΣΤΟΣΕΛΙΔΑ

Η ιστοσελίδα είναι ένα αρχείο που περιέχει πληροφορίες που είναι προορισμένες για δημοσίευση στον Παγκόσμιο Ιστό (www).[1.3] Μία ιστοσελίδα είναι προσβάσιμη από ένα Φυλλομετρητή (web browser). Οι πληροφορίες της είναι συνήθως γραμμένες με HTML ή XHTML.

Μία ιστοσελίδα μπορεί να περιέχει ένα σύνολο πληροφοριών όπως κείμενα, γραφικά, φωτογραφίες, video, ήχους, χρώματα ή ακόμα και διάφορα αρχεία. Οι επισκέπτες της ιστοσελίδας ονομάζονται web clients.

## 2.2 ΙΣΤΟΤΟΠΟΣ

Ο ιστότοπος είναι το σύνολο των ιστοσελίδων που είναι συνδεδεμένες μεταξύ τους με υπερσυνδέσεις (hyperlinks) και περιγράφουν μια δραστηριότητα. Ένας ιστότοπος μπορεί να είναι επαγγελματικός, προσωπικός ή να ανήκει σε έναν οργανισμό ή μια ομάδα. Το σύνολο των ιστοτόπων που υπάρχουν απαρτίζουν στην ουσία το παγκόσμιο ιστό(www).[1.9]

## 2.3 WEB BROWSERS

Οι Web Browsers είναι τα προγράμματα τα οποία μας επιτρέπουν να βλέπουμε σελίδες στο internet. Κάθε ένας web browser έχει τις δικές του δυνατότητες, επεκτάσεις, εχθρούς και φίλους. Αυτή τη στιγμή υπάρχει μία πλειάδα από web browsers που ικανοποιούν τα γούστα και του πιο απαιτητικού χρήστη. [1.4]

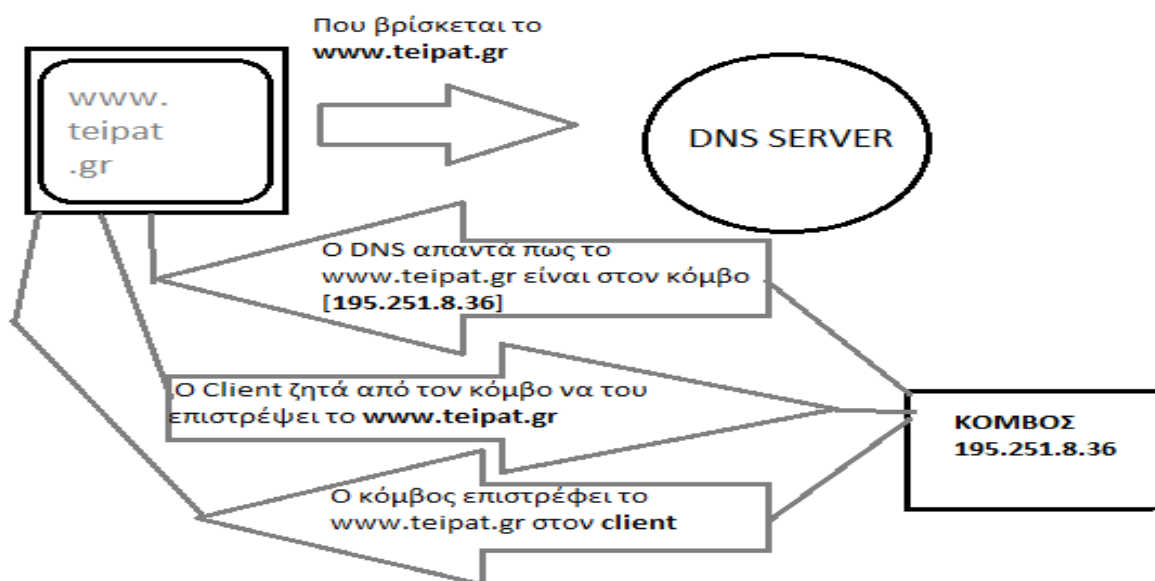
Οι πιο δημοφιλείς web browsers είναι :

1. Internet Explorer 8
2. Internet Explorer 9
3. Mozilla Firefox
4. Opera
5. Safari
6. Google Chrome
7. Netscape Navigator (σταμάτησε πλέον η υποστήριξη του)

Το κύριο χαρακτηριστικό όλων είναι ότι διανέμονται δωρεάν οπότε ο κάθε χρήστης έπειτα από δοκιμή μπορεί να βρει ποιος απ' όλους έχει όλα τα στοιχεία ώστε η πλοήγηση του στο internet να γίνεται πιο ευχάριστη. Κάποιοι που ασχολείται με την ανάπτυξη και την κατασκευή ιστοσελίδων προτείνεται να τους έχει αν όχι όλου σίγουρα τους δημοφιλέστερους ώστε να ελέγχει την εμφάνιση των ιστοσελίδων του από διαφορετικά προγράμματα, δεδομένων των μικροδιαφορών προβολής που υπάρχουν.

## 2.4 DOMAIN NAME

Τα Domain Names είναι στην ουσία η ταυτότητα των ιστοτόπων. Όταν θέλουμε να «καλέσουμε» μία ιστοσελίδα μέσα από ένα web browser θα πρέπει να γράψουμε στη γραμμή διευθύνσεων το όνομα της πχ. www.teipat.gr. Γράφοντας αυτό το όνομα στην ουσία ο web browser ψάχνει να βρει σε ποιο Web server είναι αποθηκευμένη η συγκεκριμένη σελίδα. Η αναζήτηση γίνεται με ερώτημα στον Domain Name Server (DNS).[1.16] Εάν το Domain Name που πληκτρολογήσαμε είναι υπαρκτό το αποτέλεσμα, ο Domain Name Server θα ενημερώσει τον web browser με τον κόμβο στον οποίο βρίσκεται η ιστοσελίδα, και εν συνεχεία, ο web browser θα ζητήσει την σελίδα από τον διακομιστή web (περισσότερες πληροφορίες ακολουθούν). Αν ο διακομιστής πράγματι διαθέτει την εν λόγω σελίδα, τότε αυτή θα εμφανιστεί στον Web browser.



## 2.5 ΣΤΑΤΙΚΕΣ ΙΣΤΟΣΕΛΙΔΕΣ

Στατικές Ιστοσελίδες είναι οι ιστοσελίδες που δημιουργούνται τοπικά στον υπολογιστή μας συγγράφοντας κώδικα HTML ή χρησιμοποιώντας κάποιο πρόγραμμα WYSIWYG (What you see is what you get) όπως είναι το Adobe Dreamweaver.[1.13] Οι στατικές ιστοσελίδες είναι πολύ εύκολες στην κατασκευή τους αλλά πολύ δύσκολες στην ανανέωσή τους. Αυτό συμβαίνει γιατί όταν θέλουμε να προσθέσουμε ή να αλλάξουμε το περιεχόμενό τους πρέπει να ανοίξουμε το κατάλληλο πρόγραμμα να κάνουμε τις αλλαγές που θέλουμε, να αποθηκεύσουμε και στη συνέχεια να «ανεβάσουμε» στο Web Server τις ενημερωμένες σελίδες. Όλα αυτά προϋποθέτουν γνώσεις περισσότερες από το μέσο χρήστη και φυσικά περισσότερο χρόνο. Άλλο παράδειγμα είναι ότι εάν θέλουμε να αλλάξουμε το όνομα ενός στοιχείου μενού σε όλες τις σελίδες μας (ας υποθέσουμε 30 συνολικά) θα πρέπει να ανοίξουμε και τις 30 σελίδες να αλλάξουμε το όνομα να τις αποθηκεύσουμε και να τις «ανεβάσουμε» τελικά στον Web Server. Φυσικά η ανανέωση μπορεί να είναι χρονοβόρα, όμως στον αντίποδα η στατικές σελίδες, επειδή δεν απαιτούν οποιαδήποτε επεξεργασία από τον διακομιστή για να αποσταλούν στον Web Browser, σε περιπτώσεις αυξημένου φόρτου ανταποκρίνονται ταχύτερα.

## 2.6 ΔΥΝΑΜΙΚΕΣ ΙΣΤΟΣΕΛΙΔΕΣ

Σε αντίθεση με τις στατικές οι δυναμικές ιστοσελίδες δεν δημιουργούνται στον υπολογιστή μας αλλά απ' ευθείας στον Web Server. Δίνοντας τις κατάλληλες εντολές οι ιστοσελίδες μας μπορούν να αλλάξουν, διαγραφούν ή να προστεθούν νέες. Και όλα αυτά χρησιμοποιώντας μια φιλική στο χρήστη περιοχή διαχείρισης τόσο απλή όσο μια εφαρμογή επεξεργασίας κειμένου όπως το Microsoft Word. Τα συστήματα αυτά ονομάζονται Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems ή CMS). Μερικά από τα γνωστότερα CMS ανοικτού κώδικα, είναι το Joomla και το Drupal. [1.14] Σε αρκετές περιπτώσεις, δημιουργούνται CMS από το μηδέν (from scratch) και ο βασικός στόχος ενός τέτοιου εγχειρήματος είναι να καλυφθούν οι ανάγκες με τον καλύτερο δυνατό τρόπο.

## 2.7 ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΤΥΠΟΥ ΣΥΣΤΗΜΑΤΟΣ PORTAL ΣΕ PHP

Στο διαδίκτυο, υπάρχει πληθώρα συστημάτων διαχείρισης περιεχομένου (C.M.S.) ανοικτού κώδικα, μεταξύ των οποίων τα πιο γνωστά Joomla, Drupal και άλλα. Σε αρκετές λοιπόν περιπτώσεις, η χρήση ενός έτοιμου συστήματος αποτελεί μια εύκολη λύση και είναι ικανή να καλύψει μεγάλο μέρος των αναγκών. Αυτή όμως η λύση αντιμετωπίζει μερικά προβλήματα:

### 1. Αυθεντικότητα:

Στις περισσότερες περιπτώσεις χρήσης ενός έτοιμου συστήματος, δεν μένουμε απλά στην εγκατάστασή του, τοποθετώντας του περιεχόμενο, αλλά προχωρούμε σε πλήθος μετατροπών ώστε στην πραγματικότητα να φτιάξουμε ένα πρωτότυπο site που θα καλύπτει κάθε μας ανάγκη, ουσιαστικά με κώδικα που έγραψαν άλλοι χωρίς κανείς να μπορεί να εγγυηθεί πως το σύστημα με τις τελευταίες μετατροπές που υπέστη θα είναι σε θέση: [1.14]

- a) Να δουλεύει το ίδιο αποδοτικά με βάση όσα προέβλεψε ο κατασκευαστής του κατά την υλοποίησή του.
- b) Να είναι συμβατό με μελλοντικές αναβαθμίσεις (ασφαλείας η μη) του κατασκευαστή του.

### 2. Ασφάλεια

Η χρήση έτοιμων συστημάτων διαχείρισης περιεχομένου, μπορεί κατά συνθήκη να επιφέρει κινδύνους στον τομέα της ασφάλειας του δικτυακού μας τόπου, αφού αρκετά συχνά τυγχάνουν σφοδρών επιθέσεων hacking ή cracking, επιθέσεις οι οποίες ενδεχομένως να φέρουν στο φώς τρωτά σημεία (vulnerabilities) του εκάστοτε γνωστού συστήματος c.m.s. που χρησιμοποιείται κατά χιλιάδες σε όλο τον κόσμο, τα οποία όχι μόνο δημοσιεύονται σε δικτυακούς τόπους που σχετίζονται με το hacking αλλά δημιουργούνται εκτός από οδηγούς hacking για το συγκεκριμένο c.m.s. ακόμα και αυτοματοποιημένα εργαλεία που αναλαμβάνουν με το πάτημα ακόμα και ενός κουμπιού να εκτελέσουν όλες τις ενέργειες που είτε θα αλλοιώσουν το περιεχόμενο, είτε θα αποσπάσουν προσωπικά στοιχεία ή κωδικούς πρόσβασης είτε θα θέσουν εντελώς εκτός λειτουργίας τον δικτυακό τόπο. Όσο πιο γνωστό είναι το σύστημα διαχείρισης που χρησιμοποιούμε, όσο και περισσότεροι hackers/crackers θα ενδιαφέρονταν να ανακαλύψουν κάποια τρύπα ασφαλείας σε αυτό και να δημοσιεύσουν σε forums το πώς τα κατάφεραν. Επιπροσθέτως, τα περισσότερα συστήματα διαχείρισης περιεχομένου είναι ικανά να δεχθούν προσθήκες (components ή/και modules) που επεκτείνουν τις δυνατότητές τους. Στην περίπτωση αυτή θα πρέπει να μελετάμε ξεχωριστά



τους κινδύνους που προκύπτουν για κάθε πρόσθετο ξεχωριστά: Το πόσο ανθεκτικό είναι σε επιθέσεις διαφόρων τύπων (MySQL Injections και άλλες). [1.12]

Αν για παράδειγμα, για την υλοποίηση του portal χρησιμοποιούσαμε την λύση Joomla, αν δεν προχωρούμε στις προτεινόμενες από τον κατασκευαστή αναβαθμίσεις ασφαλείας, τότε η υπάρχουσα εγκατάσταση καθίσταται ευάλωτη σε πιθανή επίθεση ασφαλείας. Οι λόγοι που λειτουργούν αποτρεπτικά ως προς την αναβάθμιση είναι κυρίως οι εξής:

- 1) Οι επεμβάσεις που έχουμε πραγματοποιήσει στον πρωτότυπο κώδικα όπως αναφέραμε προηγουμένως, θα χαθούν από την αναβάθμιση αυτή.
- 2) Κάποια από τα πρόσθετα που έχουμε εγκαταστήσει στο σύστημα διαχείρισης περιεχομένου (π.χ. πρόσθετο διεξαγωγής ψηφοφοριών) δεν είναι συμβατό με την ενημερωμένη έκδοση.

Γενικά, όπως και στην περίπτωση του Joomla αλλά και πολλών άλλων συστημάτων διαχείρισης περιεχομένου που ο κώδικάς τους είναι προσβάσιμος στο ευρύ κοινό αλλά και χρησιμοποιούνται ευρέως, λάθη, παραλείψεις, αδυναμίες και τρύπες ασφαλείας μπορούν εύκολα να βρεθούν.

### 3. Αποτελεσματικότητα

Όσο πιο εξειδικευμένες είναι οι ανάγκες μας στην κατασκευή ενός δικτυακού τόπου όπως είναι το portal, τόσο λιγότερο αποτελεσματικές είναι οι «έτοιμες λύσεις». Αυτό συμβαίνει γιατί τα συστήματα διαχείρισης περιεχομένου αυτά, δημιουργήθηκαν για να καλύψουν τον μέσο όρο των αναγκών των περισσότερων χρηστών τους.[1.12] Ακόμα και στον τομέα της διαχείρισης, οι διαθέσιμες επιλογές προσπαθούν να καλύψουν αυτόν τον μέσο όρο αναγκών και εκ των πραγμάτων δεν προσφέρουν κάποια ουσιαστική εξειδίκευση. Για παράδειγμα, ενώ σε ένα portal δημιουργημένο εξολοκλήρου από το μηδέν για να καλύψει τις ανάγκες του Α.Τ.Ε.Ι. στην κατηγορία «**Ανακοινώσεις**», τα πράγματα θα ήταν τόσο απλά στην διαχείριση όσο ένα κλικ στην επιλογή «**προσθήκη ανακοίνωσης**», σε μία έτοιμη λύση σαν το Joomla η διαδικασία προσθήκης ανακοίνωσης γίνεται αρκετά πολύπλοκη, καθώς πρέπει να ακολουθηθεί η διαδικασία «προσθήκης άρθρου» στην ενότητα «ανακοινώσεις» και στην κατηγορία «γραμματεία». Το σημαντικότερο όλων όμως, είναι πως όσες περισσότερες ανάγκες καλείται να καλύψει ένα συγκεκριμένο σύστημα, τόσο πολύπλοκη είναι η λειτουργία του και αυτό το κάνει να έχει ανάγκη όλο και περισσότερους πόρους, με αποτέλεσμα στον ίδιο ακριβώς διακομιστή να έχει την ικανότητα εξυπηρέτησης λιγότερων ταυτόχρονων χρηστών συγκριτικά με το σύστημα διαχείρισης που δημιουργήθηκε εξολοκλήρου αποκλειστικά για τις ανάγκες του portal.

Με το να δημιουργήσουμε όμως το δικό μας προσαρμοσμένο / πρωτότυπο σύστημα διαχείρισης περιχομένου, ελαχιστοποιούμε τους προαναφερόμενους κινδύνους και αδυναμίες που προκύπτουν λόγω της πολυπλοκότητας που αυξάνεται σε ένα έτοιμο σύστημα προκειμένου να επιτευχθεί σχεδόν το ίδιο αποτέλεσμα με το πρωτότυπο, απολαμβάνοντας όλα τα πλεονεκτήματα που προκύπτουν από ένα σύστημα εξατομικευμένο στις ανάγκες μας.

Επιπροσθέτως όντας οικείοι με τον κώδικα του συστήματος, εφόσον τον έχουμε παράγει οι ίδιοι, μπορούμε με μεγαλύτερη ευκολία να προβούμε είτε σε μετατροπές είτε σε νέες προσθήκες.

## 2.8 TO PORTAL ΤΟΥ ΤΕΙ ΑΜΑΛΙΑΔΑΣ

Το Portal του ΑΤΕΙ Αμαλιάδας, εφαρμόζει όλα όσα αναφέρθηκαν προηγουμένως, μιας και αποτελεί ένα project δημιουργημένο εξολοκλήρου εκ του μηδενός, ένας πρωτότυπος δυναμικός δικτυακός τόπος προσαρμοσμένος στις ανάγκες ενός portal που στόχο έχει να συγκεντρώσει όλες τις πληροφορίες που ενδιαφέρουν τα μέλη της εκπαιδευτικής κοινότητας του ΑΤΕΙ. Επειδή ακριβώς αποτελεί ένα πρωτότυπο σύστημα, δηλαδή είναι δημιουργημένο ακριβώς με βάση της πραγματικές ανάγκες ενός portal που απευθύνεται σε φοιτητές και καθηγητές, επιφέρει καλύτερα αποτελέσματα συγκριτικά με οποιαδήποτε «έτοιμη λύση». Οι δυνατότητες παραμετροποίησης, προσαρμογής ή / και μετατροπής είναι απεριόριστες ενώ η πολυπλοκότητα τόσο στην υλοποίηση όσο και στην χρήση είναι εξαιρετικά μικρή, καθώς για παράδειγμα, εάν οποιοσδήποτε διαχειριστής επιθυμήσει να προσθέσει μια ανακοίνωση, αρκεί μόνο να επιλέξει την διαδικασία «προσθήκη ανακοίνωσης» από την διαχείριση, συγκριτικά με την «έτοιμη λύση» στην οποία θα έπρεπε να ακολουθήσει την μακρόσυρτη διαδικασία προσθήκης άρθρου σε κατηγορία που επιτελεί τον ρόλο του πίνακα ανακοινώσεων, διαδικασία συχνά δυσνόητη από αρκετούς χρήστες. Χάριν στην μειωμένη πολυπλοκότητα του υλοποίηση, το σύστημα προσφέρει μεγαλύτερη ασφάλεια, αφού ο κίνδυνος ύπαρξης λογικών λαθών (bugs) είναι αρκετά μικρός ως ανύπαρκτος, αφού όσο περισσότερο στοχευμένο είναι το λογισμικό τόσο λιγότερο πολύπλοκη υλοποίηση έχει. Ένας ακόμη σημαντικός λόγος που αυξάνει δραματικά την ασφάλεια του συστήματος, είναι και ο εξής:

Το Portal, ως λογισμικό που δημιουργήθηκε για την κάλυψη των αναγκών του ΑΤΕΙ Αμαλιάδας, κατά συνέπεια θα χρησιμοποιηθεί μόνο από την εκπαιδευτική κοινότητα του ΑΤΕΙ και πρόσβαση στον πηγαίο του κώδικα θα έχουν μόνο όσοι έχουν αναλάβει το στήσιμο και την συντήρησή του. Όσο λιγότεροι γνωρίζουν το πώς λειτουργεί ένα σύστημα, λογισμικό, web site κτλ (πόσο μάλλον να αναλύσουν τον κώδικά του και ενδεχόμενα να ανακαλύψουν αδυναμίες που δεν είχαν έως τότε ανακαλυφθεί από τον κατασκευαστή), τόσο μειώνεται η πιθανότητα να βρεθεί επίδοξος cracker που

θα προσπαθήσει να εντοπίσει τρύπα ασφαλείας, άρα, τόσο περισσότερο ασφαλές είναι για τους χρήστες και για τα δεδομένα που διατηρεί.

