



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**

**Σχολή Τεχνολογικών Εφαρμογών**

**Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



**Πτυχιακή Εργασία**

**Τίτλος:**

Ηλεκτρονική Video Λέσχη με δυνατότητα on-line αγορών και ενοικιάσεων ταινιών μέσω ιστοσελίδας (e-shop).

**ΧΑΡΔΑΛΑΣ ΕΜΜΑΝΟΥΗΛ (ΑΜ: 1365)**

**Επιβλέπων Καθηγητής: ΓΙΩΡΓΟΣ Μ. ΠΑΠΑΔΟΥΡΑΚΗΣ Ph.D.**

**Επιτροπή Παρουσίασης:**

**Ημερομηνία Παρουσίασης:**

## **Ευχαριστίες**

Ευχαριστώ τον Καθηγητή-Κύριο Γιώργο Παπαδουράκη για την ανάθεση της παρούσας πτυχιακής εργασίας μέσω της οποίας ανδρώθηκα στον κόσμο του προγραμματισμού διαδικτύου και του Drupal, ενός συστήματος διαχείρισης περιεχομένου από προγραμματιστές για προγραμματιστές. Τον ευχαριστώ επίσης και για το μεγάλης σπουδαιότητας και πάντα επίκαιρο μάθημα, την Ψηφιακή Επεξεργασία Σήματος, που διδακτικά από τον ίδιο.

## **Abstract**

The purpose of this thesis is the analysis the design the implementation and the operation of a web application, an online store. For this purpose we studied existing proposals in the web, from where, a few utilized the latest techniques in object and a few other set some design needs for more functional online stores.

For the construction of the electronic shop is been used the open source CMS (Content Management System) Drupal, which is based on the Server side programming language PHP and the open source database mySQL. To extend the functionality and the applications performance, as it follows from the requirements of the web and the system to be somewhat competitive, were used for further technologies such as the formatting of the content, CSS, but also the client side programming language JavaScript.

The application that was developed on, with the subject on-line buying and renting movies, separates registered users or not. Each category of users has different rights but others are common. Each user has the ability to see the published contents of the application and use the functions it provides, such as to classify and search content dynamically, to read the forum topics and see ratings, but, for making buys rentals forum comments and content rating is necessary the user to be registered in the system.

## Σύνοψη

Σκοπός της πτυχιακής αυτής είναι η ανάλυση η σχεδίαση η υλοποίηση και η λειτουργία μιας διαδικτυακής εφαρμογής, ενός ηλεκτρονικού καταστήματος. Για το σκοπό αυτό μελετήθηκαν ήδη υπάρχουσες προτάσεις στο χώρο, απ' όπου και, από μερικές αξιοποιήθηκαν οι πιο σύγχρονες τεχνικές στο αντικείμενο και από μερικές άλλες καθορίστηκαν ανάγκες σχεδίασης πιο λειτουργικών ηλεκτρονικών καταστημάτων.

Για την κατασκευή του ηλεκτρονικού καταστήματος χρησιμοποιήθηκε το ανοιχτού κώδικα Σύστημα Διαχείρισης Περιεχομένου (Content Management System) Drupal και το επίσης ανοιχτού κώδικα e-commerce shopping cart το Ubercart, τα οποία βασίζονται στην Server side γλώσσα προγραμματισμού PHP και χρησιμοποιούν την ανοιχτού κώδικα βάση δεδομένων MySQL. Για την επέκταση της λειτουργικότητας και της απόδοσης της εφαρμογής, όπως προκύπτει από τις απαιτήσεις του χώρου και του συστήματος ώστε να είναι κατά τι ανταγωνιστικό, χρησιμοποιήθηκαν περαιτέρω τεχνολογίες όπως αυτή της μορφοποίησης του περιεχομένου, CSS, αλλά και της client side γλώσσας προγραμματισμού JavaScript.

Η εφαρμογή που αναπτύχθηκε, με θέμα την on-line αγορά και ενοικίαση ταινιών, διαχωρίζει τους χρηστές σε εγγεγραμμένους και μη. Κάθε κατηγορία χρηστών έχει διαφορετικά δικαιώματα ενώ κάποια αλλά είναι κοινά. Κάθε χρήστης έχει την δυνατότητα να δει το δημοσιευμένο περιεχόμενο της εφαρμογής και να χρησιμοποιήσει τις λειτουργίες που παρέχει, όπως το να ταξινομήσει και να αναζητήσει το περιεχόμενο δυναμικά, να διαβάσει θέματα του forum και να δει τις βαθμολογίες των προϊόντων, αλλά, για την πραγματοποίηση αγορών, ενοικιάσεων, τον σχολιασμό στο forum και την βαθμολόγηση του περιεχομένου είναι απαραίτητη η εγγραφή του χρηστή στο σύστημα.