Ο χρήστης πληκτρολογεί την διεύθυνση του portal. Εκτελείται ερώτηση στο διακομιστή DNS, από τον οποίο λαμβάνεται ως απάντηση η διεύθυνση IP του κόμβου που φιλοξενεί το Portal. Εν συνεχεία, ο φυλλομετρητής του επισκέπτη, ζητά από τον κόμβο να του αποστείλει την κεντρική σελίδα του Portal.

## ΚΕΦΑΛΑΙΟ 3

# ΥΛΟΠΟΙΗΣΗ PORTAL



```
if ($data['type'] == "dropdown") {
    $count = $data['number'];
    $call = $data['contents'];
    echo "<select name=\"$data['name']\"";
    for($i = 0; $i = $count; $i++) {
        $value = $i + 1;
        echo "<option value=\"$value\"";
        if($value == $data[5]) {
            echo " selected";
        }
        echo "> $call[$i]\n";
    }
}
if ($data['type'] == "dropdown2")
    $count = $data['number'];
    $call = $data['contents'];
    echo "<select name=\"$data['name']\"";
    foreach ($call as $c) {
        echo "\t\t<option value=\"$c\"";
        if($value == $data['default'])
            echo " selected";
    }
    echo ">";
}
```

### 3.1 ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΥΛΟΠΟΙΗΣΗ

Στην παρούσα εργασία, δημιουργούμε ένα σύστημα διαχείρισης περιεχομένου για την λειτουργία του portal του Α.Τ.Ε.Ι. Αμαλιάδας, με χρήση των τεχνολογιών PHP 5.3 , MySQL, HTML, Javascript και CSS.

Με την ολοκλήρωσή του, το σύστημα τεθεί σε λειτουργία, θα παρέχει τις ακόλουθες δυνατότητες:

- **Κεντρική Σελίδα** που θα συγκεντρώνει και θα παρουσιάζει τις επιμέρους πληροφορίες από τις υποενότητες του portal, όπως οι τελευταίες ανακοινώσεις,
- **Επιμέρους Σελίδες** που παρουσιάζουν κάθε πληροφορία που υπάρχει στο portal ξεχωριστά.
- **Σελίδες για μέλη:** Συγκεκριμένες σελίδες που είναι προσβάσιμες μόνο από εγγεγραμμένους χρήστες.
- **Συνεδρίες Χρηστών:** Οι χρήστες μπορούν να δημιουργήσουν λογαριασμό, να συνδεθούν και να αποσυνδεθούν.
- **Πίνακας Διαχείρισης:** Ο πίνακας διαχείρισης περιλαμβάνει όλες εκείνες τις επιλογές που επιτρέπουν στον διαχειριστή να προσθέτει, να αφαιρεί και να επεξεργάζεται τα περιεχόμενα του portal.

### 3.2 ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

#### 1) PHP 5.3

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο, δημιουργημένη από την εταιρία Zend.[3] Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (όπως ο Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. \*.php, \*.php4, \*.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου

(mod\_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοτόπους, FastCGI Process Manager (FPM).

Η PHP 5.3 ανήκει στις τελευταίες εκδόσεις που κυκλοφόρησαν από την Zend και ήδη χρησιμοποιείται σε αρκετούς διακομιστές. Έχει ενσωματώσει αρκετές νέες λειτουργίες (όπως anonymous functions και άλλα) που κάνουν ακόμα πιο εύκολο τον προγραμματισμό, καθώς και έχει δεχθεί τις κατάλληλες βελτιώσεις ώστε να γίνει ακόμα ταχύτερη. Συγκεκριμένα, σύμφωνα με δοκιμές που έγιναν με την PHP 5.3, είναι 30% ταχύτερη σε σχέση με την έκδοση 5.2.x.[1.11] Σύμφωνα με τις ίδιες μετρήσεις, στον πίνακα που ακολουθεί, παρουσιάζονται οι διαφορές στην ταχύτητα ορισμένων συστημάτων διαχείρισης περιεχομένου ανοικτού κώδικα:

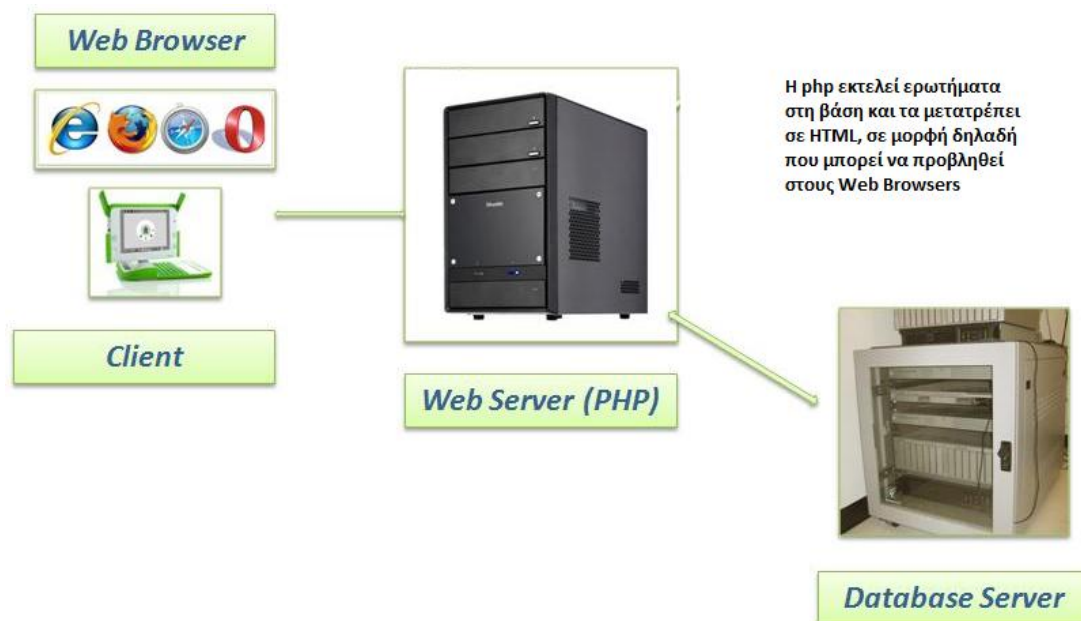
- **Drupal 20% ταχύτερο**
- **Qdig 2% ταχύτερο**
- **Typo3 30% ταχύτερο**
- **Xoops 10% και τέλος**
- **WordPress 15% βελτίωση στη ταχύτητα**

## 2) **MYSQL**

Η **MySQL** είναι ένα σύστημα διαχείρισης **σχεσιακών βάσεων δεδομένων** που μετρά περισσότερες από **11** εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντου Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.[1.12]

Η **MySQL** είναι λογισμικό ανοικτού κώδικα, ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική **MySQL AB**, η οποία σήμερα ανήκει στην **Oracle**.

Η **MySQL** είναι δημοφιλείς βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το **Flickr**, το **YouTube**, η **Wikipedia**, το **Google**, το **Facebook** και το **Twitter**.



### 3) HTML

Η HTML (ακρωνύμιο του αγγλικού Hypertext Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα).[1.7] Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Η προέλευσή της έχει ρίζες στο 1980, όταν ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο.[2] Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Από τότε η HTML έχει αναπτυχθεί σε μεγάλο βαθμό, φθάνοντας σήμερα στην έκδοση 5 η οποία χάρη στην πληθώρα δυνατοτήτων της, παρέχει την δυνατότητα εκτός από δικτυακούς τόπους και της δημιουργίας του γραφικού περιβάλλοντος desktop η mobile applications.

#### 4) CSS / CASCADING STYLESHEETS

Τα Cascading Stylesheets είναι μια γλώσσα φύλλων στυλ και χρησιμοποιείται για την περιγραφή της εμφάνισης και της διαμόρφωσης ενός εγγράφου, γραμμένου σε HTML.[1.13] Η γλώσσα CSS σχεδιάστηκε αρχικά για να επιτρέπει τον διαχωρισμό του περιεχομένου που είναι γραμμένο σε HTML από την περιγραφή της εμφάνισής του (τοποθέτηση επιπέδων, χρώματα, γραμματοσειρές και άλλα). Στην τελευταία έκδοση ( CSS 3 ) παρέχει ακόμα και δυνατότητες animation οι οποίες όμως δεν έχουν υιοθετηθεί επαρκώς από όλους τους φυλλομετρητές με αποτέλεσμα η χρήση αυτών των δυνατοτήτων να μην είναι ακόμα τόσο διαδεδομένη.

#### 5) JAVASCRIPT

Η JavaScript είναι γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών, η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) σε ιστοσελίδες. Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε beta εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator έκδοση 2.0 τον Σεπτέμβριο του 1995. LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape έκδοση 2.0B3

#### 6) APACHE HTTPD

Ο Apache είναι ο πιο διαδεδομένος Διακομιστής Ιστού σε συστήματα Linux πρωτίστως αλλά και σε Windows μέσω του layer συμβατότητας Posix.[1.10] Οι Διακομιστές Ιστού χρησιμοποιούνται για να εξυπηρετούν και να σερβίρουν Σελίδες Ιστού που ζητούνται από πελάτες υπολογιστές. Οι πελάτες τυπικά ζητούν και βλέπουν Σελίδες Ιστού χρησιμοποιώντας όπως αναφέραμε προηγουμένως εφαρμογές Φυλλομετρητών Ιστού όπως οι Firefox, Opera, ή Mozilla.

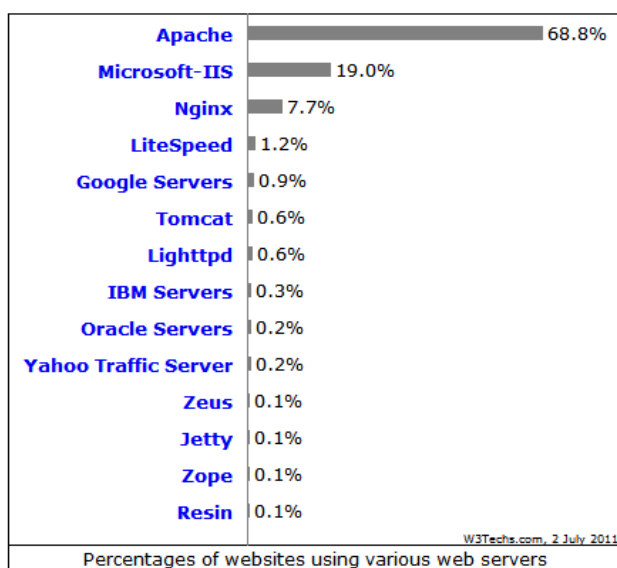
Το πιο κοινό πρωτόκολλο που χρησιμοποιείτε για τη μεταφορά ιστοσελίδων είναι το Πρωτόκολλο Μεταφοράς Υπερκειμένου (Hyper Text Transfer Protocol (HTTP)). Πρωτόκολλα όπως το HTTP πάνω από το Στρώμα Ασφαλούς Υποδοχής (Secure Sockets



Layer (HTTPS)), και το Πρωτόκολλο Μεταφοράς Αρχείων (File Transfer Protocol (FTP)), ένα πρωτόκολλο για την αποστολή και λήψη αρχείων, υποστηρίζονται επίσης.

Οι Διακομιστές Ιστού Apache συχνά χρησιμοποιούνται σε συνδυασμό με τη μηχανή βάσης δεδομένων MySQL και την PHP. Αυτή η σύνθεση ονομάζεται LAMP (Linux, Apache, MySQL and Perl/Python/PHP) η σε περιβάλλον Windows WAMP (αντικαθιστώντας το Linux με Windows) και σχηματίζει μια ισχυρή και αυτοδύναμη πλατφόρμα για την ανάπτυξη εφαρμογών βασισμένες στον Ιστό.[1.10]

Στο διάγραμμα που ακολουθεί βλέπουμε ότι ο Apache χρησιμοποιείται ευρέως και με το μεγαλύτερο ποσοστό:



## 7) HTTP COOKIE

Ένα Cookie, γνωστό και ως HTTP cookie ή web cookie ή browser cookie, είναι συνήθως ένα κομμάτι δεδομένων το οποίο αποστέλλεται από τον εκάστοτε website και αποθηκεύεται στον φυλλομετρητή του χρήστη όταν ο τελευταίος το επισκέπτεται.[1.5] Όταν ο ίδιος χρήστης ξανασυνδεθεί στο ίδιο website, τα δεδομένα που είναι αποθηκευμένα στα cookies ανακτώνται από τον αυτό, ώστε να είναι σε θέση να γνωρίζει τις προηγούμενες ενέργειες του χρήστη. Τα Cookies σχεδιάστηκαν για να αποτελέσουν έναν αξιόπιστο μηχανισμό των web sites για να υποστηρίξουν υπηρεσίες χρηστών-μελών (authentication cookies), βοηθώντας τα να «θυμούνται» ποιο μέλος έχει συνδεθεί στο λογαριασμό του ώστε να του παρέχει εξειδικευμένες υπηρεσίες και ποιο όχι, καθώς επίσης και τι ενέργειες έκανε ο ίδιος χρήστης στο παρελθόν. Αυτό μπορεί να περιλαμβάνει ακόμα και πληροφορίες για το ποια κουμπιά ή

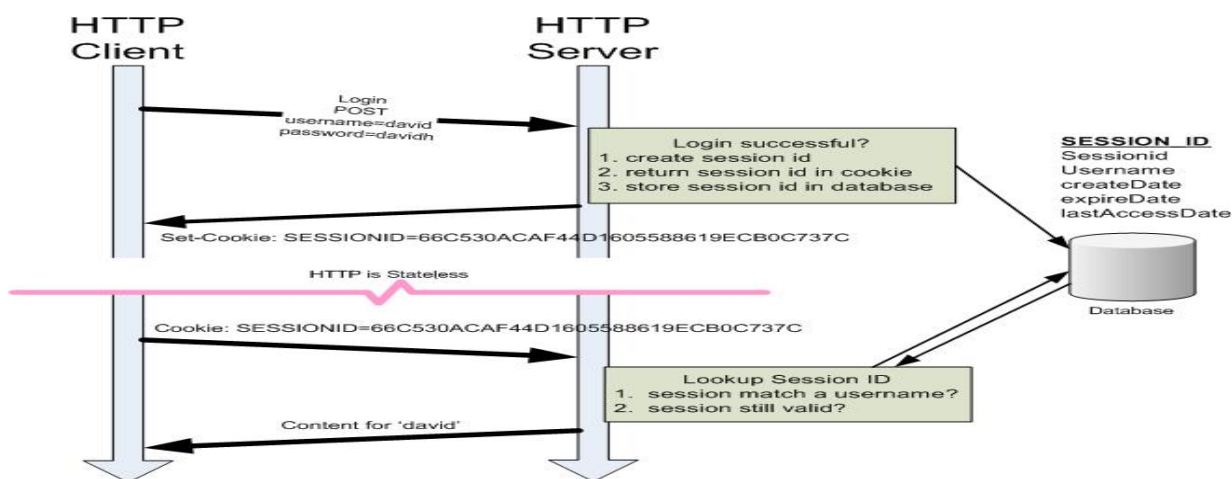
συνδέσμούς πάτησε ο χρήστης στο παρελθόν ή σε ποια σημεία του website στάθηκε περισσότερο.

**Τα Authentication Cookies** συσχετίζονται με τον σειριακό αριθμό της συνεδρίας που δημιουργήθηκε στον διακομιστή μετά την επιτυχημένη πιστοποίηση του χρήστη (έλεγχος εγκυρότητας username ή email και κωδικού) κατά την διαδικασία εισόδου.

Μπορεί τεχνικώς τα cookies να αδυνατούν να μεταδώσουν κάποιον ιό ή άλλο λογισμικό παραβίασης, όμως συχνά η κακή τους χρήση από τρίτους επιχειρεί ενέργειες που παραβιάζουν τα προσωπικά δεδομένα, όπως για παράδειγμα ο συσχετισμός ενός ιστορικού ενεργειών και επισκέψεων σε διάφορα websites με έναν συγκεκριμένο χρήστη, κάτι που συχνά οδηγεί τις αρχές προστασίας δεδομένων σε Ευρώπη και Αμερική να αναλαμβάνουν δράση.

Παρόλα αυτά, η σημαντικότερη και πιο διαδεδομένη εκδοχή των cookies είναι τα authentication cookies που βοηθά τους διακομιστές και τα websites να αναγνωρίζουν ποιος χρήστης είναι συνδεδεμένος στον λογαριασμό μέλους του και ποιος όχι.

Για την ιστορία, αξίζει να αναφερθεί πως ο όρος «cookie» έχει τις ρίζες του στο «magic cookie» το οποίο είναι ένα πακέτο δεδομένων που λαμβάνει ένα πρόγραμμα και το ξαναστέλνει χωρίς να έχει υποστεί αλλαγές. Τα «magic cookies» χρησιμοποιήθηκαν για πρώτη φορά στις επικοινωνίες WEB το 1994 από τον προγραμματιστή Lou Montulli.[1.5]



(Στην φωτογραφία βλέπουμε ένα παράδειγμα χρήστη **authentication cookie** που διατηρεί το **session id** μετά την επιτυχημένη σύνδεση του χρήστη στον λογαριασμό του. Σε κάθε κλήση προς τον **server**, ο **client** συμπεριλαμβάνει το **cookie** αυτό, ώστε ο διακομιστής να είναι σε θέση να γνωρίζει αν ο χρήστης έχει συνδεθεί στον λογαριασμό του.)

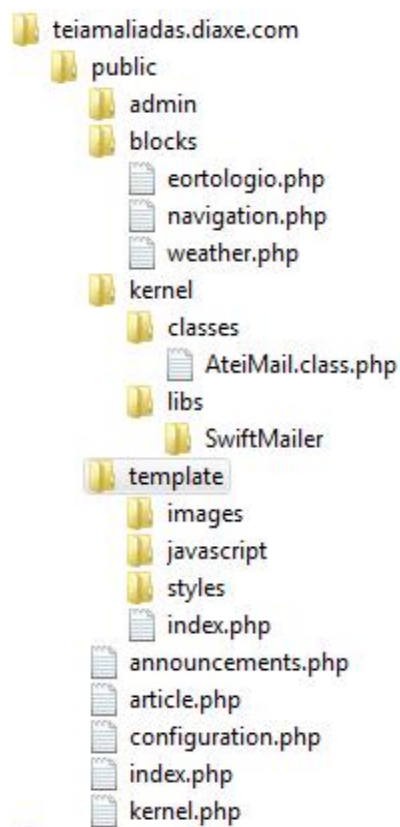
## 8) RSS / Really Simple Syndication

Το ακρωνύμιο **RSS**, από τον αγγλικό όρο Really Simple Syndication (Πολύ Απλή Διανομή), αναφέρεται σε μία προτυποποιημένη μέθοδο ανταλλαγής ψηφιακού πληροφοριακού περιεχομένου διαμέσου του Διαδικτύου, στηριγμένη στην πρότυπη, καθιερωμένη και ευρέως υποστηριζόμενη γλώσσα σήμανσης XML.[1.15] Ένας χρήστης του Διαδικτύου μπορεί έτσι να ενημερώνεται αυτομάτως για γεγονότα και νέα από όσες ιστοσελίδες υποστηρίζουν RSS, αρκεί να έχει εγγραφεί ο ίδιος συνδρομητής στην αντίστοιχη υπηρεσία της εκάστοτε ιστοσελίδας. Οι εν λόγω ενημερώσεις («ροές RSS», αγγλ: «RSS feeds») περιέχουν τα πλήρη δεδομένα, σύνοψη των δεδομένων, σχετικά μεταδεδομένα, ημερομηνία έκδοσης κλπ, ενώ αποστέλλονται αυτομάτως στον συνδρομητή μέσω Διαδικτύου. Συνήθως η διαδικασία απαιτεί ένα ειδικό λογισμικό «αναγνώστη» (το οποίο μπορεί να είναι ενσωματωμένο στον πλοηγό Web του χρήστη), όπου κάποιος πληκτρολογεί τη διεύθυνση URL της ζητούμενης ροής μιας ιστοσελίδας, ώστε να εγγραφεί σε αυτήν ως συνδρομητής. Ο αναγνώστης ελέγχει τακτικά για ενημερώσεις περιεχομένου τις καταγεγραμμένες ροές, από όλες τις διαφορετικές πηγές στις οποίες είναι εγγεγραμμένος ο συνδρομητής, αναλαμβάνοντας από μόνος του να μεταφέρει τις εν λόγω ενημερώσεις όποτε χρειάζεται και να τις συγκεντρώνει στον υπολογιστή του χρήστη. Ο αναγνώστης παρέχει στον εκάστοτε χρήστη μία κοινή και εξατομικευμένη διασύνδεση για όλες τις ροές που ο ίδιος έχει επιλέξει να παρακολουθεί. Το πρότυπο RSS υπάρχει από το 1999, ωστόσο παλαιότερες, όχι τόσο επιτυχημένες τεχνολογίες παρόμοιου σκοπού (οι οποίες βέβαια δεν αξιοποιούσαν την καθιερωμένη σήμερα γλώσσα XML, καθώς αυτή δεν είχε εμφανιστεί πριν το 1998) κυκλοφορούσαν από τα μέσα της δεκαετίας του 1990. Περί το 2005, το RSS άρχισε να υποστηρίζεται ευρέως από τους δημοφιλέστερους πλοηγούς Web και γρήγορα καθιερώθηκε, ιδιαιτέρως λόγω της εξάπλωσης κατά την ίδια περίοδο των προσωπικών ιστολογίων με τακτική ανανέωση περιεχομένου. [1.15]Οι ομάδες κατασκευής των εν λόγω πλοηγών είναι που επέλεξαν τότε ένα λογότυπο για την τεχνολογία RSS (το σημερινό ). Πολύ γρήγορα έκανε την εμφάνισή του και το ανταγωνιστικό ως προς το RSS, αλλά επίσης στηριγμένο στην XML, πρότυπο Atom.

### 3.3 ΒΗΜΑΤΑ ΑΝΑΛΥΣΗΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗΣ PORTAL

Αφού έγινε αναφορά στις βασικές τεχνολογίες που θα χρησιμοποιήσουμε στο Portal του ΑΤΕΙ Αμαλιάδας, στο κεφάλαιο αυτό, θα αναλύσουμε επισταμένως την δομή του, τόσο του πυρήνα του όσο και των επιμέρους σελίδων του.

#### 3.3.1 ΔΟΜΗ ΑΡΧΕΙΩΝ ΣΥΣΤΗΜΑΤΟΣ PORTAL



Όλα τα αρχεία του portal, βρίσκονται στον δημόσιο φάκελο του διακομιστή, στον φάκελο δηλαδή που είναι προσβάσιμος από το web. Μέσα στον δημόσιο φάκελο υπάρχουν τόσο τα αρχεία των βασικών υπηρεσιών του portal, όπως για παράδειγμα:

**index.php:** η κεντρική σελίδα, **announcements.php:** οι ανακοινώσεις, **article.php:** τα άρθρα

καθώς επίσης και βασικά αρχεία του συστήματος:

**kernel.php:** το αρχείο – κορμός του portal, το οποίο το κάνουμε require σε κάθε άλλο αρχείο υπηρεσιών.

**configuration.php:** αρχείο ρυθμίσεων του portal

Καθώς και άλλα τα οποία μαζί με τα προαναφερόμενα, αναφέρονται και αναλύονται οι λειτουργίες τους στην

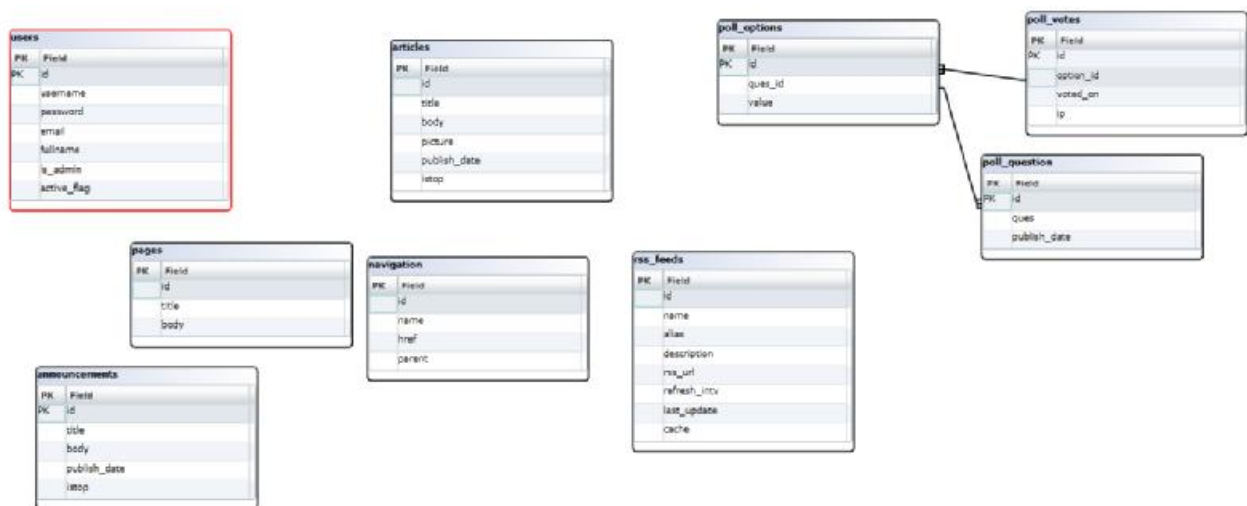
συνέχεια της εργασίας. Επίσης υπάρχουν υποφάκελοι που χωρίζουν τα συστατικά στοιχεία του portal σε ομάδες. Στον φάκελο **admin** υπάρχουν όλα εκείνα τα αρχεία που συνθέτουν τον πίνακα διαχείρισης του portal. Στον υποφάκελο **blocks** υπάρχουν όλα τα τμήματα εκείνα που ενθέτονται συμπληρωματικά στο portal, όπως είναι ο καιρός, το εορτολόγιο και άλλα. Στον υποφάκελο **kernel** περιλαμβάνονται τόσο οι βασικές κλάσεις όσο και οι βιβλιοθήκες που χρησιμοποιούνται από το portal ενώ στον φάκελο **template** βρίσκονται όλα τα αρχεία που ορίζουν την τελική εμφάνιση του portal, την θέση των blocks, τα χρώματα, τις εικόνες και γενικά ό,τι εκτελείτε στον φυλλομετρητή του χρήστη.

### 3.3.2 ΔΟΜΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΣΥΣΤΗΜΑΤΟΣ PORTAL

Για να μπορέσει να λειτουργήσει οποιοδήποτε δυναμικό site όπως είναι το portal του ΑΤΕΙ Αμαλιάδας, απαραίτητη προϋπόθεση είναι να υπάρχει βάση δεδομένων που διαθέτει και το τροφοδοτεί με όλες εκείνες τις πληροφορίες που χρειάζεται να εμφανίσει στους επισκέπτες του. Δεν έχει σημασία ποιάς μορφής βάση δεδομένων χρησιμοποιεί, από flat file (πληροφορίες σε αρχεία), NoSQL (μη σχεσιακός τύπος βάσης) ως τις ακριβότερες σε κόστος βάσεις όπως είναι η σχεσιακή βάση δεδομένων της Oracle. Το portal του ΑΤΕΙ Αμαλιάδας, χρησιμοποιεί το πιο διαδεδομένο αυτή τη στιγμή λογισμικό σχεσιακής βάσης δεδομένων, το λογισμικό ανοικτού κώδικα MySQL όπως αναφέρθηκε στο προηγούμενο κεφάλαιο. [1.12]

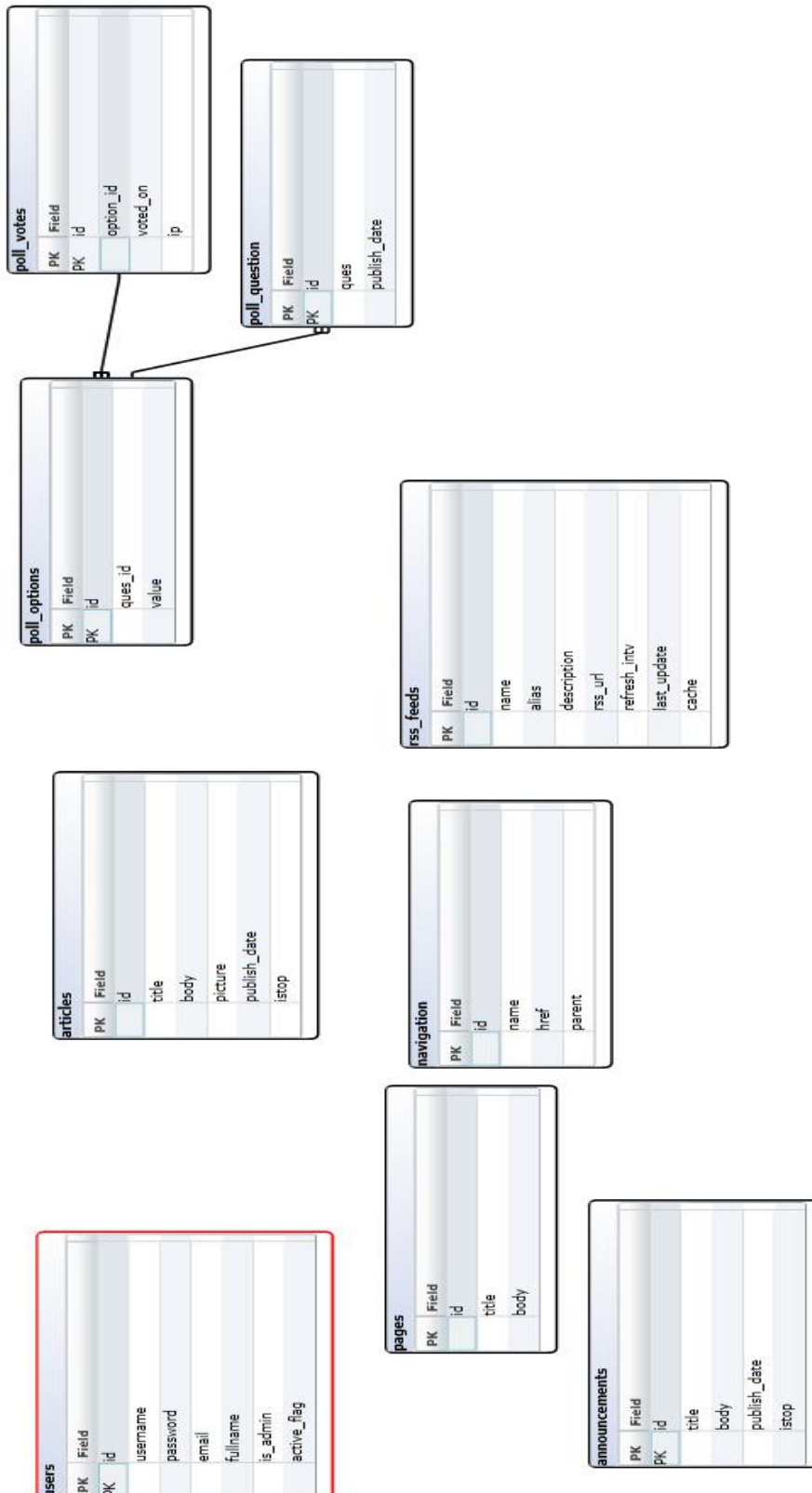


Ακολουθεί ένα **DATABASE DIAGRAM** το οποίο αναπαριστά σχηματικά την βάση δεδομένων του συστήματος, καθώς και τις σχέσεις μεταξύ των πινάκων της ψηφοφορίας (poll) μιας και με βάση την δομή του portal, είναι οι μόνιμοι που συσχετίζονται.



(Στην επόμενη σελίδα, το πάνω διάγραμμα βρίσκεται σε μεγέθυνση)

## DATABASE DIAGRAM



Η βάση δεδομένων αποτελείται από τους εξής πίνακες:

### 1) Πίνακας “articles”:

Ο πίνακας στον οποίο αποθηκεύονται όλα τα άρθρα είναι ο πίνακας **articles**, ο οποίος αποτελείται από τις εξής στήλες:

- **id:** Ο αύξων αριθμός των άρθρων. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέου άρθρου εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **title:** Στήλη αλφαριθμητικών δεδομένων που περιλαμβάνει τον τίτλο κάθε άρθρου με μέγιστο μήκος τους 160 χαρακτήρες.
- **body:** Στήλη τύπου text που περιλαμβάνει το κείμενο του άρθρου.
- **picture:** Στήλη τύπου blob (binary large object) το οποίο περιλαμβάνει τα δυαδικά δεδομένα του αρχείου της συνοδευτικής φωτογραφίας του άρθρου.
- **publish\_date:** Στήλη τύπου datetime η οποία περιλαμβάνει την ημερομηνία δημοσίευσης του άρθρου. Ο ρόλος της στήλης αυτής ξεπερνά τον ρόλο απλώς του «αρχείου» που μας δίνει την δυνατότητα να γνωρίζουμε πότε δημοσιεύτηκε τι, αλλά παίζει ενεργό ρόλο στην εμφάνιση των άρθρων στο portal, αφού η ταξινόμηση γίνεται βάση αυτού του πεδίου σε όλα τα άρθρα που έχουν ημερομηνία δημοσίευσης μικρότερη ή ίση με την ημερομηνία και ώρα επίσκεψης του χρήστη στο portal.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 <b>id</b>	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>title</b>	varchar(160)	utf8_general_ci		No	None	
<input type="checkbox"/>	3 <b>body</b>	text	utf8_general_ci		No	None	
<input type="checkbox"/>	4 <b>picture</b>	blob			No	None	
<input type="checkbox"/>	5 <b>publish_date</b>	datetime			No	None	

(οι στήλες του πίνακα articles)

### 2) Πίνακας “announcements”:

Ο πίνακας στον οποίο αποθηκεύονται όλες οι ανακοινώσεις του ΑΤΕΙ Αμαλιάδας είναι ο **announcements** ο οποίος αποτελείται από τις ακόλουθες στήλες:

- **id:** Ο αύξων αριθμός των ανακοινώσεων. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέας ανακοίνωσης εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο

οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.

- **title:** Στήλη αλφαριθμητικών δεδομένων που περιλαμβάνει τον τίτλο κάθε ανακοίνωσης με μέγιστο μήκος τους 160 χαρακτήρες.
- **body:** Στήλη τύπου text που περιλαμβάνει το κείμενο του ανακοίνωσης.
- **publish\_date:** Στήλη τύπου datetime η οποία περιλαμβάνει την ημερομηνία δημοσίευσης της ανακοίνωσης και χρησιμοποιείται ομοίως με την αντίστοιχη στήλη του πίνακα articles για ανάκτηση και ταξινόμηση.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>title</b>	varchar(160)	utf8_general_ci		No	None	
<input type="checkbox"/> 3	<b>body</b>	text	utf8_general_ci		No	None	
<input type="checkbox"/> 4	<b>publish_date</b>	datetime			No	None	

(οι στήλες του πίνακα announcements)

3) Πίνακας “**users**”: Όπως προκύπτει και από την ονομασία του, ο πίνακας αυτός φιλοξενεί όλους τους λογαριασμούς χρηστών στο portal. Αν κάποιος ή κάποιοι από αυτούς τους χρήστες είναι διαχειριστής/ές τότε η τιμή της στήλης is\_admin από 0 γίνεται 1 και ο συγκεκριμένος χρήστης έχει πρόσβαση στον πίνακα διαχείρισης του portal. Οι στήλες του πίνακα αυτού είναι οι:

- **id:** Ο αύξων αριθμός των χρηστών. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέου χρήστη εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **username:** Αλφαριθμητική στήλη που περιλαμβάνει τα usernames των εγγεγραμμένων χρηστών του portal, με μέγιστο μήκος τους 50 χαρακτήρες.
- **password:** Αλφαριθμητική στήλη που περιλαμβάνει τους κωδικούς πρόσβασης των εγγεγραμμένων χρηστών του portal, με μέγιστο μήκος τους 50 χαρακτήρες.
- **email:** Αλφαριθμητική στήλη μέγιστου μήκους και εδώ 50 χαρακτήρων που περιλαμβάνει τα e-mails των χρηστών.
- **fullname:** Αλφαριθμητική στήλη μέγιστου μήκους 50 χαρακτήρων που περιλαμβάνει το πλήρες όνομα των χρηστών.



- **is\_admin:** Στήλη που επιτελεί τον ρόλο σημαίας στο portal, τύπου tinyint το οποίο λαμβάνει την τιμή **0** όταν ο χρήστης είναι απλός και την τιμή **1** όταν ο χρήστης είναι **διαχειριστής του portal**
- **active\_flag:** Ακόμα μια στήλη που επιτελεί τον ρόλο σημαίας στο portal, επίσης τύπου tinyint η οποία λαμβάνει την τιμή 0 όταν ο λογαριασμός του χρήστη είναι ανενεργός και την τιμή 1 όταν ο λογαριασμός είναι ενεργός.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
2	<b>username</b>	varchar(50)	utf8_general_ci		No	None	
3	<b>password</b>	varchar(50)	utf8_general_ci		No	None	
4	<b>email</b>	varchar(50)	utf8_general_ci		No	None	
5	<b>fullname</b>	varchar(50)	utf8_general_ci		No	None	
6	<b>is_admin</b>	tinyint(4)			No	0	
7	<b>active_flag</b>	tinyint(4)			No	0	

( οι στήλες του πίνακα users )

#### 4) Πίνακας “navigation”:

Ο πίνακας **navigation** περιέχει όλες τις εγγραφές που συνθέτουν το μενού πλοήγησης του portal. Περιέχει τις ακόλουθες στήλες:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέου άρθρου εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **name:** Αλφαριθμητική στήλη μήκους 50 χαρακτήρων που περιλαμβάνει το εμφανιζόμενο όνομα τις κάθε επιλογής πλοήγησης.
- **href:** Αλφαριθμητική στήλη μήκους 160 χαρακτήρων που περιλαμβάνει τον σύνδεσμο προς κάθε περιεχόμενο ή υπηρεσία του portal.
- **parent:** Στήλη τύπου int που χρησιμοποιείται για την συσχέτιση συγκεκριμένων αντικειμένων πλοήγησης με άλλα, πατρικά αντικείμενα του ίδιου πίνακα. Λαμβάνει είτε την τιμή id κάποιας άλλης εγγραφής, κάτι που θα υποδηλώσει πως η παρούσα είναι «παιδί» της, ή την τιμή 0 που υποδηλώνει πως η συγκεκριμένη εγγραφή είναι πρώτη στην ιεραρχία.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
2	<b>name</b>	varchar(50)	utf8_general_ci		No	None	
3	<b>href</b>	varchar(160)	utf8_general_ci		No	None	
4	<b>parent</b>	int(11)			No	None	

( οι στήλες πίνακα navigation )

5) Πίνακας “**eortologio**”: Στον πίνακα αυτό, βρίσκονται προ-αποθηκευμένες οι ονομαστικές γιορτές, τις οποίες εμφανίζουμε στο postal σε συγκεκριμένο block. Αποτελείτε από τις ακόλουθες στήλες:

- **day**: Περιλαμβάνει την ημέρα (1 ως 31) του μήνα στην οποία υπάρχει η εκάστοτε γιορτή και είναι τύπου Int(2).
- **month**: Περιλαμβάνει τον μήνα (1 ως 12) στον οποίο την συγκεκριμένη ημέρα υπάρχει η εκάστοτε γιορτή.
- **text**: Περιλαμβάνει τις ονομαστικές γιορτές.

Εκτός αναθεώρησης, στον πίνακα αυτό δεν προβλέπονται αλλαγές τόσο από χρήστες όσο από διαχειριστές, το μόνο που χρειαζόμαστε να κάνουμε, είναι να προβούμε σε αρχικοποίηση του πίνακα, προσθέτοντάς του όλες τις εγγραφές των γιορτών ως εξής:

	day	month	text
delete	7	1	Ιωάννης, Ιωάννα, Ιώ, Πρόδρομος
delete	1	1	Βασίλης, Βασιλική

Για λεπτομέρειες των εγγραφών της αρχικοποίησης ανατρέξτε στο **παράρτημα 1.1**.

**ATEI Αμαλιάδας**

Ημερομηνία: 16/09/2012

Σήμερα γιορτάζουν: Ευθηνία, Έφη

6) Πίνακας “**rss\_feeds**”:

Ο πίνακας **rss\_feeds** αποτελεί έναν από τους σημαντικότερους πίνακες του portal, αφού συγκεντρώνει πληροφορίες από τρίτες πηγές (πχ eclass), μέσω των κατάλληλων διεργασιών τις αποθηκεύει και τις επιστρέφει στον επισκέπτη του portal. Αποτελείται από τις ακόλουθες στήλες:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέου feed εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **name:** Αλφαριθμητική στήλη μέγιστου μήκους 80 χαρακτήρων που περιέχει το όνομα του rss feed το οποίο προβάλλεται στον επισκέπτη.
- **alias:** Αλφαριθμητική στήλη μέγιστου μήκους 50 χαρακτήρων που περιέχει το εσωτερικό όνομα του rss feed, για χρήση από τον κώδικα του portal. Αν επιθυμούμε δηλαδή να καλέσουμε προς προβολή σε ένα συγκεκριμένο σημείο του portal ένα rss feed, θα το καλέσουμε βάση του alias του, για αυτό το λόγο αυτό το πεδίο χρήζεται ως index (**ALTER TABLE `rss\_feeds` ADD INDEX ( `alias` )**).
- **rss\_url:** Αλφαριθμητική στήλη μέγιστου μήκους 255 χαρακτήρων που περιλαμβάνει διευθύνσεις url του εκάστοτε rss feed.
- **refresh\_intv:** Αριθμητική στήλη τύπου int που περιλαμβάνει αριθμό δευτερολέπτων που το σύστημα θα ανανεώνει τα περιεχόμενα του rss feed στην τοπική προσωρινή μνήμη (cache).
- **last\_update:** Στήλη τύπου datetime που περιλαμβάνει την ημερομηνία της τελευταίας ενημέρωσης του πεδίου cache (που ακολουθεί).
- **cache:** Στήλη τύπου text που περιλαμβάνει το τελευταίο στιγμιότυπο όπως ελήφθη από τα επιστρεφόμενα δεδομένα του rss\_url και ενημερώνεται όταν ο χρόνος τη στιγμή επίσκεψης του χρήστη είναι μεγαλύτερος του χρόνου last\_update συν τα δευτερόλεπτα του refresh\_intv.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	name	varchar(80)	utf8_general_ci		No	None	
3	alias	varchar(50)	utf8_general_ci		No	None	
4	rss_url	varchar(255)	utf8_general_ci		No	None	
5	refresh_intv	int(11)			No	None	
6	last_update	datetime			No	None	
7	cache	text	utf8_general_ci		No	None	

( οι στήλες του πίνακα rss\_feeds )

## 7) Πίνακας “pages”:

Ο πίνακας pages διατηρεί αποθηκευμένες όλες τις υποσελίδες του portal του ΑΤΕΙ Αμαλιάδας, και αποτελείται από τα ακόλουθα πεδία:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέας σελίδας εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **title:** Η στήλη αυτή, περιέχει τον τίτλο της σελίδας, κατά συνέπεια ο τύπος του πεδίου είναι αλφαριθμητικός και έχει μέγιστο μήκος 160 χαρακτήρες.
- **body:** Η στήλη αυτή περιέχει το πλήρες κείμενο της σελίδας, απροσδιορίστου μεγέθους, για αυτό κατά συνέπεια ορίζουμε τον τύπο ως text.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<b>id</b>	int(11)			No	None	AUTO_INCREMENT
2	<b>title</b>	varchar(160)	utf8_general_ci		No	None	
3	<b>body</b>	text	utf8_general_ci		No	None	

( οι στήλες του πίνακα pages )

#### 8) Πίνακας “poll\_questions”:

Ο πίνακας αυτός διατηρεί τις ερωτήσεις που τίθενται προς δημοσκόπηση. Αποτελείται από τα ακόλουθα πεδία:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέας ερώτησης εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **ques:** Το αλφαριθμητικό αυτό πεδίο μήκους 250 χαρακτήρων περιλαμβάνει την ερώτηση που τίθεται προς ψηφοφορία.
- **publish\_date:** Το πεδίο datetime αυτό περιλαμβάνει την ημερομηνία δημοσίευσης.

Name	Type	Length
id	int	11
ques	varchar	250
publish_date	datetime	0

9) Πίνακας “**poll\_options**”:

Ο πίνακας αυτός διατηρεί τις διαθέσιμες επιλογές που τίθενται προς ψηφοφορία για κάθε ερώτηση. Αποτελείται από τα ακόλουθα πεδία:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέας επιλογής εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **ques\_id:** Το πεδίο αυτό τύπου integer συσχετίζει την παρούσα επιλογή με ερώτηση του πίνακα “**poll\_questions**”.
- **value:** Περιλαμβάνει την επιλογή που τίθεται σε ψηφοφορία.

Name	Type	Length
id	int	11
ques_id	int	11
value	varchar	300

10) Πίνακας “**poll\_votes**”:

Ο πίνακας αυτός διατηρεί τις καταχωρημένες ψήφους. Αποτελείται από τα ακόλουθα πεδία:

- **id:** Ο αύξων αριθμός των στοιχείων πλοήγησης. Κατά την εισαγωγή κάθε νέας εγγραφής (κάθε νέας ψήφου εν προκειμένω) η βάση αποδίδει αυτόματα αριθμό id ο οποίος αποτελείται από τον αριθμό της προηγούμενης εγγραφής, προσαυξημένο κατά 1, αφού στην στήλη id έχουμε αποδώσει το χαρακτηριστικό **AUTO\_INCREMENT**.
- **option\_id:** Συσχετίζει την συγκεκριμένη εγγραφή ψήφου με την επιλογή του πίνακα “**poll\_option**” που ψηφίστηκε.
- **voted\_on:** Το πεδίο datetime αυτό περιλαμβάνει την ημερομηνία καταχώρησης της ψήφου.
- **ip:** Περιλαμβάνει την IP του χρήστη που ψήφισε.

Name	Type	Length
id	int	11
ques	varchar	250
publish_date	datetime	0

### 3.3.3 ΔΟΜΗ ΠΥΡΗΝΑ ΣΥΣΤΗΜΑΤΟΣ PORTAL

Ο πυρήνας του συστήματος, αποτελείται από όλο το πλέγμα αρχείων που περιλαμβάνουν είτε τις διεργασίες είτε τις βιβλιοθήκες που χρησιμοποιούνται κατά την λειτουργία του portal. Για να γίνει κατανοητό αυτό, ένα παράδειγμα διεργασίας θεωρούμε την διαδικασία που ελέγχει εάν ο επισκέπτης του portal είναι συνδεδεμένος στον λογαριασμό του, μέσω της αναζήτησης του authentication cookie ενώ ένα παράδειγμα βιβλιοθήκης είναι η βιβλιοθήκη Swift Mailer για την PHP την οποία και χρησιμοποιούμε εμφωλευμένη στην δική μας κλάση/διεργασία αποστολής e-mails, την AteiMail.

Το βασικό αρχείο του πυρήνα, ονομάζεται **kernel.php** και βρίσκεται όπως αναφέρθηκε στον βασικό κατάλογο αρχείων του portal. Επιτελεί ρόλο «διαιτητή» ή «τροχονόμου», καθώς αυτό είναι υπεύθυνο για το φόρτωμα και την εκτέλεση όλων των αναγκαίων κλάσεων, βιβλιοθηκών ή συνδέσεων (π.χ. με την βάση δεδομένων), ανάλογα με την κάθε λειτουργία που είναι αναγκαία στην εκάστοτε σελίδα που προβάλλεται στον επισκέπτη. Έως τώρα, συνηθιζόταν σε πολλά projects να κάνουμε include κάθε κλάση και κάθε βιβλιοθήκη που έχουμε δημιουργήσει ανεξαρτήτως αν την δεδομένη στιγμή για μια συγκεκριμένη λειτουργία την χρειαζόμαστε ή όχι. Αυτό επιφέρει πρόσθετο κόστος σε πόρους, καθώς αναγκάζουμε τον διακομιστή να συμπεριλάβει στην εκτέλεση του κώδικα, αρχεία που εν τέλει δεν θα τα χρησιμοποιήσει. Για να γλιτώσουμε αυτό το κόστος, εκμεταλλευόμενοι τις δυνατότητες που προστέθηκαν στην PHP 5.3 όπως είναι οι anonymous functions, έχουμε δημιουργήσει το ακόλουθο μοντέλο κώδικα στο kernel.php:

```
// Prothema dk apo ta arxika Diamantis Kyriakakis
```

```
$GLOBALS['_cb_init'] = Array();
```

```
function dkSetInit($init, $cb) {
```

```
    $GLOBALS['_cb_init'][$init][] = $cb;
```

```
}
```

```
function dkInit($init) {
```

```
    if(func_num_args() > 1) $init = func_get_args();
```

```
    if(is_array($init)) foreach($init as $i) dkInit($i);
```

```
    else foreach((array)$GLOBALS['_cb_init'][$init] as $k=>$i) {
```

```
        if(is_callable($i)) $i();
```

```
        unset($GLOBALS['_cb_init'][$init][$k]);
```

```
    }
```

```
}
```

Το παραπάνω μοντέλο εξασφαλίζει την καλύτερη δυνατή χρήση πόρων, αφού κάθε φορά, περιλαμβάνουμε μόνο ότι ακριβώς χρειαζόμαστε στην εκτέλεση του κώδικά μας.

Αρχικώς, βλέπουμε πως έχουμε δημιουργήσει 1 array, τα:

```
$GLOBALS['_cb_init'] = Array();
```

Το array, περιέχει ουσιαστικά anonymous functions που εκτελούν τις λειτουργίες που χρειαζόμαστε.

Η function dkSetInit δέχεται 2 ορίσματα: το \$init και το \$cb. Το \$init είναι το όνομα με το οποίο βαφτίζουμε την ενέργεια, ενώ το \$cb (από το callback) είναι ουσιαστικά μια anonymous function που δηλώνουμε τις ενέργειες που θέλουμε να εκτελεστούν (π.χ. σύνδεση με την βάση, include κάποιας κλάσης ή βιβλιοθήκης κτλ). Αυτό που κάνει αυτή η function είναι να εισάγει στο πρώτο array την anonymous function που του περνάμε στο \$cb, με όνομα κλειδιού το \$init.

```
function dkSetInit($init, $cb) {  
    $GLOBALS['_cb_init'][$init][] = $cb;  
}
```

Έτσι, το μόνο που έχουμε να κάνουμε στα αρχεία που χρειαζόμαστε την εν λόγω λειτουργία που έχουμε περιλάβει στο προηγούμενο \$cb, είναι να καλέσουμε την συγκεκριμένη λειτουργία (που έχουμε βαφτίσει init) καλώντας την δεύτερη function dkInit(\$init), όπου \$init το όνομα κλειδιού της λειτουργίας που θέλουμε να εκτελέσουμε είτε ένα array που περιέχει όλα τα ονόματα των λειτουργιών που θέλουμε να εκτελέσουμε:

```
function dkInit($init) {  
    if(func_num_args() > 1) $init = func_get_args();  
    if(is_array($init)) foreach($init as $i) dkInit($i);  
    else foreach((array)$GLOBALS['_cb_init'][$init] as $k=>$i) {  
        if(is_callable($i)) $i();  
        unset($GLOBALS['_cb_init'][$init][$k]);  
    }  
}
```

Η function όπως βλέπουμε ελέγχει εάν το \$init είναι συμβολοσειρά ή array και για κάθε ένα από τα ονόματα που υπάρχουν στην μεταβλητή αυτή, ελέγχει εάν το αποθηκευμένο στοιχείο (το \$cb ουσιαστικά) μπορεί να κληθεί και αν ναι το εκτελεί αφαιρώντας το παράλληλα από το array ώστε να αποφευχθεί περεταιίρω ουσιαστικά άσκοπη κλήση του.

Για να γίνει πιο κατανοητή η χρήση του στο portal, ακολουθούν μερικά παραδείγματα δήλωσης και εκτέλεσης:

**Δήλωση ενός init στο kernel.php με όνομα database το οποίο αναλαμβάνει την σύνδεση με την βάση δεδομένων με στοιχεία σύνδεσης που παίρνει από το configuration.php:**

```
// Set database init
dkSetInit('database', function() {
    $GLOBALS['DB'] = new PDO ( 'mysql:host=' . Configuration::$DB_Host . ';dbname=' .
Configuration::$DB_Name, Configuration::$DB_User, Configuration::$DB_Pass, array
(PDO::ATTR_PERSISTENT => false ) );
    $GLOBALS['DB']->setAttribute ( PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING
);
});
```

Αφού δηλώσαμε το init αυτό στο kernel.php, το οποίο κάνουμε include στα κατά τόπους αρχεία (πχ στο index.php της κεντρικής σελίδας), για να ενεργοποιήσουμε την σύνδεση με την βάση, δεν έχουμε παρά να εκτελέσουμε το:

```
dkInit('database');
```

Για να καλέσουμε μια δεύτερη δήλωση init στο kernel.php που ονομάζεται AteiMail και αναλαμβάνει το φόρτωμα της κλάσης AteiMail, δεν έχουμε παρά να περάσουμε στο dkInit σαν δεύτερη παράμετρο το όνομα AteiMail:

```
dkInit('database', 'AteiMail');
```

Πάνω στην συγκεκριμένη δομή που δημιουργήσαμε εκμεταλλευόμενοι την PHP 5.3 είναι χτισμένος ολόκληρος ο πυρήνας του portal του ΑΤΕΙ Αμαλιάδας με όλες τις απαραίτητες λειτουργίες που επιτελεί.

Βάση της παραπάνω λογικής λειτουργεί ο πυρήνας του portal.





### 3.3.4 TEMPLATE ΚΑΙ BLOCKS

Στα δυναμικά sites όπως το portal του ΑΤΕΙ Αμαλιάδας, κάποια τμήματα κώδικα, επαναχρησιμοποιούνται στις υπόλοιπες σελίδες τους. Σε αυτή την κατηγορία εντάσσεται τόσο ο κώδικας της εμφάνισης του δικτυακού τύπου, όσο και blocks κώδικα τα οποία επαναχρησιμοποιούνται. Στο portal του ΑΤΕΙ Αμαλιάδας, στον φάκελο blocks τοποθετούμε επαναχρησιμοποιήσιμα τμήματα κώδικα τα οποία τα ανακαλούμε σε όποιο σημείο της εκάστοτε σελίδας ή στον κώδικα του template, καλώντας την μέθοδο loadBlock(**όνομα\_block**); η οποία φορτώνει το αρχείο **όνομα\_block.php** από τον φάκελο blocks.

Το template ουσιαστικά αποτελεί το «σκαρί» της εμφάνισης του portal και φορτώνεται τελευταίο, αφού η εκάστοτε σελίδα που προβάλλεται έχει ήδη συντεθεί. Για αυτό το σκοπό, εκμεταλλευόμαστε την μέθοδο **register\_shutdown\_function** της PHP η οποία ως όρισμα δέχεται ανώνυμη (ή lambda) μέθοδο την οποία και εκτελεί αφού έχει εκτελεστεί ολόκληρος ο κώδικας του portal. Στην ανώνυμη αυτή μέθοδο, περιλαμβάνουμε όλες τις διεργασίες φόρτωσης του template, από τον φάκελο template, και συγκεκριμένα το αρχείο index.php. Στο σημείο του template που θέλουμε να εμφανίζεται κάθε σελίδα του portal, εμφανίζουμε τα περιεχόμενα της μεταβλητής \$MAIN\_CONTENT που σε αυτό το στάδιο περιλαμβάνει το αποτέλεσμα της εκτέλεσης του προηγούμενου κώδικα προ της εκτέλεσης της **register\_shutdown\_function**. **Να σημειώσουμε εδώ πως το αποτέλεσμα της εκτέλεσης του προηγούμενου κώδικα το παίρνουμε εκτελώντας πρώτα : \$MAIN\_CONTENT = ob\_get\_contents(); το οποίο αντιγράφει τα περιεχόμενα αυτά από τον output buffer της php στην μεταβλητή και κατόπιν εκτελούμε την ob\_end\_clean(); για να καθαρίσουμε τον buffer αφού σε κάθε περίπτωση θα το εμφανίσουμε μέσα στο template τυπώνοντας την \$MAIN\_CONTENT.**

Το αποτέλεσμα εκτέλεσης

```
<div class="main"><?=$MAIN_CONTENT;?></div>
```

### 3.3.5 ΓΕΓΟΝΟΤΑ ΠΟΥ ΛΑΜΒΑΝΟΥΝ ΧΩΡΑ ΚΑΤΑ ΤΗΝ ΕΙΣΟΔΟ ΤΟΥ ΕΠΙΣΚΕΠΤΗ ΣΤΟ PORTAL

Όταν ο επισκέπτης επισκεφθεί το portal του ΑΤΕΙ Αμαλιάδας και ζητήσει την προβολή κάποιας σελίδας (της κεντρικής ή άλλης), λαμβάνουν χώρα τα ακόλουθα γεγονότα:

- 1) Ο browser (client) ζητά συγκεκριμένη σελίδα και στέλνει της πληροφορίες που είναι αποθηκευμένες στα cookies στο σύστημα διαχείρισης περιεχομένου του Portal.
- 2) Το σύστημα διαχείρισης περιεχομένου αναζητά την ζητούμενη σελίδα και αν υπάρχει συνθέτει όλες τις πληροφορίες που την συντελούν. Σε περίπτωση που ο χρήστης κατά την επίσκεψή του δεν ζητήσει κάποια συγκεκριμένη σελίδα (κάτι σύνηθες κατά την πρώτη επίσκεψη), το σύστημα αυτομάτως του επιστρέφει την κεντρική σελίδα.
- 3) Το υποσύστημα πιστοποίησης ελέγχει τις πληροφορίες που ελήφθησαν από τα cookies και αφορούν την ενδεχόμενη ύπαρξη συνεδρίας χρήστη (=ο χρήστης έχει εκτελέσει είσοδο στο σύστημα με τα προσωπικά του στοιχεία), τότε αντλεί από την βάση τα στοιχεία του προφίλ του και το χρησιμοποιεί. Σε περίπτωση που αυτός ο έλεγχος αποτύχει, αυτό σημαίνει πως ο επισκέπτης είτε δεν είναι εγγεγραμμένος χρήστης είτε δεν έχει συνδεθεί στον προσωπικό του λογαριασμό. Στην περίπτωση αυτή το σύστημα του εμφανίζει την φόρμα εισόδου και έναν σύνδεσμο προς την σελίδα εγγραφής σε περίπτωση που δεν έχει εγγραφεί και το επιθυμεί.
- 4) Εάν η ζητούμενη σελίδα δεν βρεθεί, είτε επειδή έχει αφαιρεθεί είτε επειδή δεν υπήρξε ποτέ και το λάθος οφείλεται στον επισκέπτη, τότε το σύστημα επιστρέφει error 404, το οποίο αναφέρεται στο ότι το περιεχόμενο που ζητήθηκε δεν υπάρχει.
- 5) Αν ο χρήστης έχει δικαίωμα πρόσβασης σε πληροφορίες η λειτουργίες που αφορούν αποκλειστικά εγγεγραμμένους χρήστες, του επιτρέπουμε την πρόσβαση σε αυτές.
- 6) Το Template που ορίζει την εμφάνιση του Portal φορτώνει και είναι έτοιμο να δεχθεί το αποτέλεσμα της σύνθεσης των πληροφοριών του συστήματος διαχείρισης περιεχομένου.
- 7) Αναλόγως την ζητούμενη σελίδα, το σύστημα εκτελεί τη ζητούμενη λειτουργία, όπως για παράδειγμα το φόρτωμα πληροφοριών από την βάση δεδομένων MySQL.
- 8) Όλες οι πληροφορίες της ζητούμενης σελίδας ενσωματώνονται στο template και δημιουργούν ένα ενιαίο αποτέλεσμα σε γλώσσα HTML.
- 9) Ο Διακομιστής επιστρέφει το παραγόμενο HTML στον Browser του χρήστη ο οποίος με την σειρά του το συνθέτει στην οθόνη του.

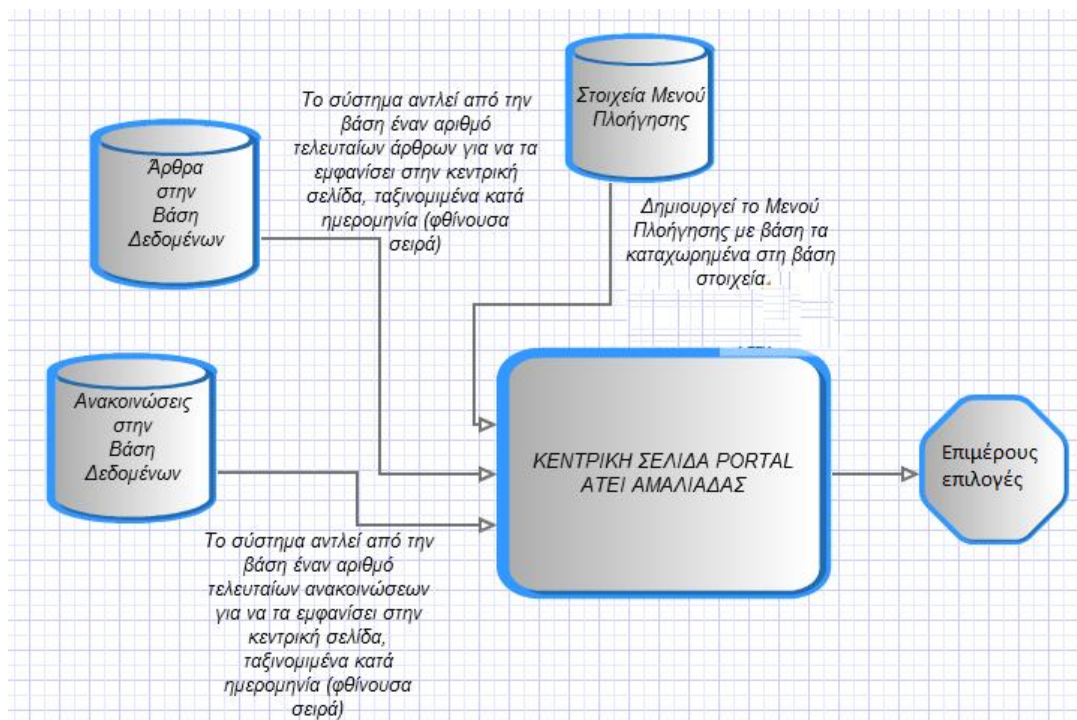
### 3.3.6 ΔΟΜΗ ΚΕΝΤΡΙΚΗΣ ΣΕΛΙΔΑΣ

Η Κεντρική σελίδα του Portal επιτελεί τον ρόλο της συγκέντρωσης και τις εμφάνισης όλων των τελευταίων πληροφοριών που έχουν εισαχθεί στο σύστημα διαχείρισης περιεχομένου καθώς και τον ρόλο της παροχής όλων των συνδέσμων που κατευθύνουν σε λοιπές πληροφορίες που δεν εμφανίζονται στην κεντρική σελίδα.

Στην κεντρική σελίδα του **portal** του ΑΤΕΙ ΑΜΑΛΙΑΔΑΣ, εμφανίζονται:

- Αναλυτικά το τελευταίο άρθρο που εισήχθη ανεξαρτήτως κατηγορίας μαζί με την φωτογραφία του.
- Το προτελευταίο άρθρο που εισήχθη ανεξαρτήτως κατηγορίας, λιγότερο αναλυτικά και με την φωτογραφία του σε μικρότερο μέγεθος.
- Τίτλος και φωτογραφία των τελευταίων τριών άρθρων μετά τα δυο προηγούμενα
- Τίτλοι άρθρων όλων των υπολοίπων κατηγοριών
- Οι τελευταίες ανακοινώσεις
- Η τελευταία ενεργή ψηφοφορία
- Εορτολόγιο
- Καιρός
- Τελευταίες ανακοινώσεις e-class μέσω rss
- Πλαίσιο εισόδου χρηστών ή τα στοιχεία του χρήστη αν είναι ήδη συνδεδεμένος.
- Επιλογές πλοήγησης (Μενού)

Ο έλεγχος των στοιχείων τα οποία συντελούν στην σύνθεση της αρχικής σελίδας (όπως ημερομηνία και ώρα) γίνεται την στιγμή εισόδου του επισκέπτη σε αυτή.



Παράδειγμα άντλησης πληροφοριών από την βάση για την εμφάνιση άρθρων, ανακοινώσεων και πλοήγησης

### 3.3.7 ΔΟΜΗ ΣΕΛΙΔΑΣ article.php

Η σελίδα article.php επιτελεί τον ρόλο της εμφάνισης άρθρων από την βάση δεδομένων, λαμβάνοντας ως όρισμα την μεταβλητή **id** μέσω GET η οποία περιέχει έναν έγκυρο αριθμό **id** άρθρου, υπαρκτό στον πίνακα articles της βάσης δεδομένων. Σε περίπτωση που το δοθέν **id** δεν ανταποκρίνεται σε εγγραφή στη βάση, είτε επειδή κάποιος επισκέπτης το άλλαξε χειροκίνητα από την γραμμή διευθύνσεων του φυλλομετρητή είτε έρχεται από μηχανή αναζήτησης η οποία έχει καταχωρήσει άρθρο που δεν υφίσταται πια στην βάση δεδομένων του portal, τότε η σελίδα αυτή εμφανίζει μήνυμα λάθους περί μη ύπαρξης του άρθρου, αλλιώς το εμφανίζει μαζί με την φωτογραφία του. Η δομή είναι που ακολουθεί η σελίδα articles.php παρουσιάζεται στην εικόνα, αριστερά.

#### ΤΙΤΛΟΣ ΑΡΘΡΟΥ



ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ  
 ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ  
 ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ  
 ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ  
 ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ

Σε περίπτωση όμως που δεν δοθεί η παράμετρος id, τότε η σελίδα θα εμφανίσει όλα τα τελευταία άρθρα στην βάση με φθίνουσα σειρά με βάση την ημερομηνία επίσκεψης στην συγκεκριμένη σελίδα και συγκεκριμένα όσα άρθρα έχουν ημερομηνία δημοσίευσης μικρότερη από την ημερομηνία

και ώρα επίσκεψης. Στην περίπτωση αυτή η σελίδα θα ακολουθήσει την δομή που παρουσιάζεται στην φωτογραφία που ακολουθεί:

<b>Φωτογρα- φία</b>	<b>ΤΙΤΛΟΣ ΑΡΘΡΟΥ</b>
	20/10/2011 - 20:21
	ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ
	<a href="#">ΠΕΡΙΣΣΟΤΕΡΑ...</a>

<b>Φωτογρα- φία</b>	<b>ΤΙΤΛΟΣ ΑΡΘΡΟΥ</b>
	20/10/2011 - 20:21
	ΚΕΙΜΕΝΟ ΚΕΙΜΕΝΟ

### 3.3.8 ΔΟΜΗ ΣΕΛΙΔΑΣ announcements.php

Η σελίδα announcements.php επιτελεί ανάλογο ρόλο με την articles.php, παρουσιάζοντας την ανακοίνωση της οποίας το id εγγραφής στη βάση δεδομένων, δοθεί στην παράμετρο GET με όνομα id. Ομοίως και σε αυτή τη σελίδα, εάν δοθεί κάποιο id που δεν ανταποκρίνεται σε εγγραφή, θα εμφανιστεί σχετικό μήνυμα λάθους, διαφορετικά αν δεν δηλωθεί, θα εμφανιστούν οι τελευταίες εγγραφές που είναι αποθηκευμένες στον πίνακα announcements ταξινομημένες με φθίνουσα σειρά, με βάση την ημερομηνία των εγγραφών που προκύπτουν και έχουν ημερομηνία δημοσίευσης μικρότερη ή ίση με αυτή της στιγμής επίσκεψης του χρήστη.

Η δομή της σελίδας είναι της μορφής που ακολουθεί. Σε περίπτωση που δοθεί σαν παράμετρος, id με αριθμό ανακοίνωσης, εμφανίζεται μόνο ένα πλαίσιο με την ανακοίνωση, διαφορετικά εμφανίζονται οι ανακοινώσεις με την διαδικασία που προαναφέρθηκε, κάθε μια μέσα σε δικό της πλαίσιο, όπως στην εικόνα:

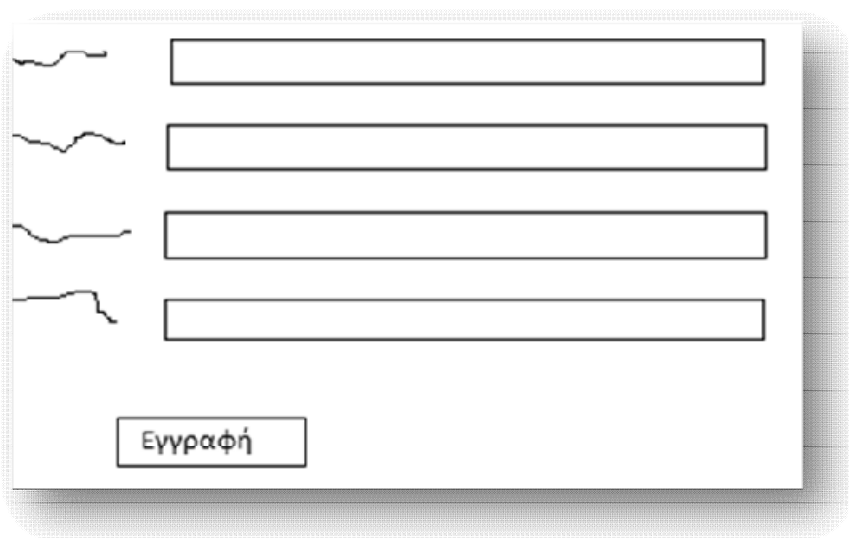


### 3.3.10 ΔΟΜΗ ΣΕΛΙΔΑΣ registration.php

Ο ρόλος της σελίδας registration.php είναι όπως προδίδει και ο τίτλος της να παρέχει την δυνατότητα στους επισκέπτες του portal να κάνουν αίτηση εγγραφής μέλους στο portal. Αφού συμπληρωθούν τα στοιχεία, το σύστημα διενεργεί ελέγχους εγκυρότητας στοιχείων:

- Ελέγχει αν στην βάση δεδομένων υπάρχει εγγεγραμμένος χρήστης με το ίδιο username ή με το ίδιο e-mail.
- Ελέγχει εάν τα ζητούμενα πεδία είναι ορθώς συμπληρωμένα

Μετά το πέρας των ελέγχων, δημιουργείται εγγραφή του νέου χρήστη στη βάση δεδομένων θέτοντας στην τιμή active\_flag = 0. Παράλληλα, ενεργοποιεί την κλάση AteiMailer του kernel και στέλνει μήνυμα στην διεύθυνση e-mail που δήλωσε ο χρήστης στην φόρμα με έναν σύνδεσμο που οδηγεί στο registration.php έχοντας αυτή τη φορά 3 παραμέτρους GET, την παράμετρο **action** η οποία έχει την τιμή **activation**, την τιμή **userid** που έχει ως τιμή το id της νεοεισαχθείσας εγγραφής χρήστη στη βάση δεδομένων την οποία και θέλουμε να ενεργοποιήσουμε καθώς και την παράμετρο **code** που έχει ως τιμή ανεστραμμένο το αποτέλεσμα της md5 της συνένωσης του e-mail, του κωδικού και του username. Όταν η registration.php δέχεται την παράμετρο action ίση με την τιμή **activation** αλλάζει την λειτουργία της και επιχειρεί επαλήθευση της τιμής code με βάση τα υπάρχοντα στοιχεία στη βάση δεδομένων. Εάν αυτό επιτύχει, τότε εκτελεί update στην εγγραφή του χρήστη στη βάση, θέτοντας την στήλη **active\_flag** ίση με **1**, ενεργοποιώντας ουσιαστικά τον λογαριασμό του χρήστη, δίνοντάς του το δικαίωμα να συνδεθεί στον λογαριασμό του με το username και τον κωδικό πρόσβασής του. Η δομή της σελίδας αυτής, ακολουθεί τη δομή πεδίων εισαγωγής κάθετα στοιχισμένα όπως απαντώνται στην πλειοψηφία των φορμών εγγραφής στο internet:



The image shows a registration form with four input fields stacked vertically. Each field is preceded by a small icon: a wavy line for the first, a jagged line for the second, a wavy line for the third, and a jagged line for the fourth. Below the input fields is a rectangular button with the text 'Εγγραφή' (Registration).

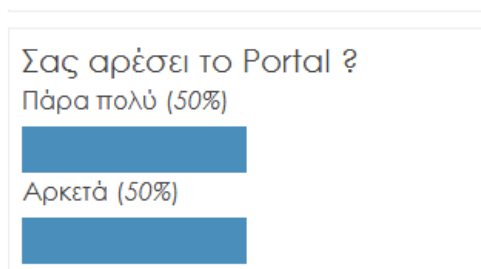


### 3.3.11 ΔΟΜΗ ΣΕΛΙΔΑΣ login.php

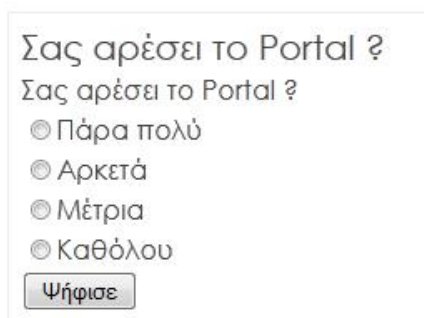
Η σελίδα **login.php** επιτελεί τον σκοπό της παροχής δυνατότητας στον επισκέπτη του portal να συνδεθεί στον λογαριασμό του. Του παρέχει μια φόρμα με το πρώτο πεδίο να δέχεται είτε το **email** είτε το username του και το δεύτερο πεδίο να δέχεται τον κωδικό πρόσβασης. Μόλις ο επισκέπτης συμπληρώσει αυτά τα 2 πεδία και πατήσει το κουμπί είσοδος, το σύστημα ελέγχει την εγκυρότητα των στοιχείων με βάση τις εγγραφές του πίνακα users στην βάση δεδομένων. Σε περίπτωση που βρεθεί εγγραφή στη βάση που φέρει το δοθέν username ή email και password, αυτό πιστοποιεί την εγκυρότητα του χρήστη, με αποτέλεσμα το σύστημα να δημιουργεί το authentication cookie το οποίο το αποστέλλει στον φυλλομετρητή του χρήστη, το οποίο το αντιστοιχεί στον σειριακό κωδικό της συνεδρίας που μόλις δημιουργήθηκε στο διακομιστή. Κατόπιν, ο χρήστης ανακατευθύνεται αυτόματα στην κεντρική σελίδα επιλογών του λογαριασμού του. Στην σελίδα **login.php** παρέχεται επίσης η δυνατότητα της αποστολής του κωδικού πρόσβασης του χρήστη στην δηλωμένη διεύθυνση email καθώς και ένας σύνδεσμος που οδηγεί τους νέους χρήστες οι οποίοι για οποιονδήποτε λόγο βρέθηκαν σε αυτή τη σελίδα χωρίς να είναι μέλη, στην σελίδα **registration.php**.

### 3.3.12 ΔΟΜΗ ΣΥΣΤΗΜΑΤΟΣ ΔΗΜΟΣΚΟΠΗΣΕΩΝ

Το σύστημα δημοσκοπήσεων, αποτελείται από 3 πίνακες βάσης δεδομένων στους οποίους αναφερθήκαμε στο κεφάλαιο «ΔΟΜΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ», από την κλάση Ateipoll του kernel που περιλαμβάνει τον βασικό κώδικα του συστήματος ο οποίος είναι υπεύθυνος για την εμφάνιση των ερωτήσεων και των διαθέσιμων επιλογών τους, για την καταχώρηση κάθε νέας ψήφου τοποθετώντας παράλληλα cookie στον φυλλομετρητή του χρήστη ώστε να απορρίπτονται οι διπλές ψήφοι. Τέλος, η δομή του συστήματος δημοσκοπήσεων περιλαμβάνει ένα block με όνομα **poll** που περιλαμβάνει τον κώδικα εμφάνισης είτε των διαθέσιμων επιλογών της δημοσκόπησης είτε των αποτελεσμάτων εφόσον ο χρήστης έχει ψηφίσει.



Σας αρέσει το Portal ?  
Πάρα πολύ (50%)  
Αρκετά (50%)



Σας αρέσει το Portal ?  
Σας αρέσει το Portal ?  
 Πάρα πολύ  
 Αρκετά  
 Μέτρια  
 Καθόλου

#### 4. ΔΙΑΧΕΙΡΗΣΤΙΚΑ ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ

Το Portal του ΑΤΕΙ Αμαλιάδας περιλαμβάνει διαχείριση στην οποία έχουν πρόσβαση όσοι χρήστες έχουν την τιμή **1** στο πεδίο **is\_admin** στον πίνακα **users**. Οι επιλογές διαχείρισης αφορούν όλο το εύρος των υπηρεσιών του portal. Ο Διαχειριστής έχει την δυνατότητα να προσθέτει και να αφαιρεί rss feeds και να ρυθμίζει τις παραμέτρους τους, να προσθέτει νέα άρθρα, ανακοινώσεις και στατικές σελίδες που περιλαμβάνουν πχ τα στοιχεία επικοινωνίας του ΤΕΙ. Σκοπός του Portal είναι να συγκεντρώσει και να παρουσιάσει στον επισκέπτη όλες τις πληροφορίες οι οποίες ενδιαφέρουν κάθε μέλος της εκπαιδευτικής κοινότητας του ΑΤΕΙ Αμαλιάδας.

Για παράδειγμα, στην κεντρική σελίδα, χάρη στην τεχνολογία RSS, εμφανίζονται τοπικές ειδήσεις από την Αμαλιάδα (πηγή Google), τελευταία νέα που αφορούν σπουδαστικά ζητήματα, οι τελευταίες ανακοινώσεις του e-class, τελευταία νέα που αφορούν το ΤΕΙ της Πάτρας καθώς και οι τελευταίες ανακοινώσεις του Υπουργείου παιδείας.

Συνυπολογίζοντας όλα τα παραπάνω, διαπιστώνουμε πως ο επισκέπτης (και μέλος του ΑΤΕΙ Αμαλιάδας) με την επίσκεψή του στο Portal, ενημερώνεται για κάθε θέμα που τον αφορά σχετικά με τις σπουδές του αν είναι φοιτητής ή την εργασία του αν είναι συνεργάτης.

Τέλος, μέσω της διαδικασίας των ψηφοφοριών, υπάρχει η δυνατότητα να τίθενται στους επισκέπτες ερωτήσεις που αφορούν κρίσιμα ζητήματα της λειτουργίας του τμήματος ΕΠΔΟ ώστε να λαμβάνεται υπόψη η άποψη των μελών της εκπαιδευτικής κοινότητας.

## 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Τα συμπεράσματα της εργασίας είναι αρκετά θετικά ως προς την δυνατότητα δημιουργίας ενός μέσου που αναλαμβάνει να συγκεντρώσει και να παρουσιάσει όσο το δυνατόν περισσότερες πληροφορίες που αφορούν την εκπαιδευτική λειτουργία ενός τμήματος ΑΤΕΙ σαν το τμήμα ΕΠΔΟ Αμαλιάδας. Με βάση τις εμπειρίες της παρούσας εργασίας, η υλοποίηση ενός έργου σαν αυτό της παρούσας εργασίας, δηλαδή το Portal του ΑΤΕΙ Αμαλιάδας. Για την συγκέντρωση των επιμέρους πληροφοριών, όσων δηλαδή δεν καταχωρούνται απευθείας στο σύστημα, όπως είναι τα άρθρα, οι ανακοινώσεις, οι σελίδες και οι δημοσκοπήσεις, όλα τα υπόλοιπα δεδομένα αντλούνται μέσω της τεχνολογίας RSS. Στην παρούσα φάση, καθαρές πηγές RSS εντοπίστηκαν μόνο στην πλατφόρμα e-class του teipat.gr και χρησιμοποιήθηκε στην παρούσα εργασία. Εάν αρκετοί δικτυακοί τόποι του teipat.gr υποστήριζαν τεχνολογία RSS θα ήταν εφικτό να συγκεντρώσουμε και να προβάλλουμε ακόμα περισσότερες πληροφορίες στο portal. Αν προστεθεί RSS σε κάθε δικτυακό τόπο που αφορά το ΤΕΙ Πάτρας, τότε η βελτίωση των παρεχόμενων υπηρεσιών του Portal είναι δεδομένη, αφού θα είναι σε θέση να παρουσιάσει ακόμα περισσότερες και καλύτερα στοχευμένες πληροφορίες στους επισκέπτες. Στο portal είναι εφικτό να προστεθούν αρκετές νέες δυνατότητες στο μέλλον, όπως για παράδειγμα η δυνατότητα υποβολής αιτήσεων στην γραμματεία (είτε απευθείας είτε μέσω κάποιου API/Application Programming Interface παρόντος συστήματος), σύστημα ανακοινώσεων βαθμολογιών και άλλα. Όλα αυτά, σε συνεργασία με τα μέλη της εκπαιδευτικής κοινότητας, ώστε το portal πέρα από κόμβο ενημέρωσης, να αποτελέσει και σημαντικό εργαλείο στην εκπαιδευτική διαδικασία.

Μελλοντικές βελτιώσεις είναι δυνατόν να υπάρξουν και στο επίπεδο των παρεχόμενων υπηρεσιών. Για παράδειγμα, μια από αυτές τις βελτιώσεις θα ήταν δυνατό να περιλαμβάνει την εξής νέα δυνατότητα προς στους εγγεγραμμένους χρήστες: Να λαμβάνουν στην ηλεκτρονική διεύθυνση που έχουν δηλώσει κατά την εγγραφή τους κάθε νέα ανακοίνωση που αναρτάται στο portal. Αυτό αποτελεί σημαντική βελτίωση, μιας και εξαλείφεται στο ελάχιστο ο κίνδυνος κάποια ανακοίνωση να μην γίνει αντιληπτή από τον χρήστη, εάν την δεδομένη στιγμή της ανάρτησης δεν επισκεφθεί το Portal. Μελλοντικές βελτιώσεις σαν αυτή που μόλις αναφέρθηκε, είναι αρκετά εύκολο να πραγματοποιηθούν, δεδομένης της απλής δομής του portal. Έτσι με σχετική ευκολία, το portal έχει την ικανότητα να προσαρμόζεται στην εκάστοτε ανάγκες της εκπαιδευτικής κοινότητας.

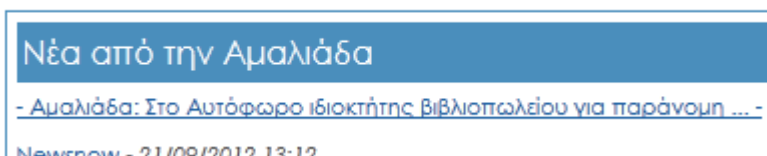
## 6. ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ ΤΟΥ PORTAL ΜΕΣΑ ΑΠΟ ΣΤΙΓΜΙΟΤΥΠΑ (SCREENSHOTS)

Στο στιγμιότυπο που ακολουθεί βλέπουμε την κεντρική σελίδα του portal. Δίπλα στο λογότυπο, διακρίνεται το εορτολόγιο, στο κυρίως μέρος, η θερμοκρασία της Αμαλιάδας (δεξιά) ενώ αριστερά βρίσκεται τα κορυφαία άρθρα (το νεότερο αναλυτικά και τα παλαιότερα σε τίτλους). Επάνω από τα κορυφαία άρθρα, εμφανίζεται η τελευταία κορυφαία ανακοίνωση. Η θέση της δεν επιλέχθηκε τυχαία, μιας και σε αυτό το σημείο θα γίνει αντιληπτή από τους περισσότερους επισκέπτες.

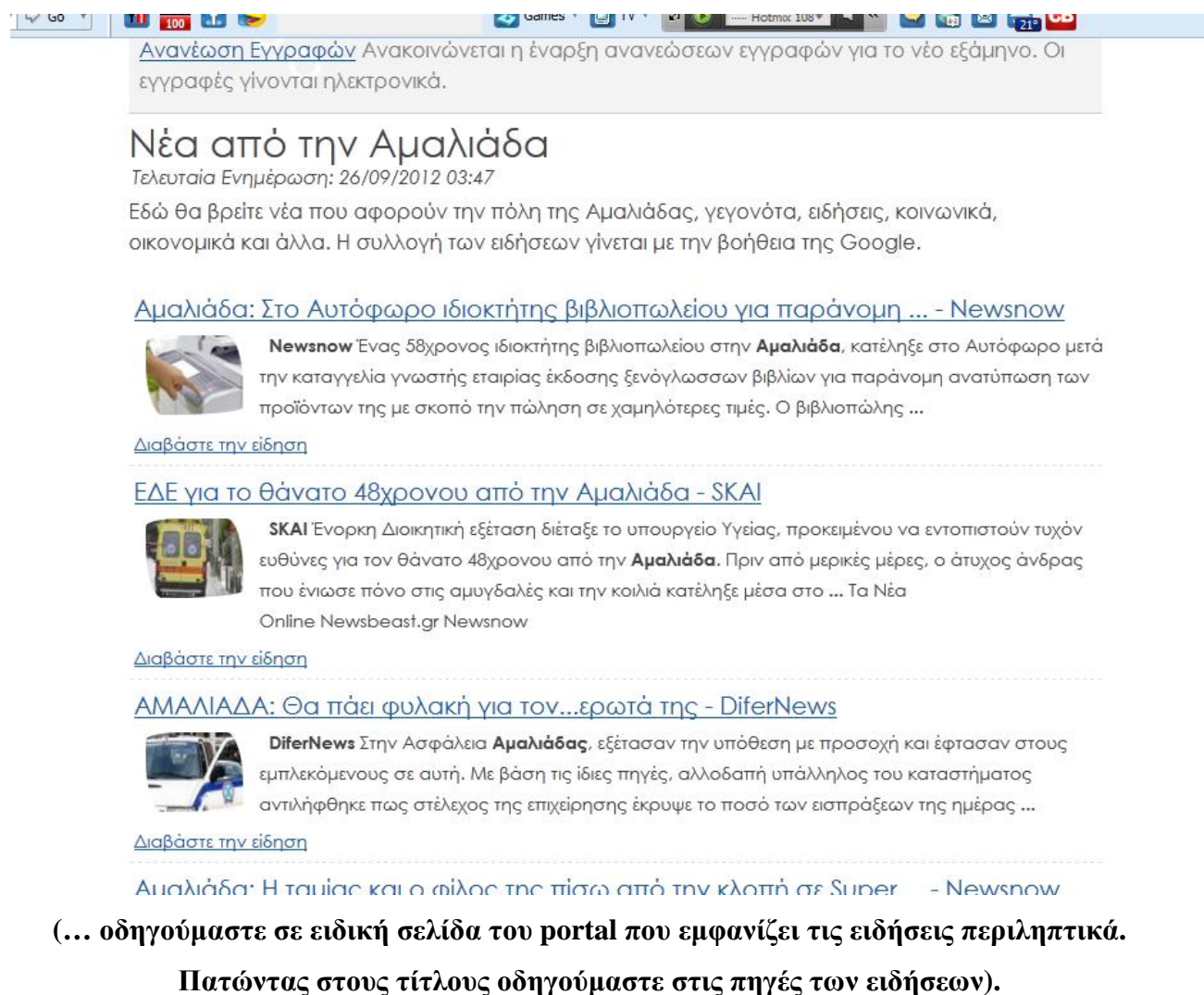


(Στιγμιότυπο Κεντρικής Σελίδας)

Κάτω από τα κορυφαία άρθρα εμφανίζονται οι επικεφαλίδες των ειδήσεων που αντλούνται από άλλους δικτυακούς τόπους μέσω RSS (οι οποίες να αναφέρουμε εδώ πως διατηρούνται στον διακομιστή για όσο χρόνο σε δευτερόλεπτα ορίζει η τιμή refresh\_intv του πίνακα rss\_feeds). Πατώντας στην κάθε επικεφαλίδα, οδηγούμαστε απευθείας στην πηγή της είδησης, ενώ πατώντας στην επικεφαλίδα του κάθε πλαισίου, βλέπουμε όλες τις διαθέσιμες επικεφαλίδες ειδήσεων.



(Πατώντας το «Νέα από την Αμαλιάδα» ...)




Ανανέωση Εγγραφών Ανακοινώνεται η έναρξη ανανεώσεων εγγραφών για το νέο εξάμηνο. Οι εγγραφές γίνονται ηλεκτρονικά.

## Νέα από την Αμαλιάδα

Τελευταία Ενημέρωση: 26/09/2012 03:47

Εδώ θα βρείτε νέα που αφορούν την πόλη της Αμαλιάδας, γεγονότα, ειδήσεις, κοινωνικά, οικονομικά και άλλα. Η συλλογή των ειδήσεων γίνεται με την βοήθεια της Google.


[Αμαλιάδα: Στο Αυτόφωρο ιδιοκτήτης βιβλιοπωλείου για παράνομη ... - Newsnow](#)



**Newsnow** Ένας 58χρονος ιδιοκτήτης βιβλιοπωλείου στην **Αμαλιάδα**, κατέληξε στο Αυτόφωρο μετά την καταγγελία γνωστής εταιρίας έκδοσης ξενόγλωσσων βιβλίων για παράνομη ανατύπωση των προϊόντων της με σκοπό την πώληση σε χαμηλότερες τιμές. Ο βιβλιοπώλης ...

[Διαβάστε την είδηση](#)


[ΕΔΕ για το θάνατο 48χρονου από την Αμαλιάδα - SKAI](#)



**SKAI** Ένορκη Διοικητική εξέταση διέταξε το υπουργείο Υγείας, προκειμένου να εντοπιστούν τυχόν ευθύνες για τον θάνατο 48χρονου από την **Αμαλιάδα**. Πριν από μερικές μέρες, ο άτυχος άνδρας που ένωσε πόνο στις αμυγδαλές και την κοιλιά κατέληξε μέσα στο ... Τα Νέα Online Newsbeast.gr Newsnow

[Διαβάστε την είδηση](#)

[ΑΜΑΛΙΑΔΑ: Θα πάει φυλακή για τον...ερωτά της - DiferNews](#)



**DiferNews** Στην Ασφάλεια **Αμαλιάδας**, εξέτασαν την υπόθεση με προσοχή και έφτασαν στους εμπλεκόμενους σε αυτή. Με βάση τις ίδιες πηγές, αλλοδαπή υπάλληλος του καταστήματος αντιλήφθηκε πως στέλεχος της επιχείρησης έκρυψε το ποσό των εισπράξεων της ημέρας ...

[Διαβάστε την είδηση](#)

[Αμαλιάδα: Η ταίριας και ο φίλος της πίσω από την κλοπή σε Super - Newsnow](#)

(... οδηγούμαστε σε ειδική σελίδα του portal που εμφανίζει τις ειδήσεις περιληπτικά.

**Πατώντας στους τίτλους οδηγούμαστε στις πηγές των ειδήσεων).**

Στην κεντρική σελίδα του Portal όπως είδαμε προηγουμένως, διακρίνουμε τα άρθρα, όπως φαίνονται στο screenshot που ακολουθεί:

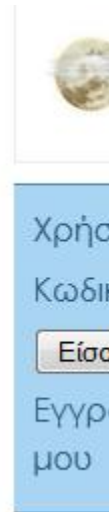
## Αυτό είναι το Νέο Site!



Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας

[Δηλώσεις Μαθημάτων](#) - Σας ενημερώνουμε ότι οι δηλώσεις ... - 15/09/2012 00:00

[Νέο Site!](#) - Νέο Site Νέο Site Νέο Site Νέο Site... - 15/09/2012 00:00



Πατώντας είτε στον τίτλο “Αυτό είναι το Νέο Site!”, είτε στις επικεφαλίδες των επόμενων άρθρων, κατευθυνόμαστε στην επόμενη σελίδα που εμφανίζει με λεπτομέρεια το άρθρο:

[Ανανέωση Εγγραφών](#) Ανακοινώνεται η έναρξη ανανέωσης εγγραφών για το νέο εξάμηνο. Οι εγγραφές γίνονται ηλεκτρονικά.

## Αυτό είναι το Νέο Site!

Δημοσίευση: 15/09/2012 00:00



Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδας

Επίσης, κάνοντας κλικ στο «**Άρθρα**» του μενού πλοήγησης, οδηγούμαστε στην σελίδα article.php εμφανίζονται τα άρθρα, ταξινομημένα κατά ημερομηνία, με φθίνουσα σειρά:

## Τελευταία Άρθρα

### Αυτό είναι το Νέο Site!

Δημοσίευση: 2012-09-15 00:00:00



Αυτό είναι το νέο site του ΑΤΕΙ Αμαλιάδα  
Αμαλιάδας Αυτό είναι το νέο site του ΑΤΙ  
ΑΤΕΙ Αμαλιάδας Αυτό είναι το νέο site το

### Δηλώσεις Μαθημάτων

Δημοσίευση: 2012-09-15 00:00:00



Σας ενημερώνουμε ότι οι δηλώσεις μαθη  
πραγματοποιηθούν από τις 24-09-2012  
κωδικούς σας μπορείτε να εισέρχεστε σι

### Νέο Site!

Δημοσίευση: 2012-09-15 00:00:00



Νέο Site Νέο Site Νέο Site Νέο Site Νέο Si  
Νέο Site Νέο Site Νέο Site Νέο Site Νέο Si

Επίσης, διακρίνουμε επίσης το πλαίσιο με τις τελευταίες ανακοινώσεις:

#### Ανανέωση Εγγραφών - 01/01/1970 02:00

Ανακοινώνεται η έναρξη ανανεώσεων  
εγγραφών για το νέο εξάμηνο. Οι εγγραφές  
γίνονται ηλεκτρονικά...

#### Ανακοίνωση Ορκωμοσίας

#### Πτυχιούχων - 01/01/1970 02:00

Σας προσκαλούμε στην τελετή  
καθομολόγησης των πτυχιούχων του  
Τμήματος Εφαρμογών Πληροφορικής στ...

Πατώντας ομοίως και εδώ στις επικεφαλίδες, κατευθυνόμαστε στην σελίδα που προβάλλει αναλυτικά την εκάστοτε ανακοίνωση (announcements.php):

[Ανανέωση Εγγραφών](#) Ανακοινώνεται η έναρξη ανανεώσεων εγγραφών για το νέο εξάμηνο. Οι εγγραφές γίνονται ηλεκτρονικά.

## Σελίδα Ανακοινώσεων Ανανέωση Εγγραφών

Δημοσίευση: 2012-09-12 15:16:01

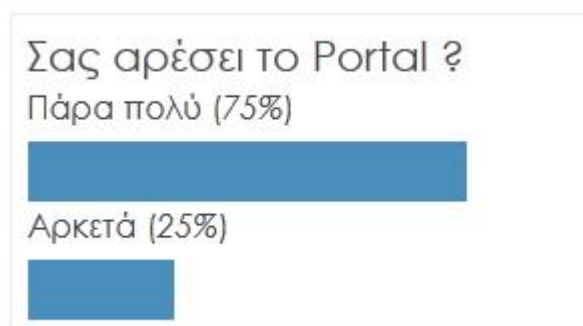
Ανακοινώνεται η έναρξη ανανεώσεων εγγραφών για το νέο εξάμηνο. Οι εγγραφές γίνονται ηλεκτρονικά.

Στην κεντρική σελίδα, εμφανίζεται η δημοσκόπηση (η ψηφοφορία). Εάν δεν έχουμε ακόμα ψηφίσει τότε το πλαίσιο της δημοσκόπησης, φαίνεται ως εξής:

Σας αρέσει το Portal ?  
Σας αρέσει το Portal ?

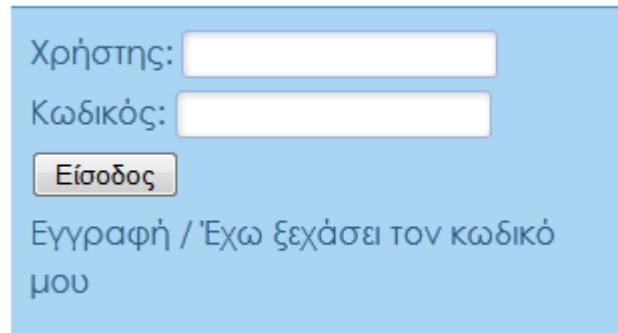
Πάρα πολύ  
 Αρκετά  
 Μέτρια  
 Καθόλου

Για να ψηφίσουμε, επιλέγουμε την απάντηση που επιθυμούμε και πατάμε το **Ψήφισε**. Η ψήφος μας καταχωρείται και προσμετράται στα αποτελέσματα που βλέπουμε ευθύς αμέσως:





Για να συνδεθεί ο χρήστης στον λογαριασμό του (ή ο διαχειριστής στον πίνακα διαχείρισης), αρκεί να συμπληρωθεί το όνομα χρήστη και ο κωδικός πρόσβασης στο πλαίσιο εισόδου και να κάνει κλικ στο κουμπί «είσοδος». Επιπροσθέτως υπάρχουν οι σύνδεσμοι «Εγγραφή» και «Έχω ξεχάσει τον κωδικό μου» που δίνουν την δυνατότητα στον επισκέπτη είτε να δημιουργήσει λογαριασμό είτε να μεταβεί στην διαδικασία ανάκτησης του κωδικού πρόσβασης.



Χρήστης:

Κωδικός:

[Εγγραφή / Έχω ξεχάσει τον κωδικό μου](#)

Αν ο χρήστης που συνδέθηκε είναι Διαχειριστής (Administrator), τότε μεταφέρεται αυτομάτως στον **πίνακα διαχείρισης** του portal.



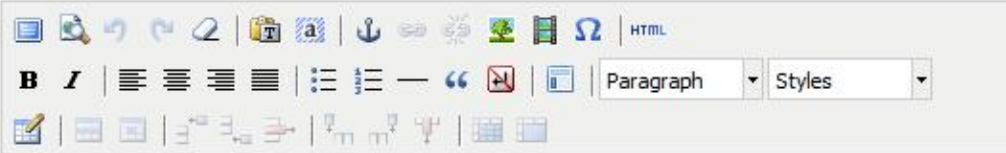
(Πλαίσιο βασικών επιλογών. Για τον σκοπό του παραδείγματος, πατάμε «Προσθήκη Ανακοίνωσης» )

Αφού πατήσουμε «**Προσθήκη Ανακοίνωσης**», οδηγούμαστε στην φόρμα προσθήκης ανακοίνωσης, όπως φαίνεται στο screenshot που ακολουθεί:

Τίτλος Ανακοίνωσης:

Εμφάνιση ανακοίνωσης:

Κείμενο Ανακοίνωσης:



Σας προσκαλούμε στην τελετή καθομολόγησης των πτυχιούχων του Τμήματος Εφαρμογών Πληροφορικής στη Διοίκηση και στην Οικονομία της Σχολής Διοίκησης και Οικονομίας, που θα πραγματοποιηθεί στις εγκαταστάσεις του Τμήματος (Θεοδωροπούλου Τέρμα στην Αμαλιάδα) την Τετάρτη 26 Σεπτεμβρίου 2012 & ώρα 13:00 μ.μ.  
Ο Προϊστάμενος του Τμήματος  
Δρ. Ιωάννης Δ. Ζαχαράκης  
Επικ. Καθηγητής

Αφού ολοκληρώσουμε την σύνταξη της ανακοίνωσης, πατάμε “**Προσθήκη**” και η ανακοίνωσή εμφανίζεται στην κεντρική σελίδα:

[Ανακοίνωση Ορκωμοσίας Πτυχιούχων - 01/01/1970 02:00](#)  
Σας προσκαλούμε στην τελετή καθομολόγησης των πτυχιούχων του Τμήματος Εφαρμογών Πληροφορικής στ...

## 7. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. WIKIPEDIA / Ελεύθερη OnLine Βιβλιοθήκη
  - 1.1 <http://el.wikipedia.org/wiki/> Διαδίκτυο
  - 1.2 [http://en.wikipedia.org/wiki/Web\\_portal](http://en.wikipedia.org/wiki/Web_portal)
  - 1.3 <http://el.wikipedia.org/wiki/> Ιστοσελίδα
  - 1.4 <http://el.wikipedia.org/wiki/> Web Browsers
  - 1.5 <http://en.wikipedia.org/wiki/Http>
  - 1.6 <http://en.wikipedia.org/wiki/Https>
  - 1.7 <http://en.wikipedia.org/wiki/Html>
  - 1.8 <http://en.wikipedia.org/wiki/Xml>
  - 1.9 <http://en.wikipedia.org/wiki/Www>
  - 1.10 [http://en.wikipedia.org/wiki/Apache\\_server](http://en.wikipedia.org/wiki/Apache_server)
  - 1.11 <http://en.wikipedia.org/wiki/Php>
  - 1.12 <http://en.wikipedia.org/wiki/Mysql>
  - 1.13 <http://en.wikipedia.org/wiki/Css>
  - 1.14 [http://en.wikipedia.org/wiki/Content\\_Management\\_System](http://en.wikipedia.org/wiki/Content_Management_System)
  - 1.15 <http://en.wikipedia.org/wiki/RSS>
  - 1.16 [http://en.wikipedia.org/wiki/Domaine Name](http://en.wikipedia.org/wiki/Domaine_Name)
2. PHP 5.3 is quicker than PHP 5.2: <http://akrobat.com/php/php-5-3-is-quicker-than-php-5-2-2/>
3. [http://metatorial.com/downloads/Boiko\\_Wp\\_UnderstandingContentManagemement.pdf](http://metatorial.com/downloads/Boiko_Wp_UnderstandingContentManagemement.pdf)

## 8. ΠΑΡΑΡΤΗΜΑ

### Αρχικοποίηση πίνακα eortologio με τις εγγραφές των γιορτών:

---

```
INSERT INTO `eortologio` VALUES (7, 1, 'Ιωάννης, Ιωάννα, Ιώ, Πρόδρομος');
```

```
INSERT INTO `eortologio` VALUES (1, 1, 'Βασίλης, Βασιλική');
```

```
INSERT INTO `eortologio` VALUES (6, 1, 'Φώτης, Φωτεινή, Θεοφάνης, Θεοφανία, Φανή');
```

```
INSERT INTO `eortologio` VALUES (9, 1, 'Ευστράτιος, Ευστρατία');
```

```
INSERT INTO `eortologio` VALUES (11, 1, 'Θεοδόσιος');
```

```
INSERT INTO `eortologio` VALUES (12, 1, 'Τατιάνα');
```

```
INSERT INTO `eortologio` VALUES (17, 1, 'Αντώνης, Αντωνία');
```

```
INSERT INTO `eortologio` VALUES (18, 1, 'Θανάσης, Αθανασία');
```

```
INSERT INTO `eortologio` VALUES (20, 1, 'Ευθύμιος, Ευθυμία');
```

```
INSERT INTO `eortologio` VALUES (21, 1, 'Μάξιμος');
```

```
INSERT INTO `eortologio` VALUES (22, 1, 'Τιμόθεος');
```

```
INSERT INTO `eortologio` VALUES (24, 1, 'Ξένη');
```

```
INSERT INTO `eortologio` VALUES (25, 1, 'Γρηγόρης, Γρηγορία');
```

```
INSERT INTO `eortologio` VALUES (26, 1, 'Ξενοφώντας');
```

```
INSERT INTO `eortologio` VALUES (1, 2, 'Τρύφωνας');
```

```
INSERT INTO `eortologio` VALUES (2, 2, 'Ιορδάνης');
```

```
INSERT INTO `eortologio` VALUES (4, 2, 'Ισίδωρος');
```

```
INSERT INTO `eortologio` VALUES (5, 2, 'Αγαθή');
```

```
INSERT INTO `eortologio` VALUES (7, 2, 'Παρθένα, Παρθένιος');
```

```
INSERT INTO `eortologio` VALUES (8, 2, 'Ζαχαρίας, Ζάχος, Ζαχαρούλα');
```

```
INSERT INTO `eortologio` VALUES (9, 2, 'Νικηφόρος');
```

```
INSERT INTO `eortologio` VALUES (10, 2, 'Χαράλαμπος, Χαρίκλεια, Κλειώ');
```

```
INSERT INTO `eortologio` VALUES (11, 2, 'Βλάσσης');
```

```
INSERT INTO `eortologio` VALUES (12, 2, 'Μελέτης');
```

```
INSERT INTO `eortologio` VALUES (14, 2, 'Βαλεντίνος, Βαλεντίνα');
```

```
INSERT INTO `eortologio` VALUES (18, 2, 'Λέων');
```

```
INSERT INTO `eortologio` VALUES (19, 2, 'Φιλοθέη');
```

```
INSERT INTO `eortologio` VALUES (23, 2, 'Πολύκαρπος');
```

```
INSERT INTO `eortologio` VALUES (26, 2, 'Πορφύριος');
```

```
INSERT INTO `eortologio` VALUES (27, 2, 'Ασκληπιός');
```

```
INSERT INTO `eortologio` VALUES (1, 3, 'Ευδοκία');
```

```
INSERT INTO `eortologio` VALUES (2, 3, 'Θάλια');
```

```
INSERT INTO `eortologio` VALUES (8, 3, 'Ερμής');
```

```
INSERT INTO `eortologio` VALUES (17, 3, 'Αλέξιος, Αλεξία, Αλίκη');
```

```
INSERT INTO `eortologio` VALUES (18, 3, 'Θεόδωρος');
```

```
INSERT INTO `eortologio` VALUES (19, 3, 'Χρυσανθος, Χρυσάνθη');
```

```
INSERT INTO `eortologio` VALUES (25, 3, 'Βαγγέλης, Ευαγγελία');
```

```
INSERT INTO `eortologio` VALUES (6, 4, 'Ευτύχιος, Ευτυχία');
```

```
INSERT INTO `eortologio` VALUES (14, 4, 'Ανθή');
```

```
INSERT INTO `eortologio` VALUES (15, 4, 'Λεωνίδα');
```

```
INSERT INTO `eortologio` VALUES (20, 4, 'Ζωή, Πηγή');
```

```
INSERT INTO `eortologio` VALUES (22, 4, 'Ναθαναήλ');
```

```
INSERT INTO `eortologio` VALUES (23, 4, 'Γιώργος, Γεωργία');
```

```
INSERT INTO `eortologio` VALUES (24, 4, 'Ελισσάβετ, Λίζα');
```

---

---

```
INSERT INTO `eortologio` VALUES (25, 4, 'Μάρκος');
INSERT INTO `eortologio` VALUES (29, 4, 'Ίάσωνας');
INSERT INTO `eortologio` VALUES (30, 4, 'Ίάκωβος');
INSERT INTO `eortologio` VALUES (1, 5, 'Ιερεμίας');
INSERT INTO `eortologio` VALUES (5, 5, 'Ειρήνη');
INSERT INTO `eortologio` VALUES (8, 5, 'Αρσένιος, Θεολόγος');
INSERT INTO `eortologio` VALUES (9, 5, 'Ήσαΐας, Χριστόφορος');
INSERT INTO `eortologio` VALUES (11, 5, 'Μεθόδιος');
INSERT INTO `eortologio` VALUES (13, 5, 'Γλυκερία');
INSERT INTO `eortologio` VALUES (18, 5, 'Ιουλία');
INSERT INTO `eortologio` VALUES (21, 5, 'Κωνσταντίνος, Ελένη');
INSERT INTO `eortologio` VALUES (29, 5, 'Θεοδοσία');
INSERT INTO `eortologio` VALUES (4, 6, 'Μάρθα');
INSERT INTO `eortologio` VALUES (5, 6, 'Δωροθέος, Δωροθέα, Απόλλωνας');
INSERT INTO `eortologio` VALUES (8, 6, 'Καλλιόπη');
INSERT INTO `eortologio` VALUES (11, 6, 'Βαρναβάς, Βαρθολομαίος');
INSERT INTO `eortologio` VALUES (22, 6, 'Ευσέβιος');
INSERT INTO `eortologio` VALUES (27, 6, 'Παντελής');
INSERT INTO `eortologio` VALUES (29, 6, 'Πάυλος, Πέτρος, Πετρούλα, Παυλίνα, Πωλίνα');
INSERT INTO `eortologio` VALUES (30, 6, 'Απόστολος');
INSERT INTO `eortologio` VALUES (1, 7, 'Αργύρης, Αργυρώ, Δαμιανός, Κοσμάς');
INSERT INTO `eortologio` VALUES (7, 7, 'Κυριάκος, Κυριακή');
INSERT INTO `eortologio` VALUES (8, 7, 'Θεόφιλος, Προκόπης');
INSERT INTO `eortologio` VALUES (11, 7, 'Ευθημία, Όλγα');
INSERT INTO `eortologio` VALUES (17, 7, 'Μαρίνος, Μαρίνα');
INSERT INTO `eortologio` VALUES (18, 7, 'Αμίλιος, Αμιλία');
INSERT INTO `eortologio` VALUES (20, 7, 'Ήλιος, Ηλιάνα');
INSERT INTO `eortologio` VALUES (22, 7, 'Μαγδαληνή, Μαρκέλα, Άννα');
INSERT INTO `eortologio` VALUES (26, 7, 'Παρασκευή, Παρασκευάς');
INSERT INTO `eortologio` VALUES (27, 7, 'Παντελεήμων, Παντελής, Λάκης');
INSERT INTO `eortologio` VALUES (29, 7, 'Καλλίνικος');
INSERT INTO `eortologio` VALUES (31, 7, 'Ιωσήφ');
INSERT INTO `eortologio` VALUES (6, 8, 'Σωτήρης, Σωτηρία');
INSERT INTO `eortologio` VALUES (8, 8, 'Αιμιλιανός, Τριαντάφυλλος');
INSERT INTO `eortologio` VALUES (10, 8, 'Λαυρέντης');
INSERT INTO `eortologio` VALUES (15, 8, 'Παναγιώτης, Παναγιώτα, Μαριάννα, Μαρία, Μάριος, Δέσποινα');
INSERT INTO `eortologio` VALUES (20, 8, 'Σαμουήλ');
INSERT INTO `eortologio` VALUES (25, 8, 'Τίτος, Πάστης');
INSERT INTO `eortologio` VALUES (26, 8, 'Ναταλία');
INSERT INTO `eortologio` VALUES (27, 8, 'Φανούριος, Φανουριά');
INSERT INTO `eortologio` VALUES (30, 8, 'Αλέξανδρος, Αλεξάνδρα, Αλέκα, Αλέκος');
INSERT INTO `eortologio` VALUES (1, 9, 'Συμεών');
INSERT INTO `eortologio` VALUES (3, 9, 'Άνθμος');
INSERT INTO `eortologio` VALUES (5, 9, 'Ζαχαρίας');
INSERT INTO `eortologio` VALUES (9, 9, 'Ιωακείμ');
INSERT INTO `eortologio` VALUES (11, 9, 'Ευανθία, Εύα');
INSERT INTO `eortologio` VALUES (14, 9, 'Σταύρος, Σταυρούλα');
INSERT INTO `eortologio` VALUES (15, 9, 'Νικήτας');
INSERT INTO `eortologio` VALUES (16, 9, 'Ευθημία, Έφη');
```

---

---

```
INSERT INTO `eortologio` VALUES (17, 9, 'Αγάπη, Σοφία, Ελπίδα');
INSERT INTO `eortologio` VALUES (18, 9, 'Αριάδνη');
INSERT INTO `eortologio` VALUES (20, 9, 'Στάθης, Ευσταθία');
INSERT INTO `eortologio` VALUES (23, 9, 'Ξανθιππη, Πολυξένη, Ξένια');
INSERT INTO `eortologio` VALUES (24, 9, 'Θέκλα, Μυρτώ');
INSERT INTO `eortologio` VALUES (25, 9, 'Φρόσω');
INSERT INTO `eortologio` VALUES (2, 10, 'Κυπριανός');
INSERT INTO `eortologio` VALUES (3, 10, 'Διονύσιος, Διονυσία');
INSERT INTO `eortologio` VALUES (4, 10, 'Τερόθεος, Φραγκίσκος');
INSERT INTO `eortologio` VALUES (5, 10, 'Χαριτίνη');
INSERT INTO `eortologio` VALUES (7, 10, 'Σέργιος');
INSERT INTO `eortologio` VALUES (8, 10, 'Πελαγία, Πέλλα');
INSERT INTO `eortologio` VALUES (10, 10, 'Ευλάμπιος, Ευλαμπία');
INSERT INTO `eortologio` VALUES (15, 10, 'Λουκιανός');
INSERT INTO `eortologio` VALUES (18, 10, 'Λουκάς');
INSERT INTO `eortologio` VALUES (20, 10, 'Αρτέμιος, Γεράσιμος');
INSERT INTO `eortologio` VALUES (21, 10, 'Σωκράτης');
INSERT INTO `eortologio` VALUES (23, 10, 'Ίάκωβος');
INSERT INTO `eortologio` VALUES (26, 10, 'Δήμητρα, Δημήτρης');
INSERT INTO `eortologio` VALUES (27, 10, 'Νέστωρας');
INSERT INTO `eortologio` VALUES (1, 11, 'Αργύρης, Αργυρώ, Δαμιανός, Κοσμάς');
INSERT INTO `eortologio` VALUES (8, 11, 'Αγγελική, Άγγελος, Ταξιάρχης, Γαρβύιλ, Μιχάλης, Σεραφείμ, Σταμάτης, Σταματία');
INSERT INTO `eortologio` VALUES (9, 11, 'Νεκτάριος, Νεκταρία');
INSERT INTO `eortologio` VALUES (10, 11, 'Ορέστης');
INSERT INTO `eortologio` VALUES (11, 11, 'Βίκτωρας, Βικτώρια, Μηνάς');
INSERT INTO `eortologio` VALUES (13, 11, 'Χρυσόστομος');
INSERT INTO `eortologio` VALUES (14, 11, 'Φίλιππος');
INSERT INTO `eortologio` VALUES (16, 11, 'Ματθαίος');
INSERT INTO `eortologio` VALUES (18, 11, 'Πλάτωνας');
INSERT INTO `eortologio` VALUES (21, 11, 'Μαρία, Δέσποινα, Παναγιώτα');
INSERT INTO `eortologio` VALUES (25, 11, 'Κατερίνα, Κάτια, Μερκούρης');
INSERT INTO `eortologio` VALUES (26, 11, 'Στυλιανός, Στέλιος, Στέλλα');
INSERT INTO `eortologio` VALUES (30, 11, 'Ανδρέας, Ανδριαννή');
INSERT INTO `eortologio` VALUES (4, 12, 'Βαρβάρα');
INSERT INTO `eortologio` VALUES (5, 12, 'Σάββας, Διογένης');
INSERT INTO `eortologio` VALUES (6, 12, 'Νίκος, Νικολέτα');
INSERT INTO `eortologio` VALUES (7, 12, 'Αμβρόσιος');
INSERT INTO `eortologio` VALUES (9, 12, 'Άννα');
INSERT INTO `eortologio` VALUES (12, 12, 'Σπύρος, Σπυριδούλα');
INSERT INTO `eortologio` VALUES (13, 12, 'Λουκία, Στράτος');
INSERT INTO `eortologio` VALUES (15, 12, 'Λευτέρης, Ελευθερία');
INSERT INTO `eortologio` VALUES (17, 12, 'Δανιήλ, Διονύσης, Διονυσία');
INSERT INTO `eortologio` VALUES (19, 12, 'Άγλαΐα');
INSERT INTO `eortologio` VALUES (20, 12, 'Γιγνάτιος');
INSERT INTO `eortologio` VALUES (21, 12, 'Θεμιστοκλής');
INSERT INTO `eortologio` VALUES (22, 12, 'Αναστασία');
INSERT INTO `eortologio` VALUES (24, 12, 'Ευγένιος, Ευγενία');
INSERT INTO `eortologio` VALUES (25, 12, 'Χρήστος, Χριστίνα, Χρύσα, Μανώλης');
```

---

---

*INSERT INTO `eortologio` VALUES (27, 12, 'Στέφανος, Στεφανία');*

---

## Ενδεικτικά Αποσπάσματα πηγαίου κώδικα του Portal

### Kernel.php - Το κεντρικό αρχείο του Portal [3]

```
<?php

if(!defined('ATEIPORTAL')) die('You are not allowed to access this file directly.');
```

**require 'configuration.php';**

**// dk Core Function PHP 5.3**

```
$GLOBALS['_cb_init'] = Array();
```

```
$GLOBALS['_cb_init_called'] = Array();
```

```
$GLOBALS['DK'] = Array();
```

```
function dkSetInit($init, $cb) {
```

```
    $GLOBALS['_cb_init'][$init][] = $cb;
```

```
}
```

```
function dkInit($init) {
```

```
    if(func_num_args() > 1) $init = func_get_args();
```

```
    if(is_array($init)) foreach($init as $i) dkInit($i);
```

```
    else foreach((array)$GLOBALS['_cb_init'][$init] as $k=>$i) {
```

```
        if(is_callable($i)) $i();
```

```
        unset($GLOBALS['_cb_init'][$init][$k]);
```

```
    }
```

```
}
```

```
function dkGlobal($key,$val='') {
```

```
if($val) {

    $GLOBALS['DK'][$key] = $val;

} else {

    return $GLOBALS['DK'][$key]?$GLOBALS['DK'][$key]:false;

}

}

function loadBlock($block,$args=array()){

    ob_start();

    unset($args[0]);

    $args = array_values($args);

    if(file_exists(Configuration::$ROOT_PATH.'/blocks/'.$block.'.php')) {

        require Configuration::$ROOT_PATH.'/blocks/'.$block.'.php';

    } else echo "Block {$block} not exists";

    $return = ob_get_contents();

    ob_end_clean();

    echo $return;

    return;

}

// Set Inits Bellow This Line

dkSetInit('database', function() {

    $GLOBALS['DB'] = new PDO ( 'mysql:host=' . Configuration::$DB_Host . ';dbname='

. Configuration::$DB_Name, Configuration::$DB_User, Configuration::$DB_Pass, array

(PDO::ATTR_PERSISTENT => false, PDO::MYSQL_ATTR_INIT_COMMAND => 'SET

NAMES \\'UTF8\\' ) );
```



```
$GLOBALS['DB']->setAttribute ( PDO::ATTR_ERRMODE,  
PDO::ERRMODE_WARNING );  
  
});  
  
dkSetInit('template', function() {  
  
    ob_start();  
  
    register_shutdown_function(function() {  
  
        $MAIN_CONTENT = ob_get_contents();  
  
        ob_end_clean();  
  
        $ctp = dkGlobal('TEMPLATE_FILE');  
  
        if(file_exists(Configuration::$ROOT_PATH.'/template/' . ($ctp?$ctp:'index.php'))) {  
  
            // Template Vars  
  
            require  
Configuration::$ROOT_PATH.'/template/' . ($ctp?$ctp:'index.php');  
  
            } else die("Template file not found");  
  
        });  
  
    });  
  
dkSetInit('newssyndicator', function() {  
  
    require 'kernel/classes/NewsSyndicator.class.php';  
  
    dkGlobal('NewsSyndicator', new NewsSyndicator());  
  
});  
  
dkSetInit('ateiweather', function() {  
  
    require 'kernel/classes/AteiWeather.class.php';  
  
    dkGlobal('AteiWeather', new AteiWeather());  
  
});
```

```
});
```

```
dkSetInit('articles', function() {
```

```
    require 'kernel/classes/Articles.class.php';
```

```
    dkGlobal('Articles', new Articles());
```

```
});
```

```
dkSetInit('announcements', function() {
```

```
    require 'kernel/classes/Announcements.class.php';
```

```
    dkGlobal('Announcements', new Announcements());
```

```
});
```

```
dkSetInit('ateipoll', function() {
```

```
    require 'kernel/classes/AteiPoll.class.php';
```

```
    dkGlobal('AteiPoll', new AteiPoll());
```

```
});
```

```
dkSetInit('ateipages', function() {
```

```
    require 'kernel/classes/AteiPages.class.php';
```

```
    dkGlobal('AteiPages', new AteiPages());
```

```
});
```

```
dkSetInit('ateiuser', function() {
```

```
    require 'kernel/classes/AteiUser.class.php';
```

```
    dkGlobal('AteiUser', new AteiUser());
```

```
});
```

```
dkSetInit('ateinavigation', function() {
```

```
    require 'kernel/classes/AteiNavigation.class.php';
```

```
dkGlobal('AteiNavigation', new AteiNavigation());
```

```
});
```

```
?>
```

Οι μέθοδοι **dkSetInit** και **dkInit** επεξηγούνται στα κεφάλαια της εργασίας. Στο kernel, διακρίνουμε και την μέθοδο **dkGlobal** που ουσιαστικά χρησιμοποιείται για να δημιουργεί global scope μεταβλητές με συγκεκριμένο τρόπο (prefix prepend) ώστε να μην υπάρξει conflict με μεταβλητή που φέρει το ίδιο όνομα. Επίσης διακρίνουμε την **loadBlock** που δέχεται ως παράμετρο το όνομα του block που θέλουμε να εμφανίσουμε στην συγκεκριμένη θέση και εν συνεχεία ελέγχει την ύπαρξή του στον φάκελο blocks. Σε περίπτωση που υπάρχει, το φορτώνει, διαφορετικά τυπώνει σχετικό μήνυμα λάθους.

Εάν καλέσουμε για παράδειγμα την εντολή **dkInit('database','ateiuser')** θα εκτελεστούν τα κάτωθι:

**Database :**

```
dkSetInit('database', function() {
```

```
    $GLOBALS['DB'] = new PDO (
```

```
    'mysql:host='.Configuration::$DB_Host.';dbname='.Configuration::$DB_Name,
```

```
    Configuration::$DB_User, Configuration::$DB_Pass, array (PDO::ATTR_PERSISTENT =>  
false, PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES \'UTF8\'' ) );
```

```
    $GLOBALS['DB']->setAttribute(PDO::ATTR_ERRMODE,
```

```
    PDO::ERRMODE_WARNING );
```

```
});
```

Όπως βλέπουμε, εκτελείται σύνδεση με την βάση δεδομένων **MySQL** μέσω του **PDO Driver** της PHP, με παραμέτρους από το **object Configuration** το οποίο βρίσκεται στο **configuration.php** και όπως βλέπουμε στις πρώτες γραμμές του **kernel.php**, συμπεριλαμβάνεται (require).

**AteiUser:**

```
dkSetInit('ateiuser', function() {
```

```
require 'kernel/classes/AteiUser.class.php';
```

```
dkGlobal('AteiUser', new AteiUser());
```

```
});
```

Όπως βλέπουμε, εκτελείται **require** η οποία συμπεριλαμβάνει στην εκτέλεση της σελίδας τον κώδικα που βρίσκεται στο αρχείο **AteiUser.class.php** ενώ στην συνέχεια εκτελεί την κλάση που εμπεριέχεται στον κώδικα αυτό, και αποθηκεύει το instance της στην global μεταβλητή **AteiUser** για περαιτέρω χρήση.

Για παράδειγμα, ο κώδικας του αρχείου **AteiUser.class**, ο οποίος είναι υπεύθυνος για της ενέργειες που αφορούν τον πίνακα της βάσης **user** και συνεπώς για ό,τι αφορά τις υπηρεσίες μελών του portal (εγγραφή χρήστη, σύνδεση στον λογαριασμό του, επιβεβαίωση λογαριασμού κτλ), είναι ο ακόλουθος:

```
<?php
```

```
class AteiUser {
```

```
    public function __construct() {
```

```
        dkInit('database');
```

```
        session_start();
```

```
    }
```

```
    public function Authenticate($username, $password) {
```

```
        $autps = $GLOBALS['DB']->prepare('SELECT * FROM users WHERE  
username=:username AND password=:password');
```

```
        $autps->execute(Array(':username'=>$username,':password'=>$password));
```

```
        if($res = $autps->fetch(PDO::FETCH_OBJ)) {
```

```
            $_SESSION['_Logged_User'] = $res;
```

```
            return true;
```

```
        } else {
```

```
return false;

$_SESSION['_Logged_User'] = false;

}

}

public function IsLoggedIn() {

    if($_SESSION['_Logged_User']) return true;

    else return false;

}

public function IsAdmin() {

    if($_SESSION['_Logged_User'] && $_SESSION['_Logged_User']->is_admin == 1)
return true;

    else return false;

}

public function GetAuthenticatedUser() {

    return $_SESSION['_Logged_User'];

}

public function GetUserInfoByUsername($username) {

    $gups = $GLOBALS['DB']->prepare('SELECT * FROM users WHERE
username=:username');

    $gups->execute(Array(':username'=>$username));

    $res = $gups->fetchAll(PDO::FETCH_OBJ);

    if(sizeof($res) > 0) {

        return $res;

    }

}
```

```
    } else {  
  
        return false;  
  
    }  
  
}  
  
public function GetUserInfoByEmail($username) {  
  
    $gups = $GLOBALS['DB']->prepare('SELECT * FROM users WHERE  
email=:email');  
  
    $gups->execute(Array(':email'=>$email));  
  
    $res = $gups->fetchAll(PDO::FETCH_OBJ);  
  
    if(sizeof($res) > 0) {  
  
        return $res;  
  
    } else {  
  
        return false;  
  
    }  
  
}  
  
public function RegisterUser($userarr, $callback=null) {  
  
    extract($userarr);  
  
    if(!$username || !$password || !$email || !$fullname) return  
array('result'=>false,'reason'=>'MISSING_PARAMS');  
  
    if($this->GetUserInfoByUsername($username) return  
array('result'=>false,'reason'=>'USERNAME_EXISTS');  
  
    if($this->GetUserInfoByEmail($email) return  
array('result'=>false,'reason'=>'EMAIL_EXISTS');  
  
    if(!$is_admin) $is_admin = 0;
```

```
if(!$active_flag) $active_flag = 0;

$rgups = $GLOBALS['DB']->prepare('INSERT INTO users (username, password,
email, fullname, is_admin, active_flag)

VALUES (:username, :password, :email, :fullname, :is_admin, :active_flag)');

$rgups->execute(array(':username'=>$username,           ':password'=>$password,
'email'=>$email, ':fullname'=>$fullname,

':is_admin'=>$is_admin, ':active_flag'=>$active_flag));

if($insertID = $GLOBALS['DB']->lastInsertId()) {

    if(is_callable($callback)) $res = $callback($insertID, $userarr);

    return array('result'=>true,'callback_result'=>$res);

} else return array('result'=>false,'reason'=>'DATABASE_PROBLEM');

}

public function UpdateActiveFlag($userid,$active_flag) {

    $supafps = $GLOBALS['DB']->prepare('UPDATE users SET active_flag =
:active_flag WHERE id = :userid');

    $supafps->execute(array(':userid'=>$userid, ':active_flag'=>$active_flag));

    if($supafps->rowCount() > 0) return true;

    else return false;

}

public function UpdateUserDetails($userid, $userarr, $callback=null) {

    extract($userarr);

    if(!$username || !$password || !$email || !$fullname) return
array('result'=>false,'reason'=>'MISSING_PARAMS');
```

```
        if($this->GetUserInfoByUsername($username))                                return
array('result'=>false,'reason'=>'USERNAME_EXISTS');

        if($this->GetUserInfoByEmail($email))                                    return
array('result'=>false,'reason'=>'EMAIL_EXISTS');

        if(!$is_admin) $is_admin = 0;

        if(!$active_flag) $active_flag = 0;

        $rgups = $GLOBALS['DB']->prepare('UPDATE users SET username=:username,
password=:password,          email=:email,          fullname=:fullname,          is_admin=:is_admin,
active_flag=:active_flag WHERE id=:userid');

        $rgups->execute(array(':userid'=>$userid,':username'=>$username,
':password'=>$password, ':email'=>$email, ':fullname'=>$fullname,
':is_admin'=>$is_admin, ':active_flag'=>$active_flag));

        if($GLOBALS['DB']->rowCount() {

                if(is_callable($callback)) $res = $callback($userarr);

                return array('result'=>true,'callback_result'=>$res);

        } else return array('result'=>false,'reason'=>'DATABASE_PROBLEM');

    }

}

?>
```

Ένα ακόμα παράδειγμα πηγαίου κώδικα, είναι οι δημοσκοπήσεις (polls) οι οποίες βασίζονται σε 2 αρχεία: στο AteiPoll.class.php που περιλαμβάνει την βασική κλάση, και στο block poll το οποίο εμφανίζει την εκάστοτε δημοσκόπηση.

Το αρχείο AteiPoll.class.php περιέχει τα ακόλουθα:

```
class AteiPoll {
```



```
public function __construct() {  
  
    dkInit('database');  
  
}  
  
public function getLastQuestion() {  
  
    $qsps = $GLOBALS['DB']->prepare('SELECT * FROM poll_questions WHERE  
publish_date < NOW() ORDER BY publish_date DESC LIMIT 1');  
  
    $qsps->execute();  
  
    if($last_question = $qsps->fetch(PDO::FETCH_ASSOC)) return $last_question;  
  
    return false;  
  
}  
  
public function getOptions($ques_id) {  
  
    $qsps = $GLOBALS['DB']->prepare('SELECT * FROM poll_options WHERE  
ques_id = :ques_id');  
  
    $qsps->execute(Array(':ques_id'=>$ques_id));  
  
    if($options = $qsps->fetchAll(PDO::FETCH_ASSOC)) return $options;  
  
    return false;  
  
}  
  
public function voteOption($ques_id,$option_id,$voted_on,$ip) {  
  
    $qsps = $GLOBALS['DB']->prepare('INSERT INTO poll_votes  
(option_id,voted_on,ip) VALUES(:option_id,:voted_on,:ip)');  
  
    $qsps->execute(Array(':option_id'=>$option_id,':voted_on'=>$voted_on,':ip'=>$ip));  
  
    setcookie("ateipoll_". $ques_id, 'yes', time()+86400*300);  
  
    return true;  
  
}
```

```
}

public function getResults($ques_id) {

    $rstps = $GLOBALS['DB']->prepare("SELECT COUNT(*) as totalvotes FROM
poll_votes WHERE option_id IN(SELECT id FROM poll_options WHERE ques_id=:ques_id)");

    $rstps->execute(array(':ques_id'=>$ques_id));

    $Res0 = $rstps->fetch();

    $total_votes = $Res0['totalvotes'];

    $rsps = $GLOBALS['DB']->prepare("SELECT poll_options.id, poll_options.value,
COUNT(*) as votes FROM poll_votes, poll_options WHERE poll_votes.option_id=poll_options.id
AND poll_votes.option_id IN(SELECT id FROM poll_options WHERE ques_id=:ques_id) GROUP
BY poll_votes.option_id");

    $rsps->execute(array(':ques_id'=>$ques_id));

    while($Res1 = $rsps->fetch(PDO::FETCH_ASSOC)) {

        $percent=round(($Res1['votes']*100)/$total_votes);

        $value=$Res1['value'];

        $vote_details[] = Array('value'=>$value,'percent'=>$percent);

    }

    return array('total_votes'=>$total_votes, 'details'=>$vote_details);

}

}
```

Ενώ το **poll.php** περιέχει τα ακόλουθα (δίνουμε βάση στο `dkInit` το οποίο αποτελεί τον συνδετικό κρίκο αυτού του αρχείου με το **AteiPoll.class.php** μέσω του `kernel.php`:

```
<?php
dkInit('ateipoll');

$AteiPoll = dkGlobal('AteiPoll');

$LastQuestion = $AteiPoll->getLastQuestion();

if($_POST['send_poll'] == 'true' && $_POST['option_value']) {

    $AteiPoll->voteOption($_POST['ques_id'],$_POST['option_value'],date('Y-m-d
H:i:s'),$_SERVER['REMOTE_ADDR']);

}

?>

<div class="poll">

<span><?=$LastQuestion['ques']; ?></span>

<?

if($_POST['send_poll'] == 'true' || $_COOKIE["ateipoll_".$LastQuestion['id']]=='yes') {

    $row = $AteiPoll->getResults($LastQuestion['id']);

    foreach($row['details'] as $r):

        echo '<div class="option" ><p>'.$r['value'].' (<em>'.$r['percent'].'%</em>)</p>';

        echo '<div class="bar '

        echo '" style="width: '.$r['percent'].'%'; " ></div></div>';

    endforeach;

} else {

?>

<form method="POST">

<div class="title"><?=$LastQuestion['ques']; ?></div>
```

```
<div class="options">

<?php

$polloptions = $AteiPoll->getOptions($LastQuestion['id']);

foreach($polloptions as $polloption) {

?>

<input type="radio" name="option_value"
value="<?=$polloption['id'];?>"><?=$polloption['value'];?><br>

<?php } ?>

<input type="hidden" name="ques_id" value="<?=$LastQuestion['id']; ?>" />

<input type="hidden" name="send_poll" value="true" />

</div>

<button type="submit">Ψήφισε</button>

</form>

</div>

<? } ?>
```

**Τέλος**, ακόμα ένα χαρακτηριστικό τμήμα πηγαίου κώδικα του portal είναι και ο κώδικας του template (**template/index.php**) στον οποίο με έντονο χρωματισμό διακρίνονται σημεία που έχουν αναφερθεί στα κεφάλαια της εργασίας:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
    <meta name="author" content="" />
    <meta http-equiv="content-type" content="text/html; charset=utf8" />
    <link rel="stylesheet" href="/template/style/style.css" type="text/css" />
```

```
<title><?=Configuration::$SITE_TITLE;?><?=(dkGlobal('PAGE_TITLE')?' ::
'.dkGlobal('PAGE_TITLE':))?></title>
<?=implode("\n",(array)$GLOBALS['HTML_HEAD']);?>
</head>
<body>
<div class="content">
<div class="header">
<div class="top_info">
<div class="top_info_right">
</div>
<div class="top_info_left">
<?php
<b>loadBlock('eortologio');
?>
</div>
</div>
<div class="logo">
<h1><a href="#" title="Centralized Internet Content">ΑΤΕΙ
Αμαλιάδας</a></h1>
</div>
</div>
<div class="bar">
<ul>
<li class="slogan">Καλωσήρθατε στο Portal</li>
<li><a href="#" class="active">Κεντρική</a></li>
<li>People</li>
<li><a href="#" accesskey="m">Music</a></li>
<li><a href="#" accesskey="p">Pictures</a></li>
<li><a href="#" accesskey="s">Services</a></li>
<li><a href="#" accesskey="r">Products</a></li>
</ul>
</div>
<div class="search_field">
```

```
<form method="post" action="?">
    <p><input type="text" name="search" class="search" /> <input
type="submit" value="Search" class="searchSubmit" /></p>
    </form>
</div>
<div class="subheader">
    <? loadBlock('announcement_lasttop'); ?>
</div>
<div class="main">
    <?=$MAIN_CONTENT;?>
</div>
<div class="footer">

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