## Πίνακας περιεχομένων

<b>Abstract</b> .....	<b>2</b>
<b>Σύνοψη</b> .....	<b>3</b>
<b>Πίνακας εικόνων</b> .....	<b>6</b>
<b>Λίστα πινάκων</b> .....	<b>7</b>
<b>1 Εισαγωγή</b> .....	<b>8</b>
1.1 Περίληψη.....	9
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	10
1.3 Σκοπός και στόχοι της εργασίας.....	10
1.4 Δομή εργασίας.....	10
<b>2 Μεθοδολογία υλοποίησης</b> .....	<b>11</b>
2.1 Μέθοδος ανάλυσης & ανάπτυξης πτυχιακής.....	12
2.1.1 Αλγόριθμοι.....	13
2.1.2 Θεωρίες.....	13
<b>3 Σχέδιο δράσης για την εκπόνηση της εργασίας</b> .....	<b>15</b>
3.1 <i>State of the art</i> .....	15
3.1.1 Εισαγωγή στο Σύστημα Διαχείρισης Περιεχομένου.....	15
3.1.2 Ιστορικά στοιχεία.....	16
3.1.3 Ορισμός-Περιγραφή.....	16
3.1.4 Γενικά και Εξειδικευμένα Πλεονεκτήματα.....	17
3.1.5 Βασικά Χαρακτηριστικά.....	19
3.1.6 Είδη Συστημάτων Διαχείρισης Περιεχομένου.....	21
3.1.7 Κριτήρια Επιλογής CMS.....	22
3.1.8 Εισαγωγή στο Drupal.....	24
3.1.9 Βασικές έννοιες στο Drupal.....	28
3.1.10 Η στοίβα τεχνολογίας του Drupal και ο πυρήνας του- <i>Technology Stack and the core</i> .....	31
3.1.11 Μονάδες και Hook system στο Drupal.....	32
3.1.12 Το Drupal API.....	33
3.1.13 Τι είναι θέμα και τι Μηχανή παραγωγής θεμάτων- <i>Themes and Theme Engine</i> .....	34
3.1.14 Το σύστημα αρχείων του Drupal.....	37
3.1.15 Η διαδικασία εξυπηρέτησης ενός αιτήματος από το Drupal- <i>Serving a request</i> .....	39
3.1.16 Το σύστημα πλοήγησης- <i>The Menu System</i> .....	40
3.1.17 Το abstract επίπεδο της βάσης δεδομένων του Drupal- <i>The database abstract layer</i> .....	41
3.1.18 Συνεδρίες- <i>Sessions</i> .....	42
3.1.19 Το Form API του Drupal.....	45
<b>4 Κόριο μέρος πτυχιακής</b> .....	<b>47</b>
4.1 Ανάλυση προβλήματος.....	47
4.1.2 Απαιτήσεις συστήματος.....	49
4.2 Σχεδιασμός υλοποίησης.....	50
4.3 Υλοποίηση.....	54
4.3.1 Υλοποίηση με Views.....	60
4.3.2 Δυο λόγια για το CSS και τον Firebug.....	72

4.3.3	Υλοποίηση της πλήρους μορφής του προϊόντος με τη χρήση των <i>template</i> αρχείων.....	74
4.3.4	Υλοποίηση της <i>custom</i> μονάδας <i>book_form</i> .....	82
<b>5</b>	<b>Αποτελέσματα.....</b>	<b>86</b>
5.1	Συμπεράσματα.....	86
5.2	Μελλοντική εργασία και επεκτάσεις.....	86
5.3	Βιβλιογραφία .....	87
<b>6</b>	<b>Παράρτημα .....</b>	<b>90</b>
6.1	Παρουσίαση .....	90

## Πίνακας εικόνων

<b>Εικόνα 1:</b> Κεφαλίδα απάντησης της HTTP με το PHPSESSID προς τον browser.....	43
<b>Εικόνα 2:</b> Αίτηση HTTP προς τον server με το PHPSESSID που αποδόθηκε από πριν.....	43
<b>Εικόνα 3:</b> XAMPP.....	49
<b>Εικόνα 4:</b> WinSCP–PuTTY.....	49
<b>Εικόνα 5:</b> Λίστα με τα δικαιώματα για κάθε χρήστη.....	54
<b>Εικόνα 6:</b> Δημιουργία κλάσης movie.....	55
<b>Εικόνα 7:</b> Πεδία του βασικού τύπου περιεχομένου movie.....	55
<b>Εικόνα 8:</b> Λεξικά που συνδέονται με τον τύπο περιεχομένου movie.....	56
<b>Εικόνα 9:</b> Πεδία τίτλος, περιγραφή και επιπρόσθετα πεδία από το CCK.....	57
<b>Εικόνα 10:</b> Πεδία λεξικών που συνδέονται με τον τύπο movie.....	57
<b>Εικόνα 11:</b> Επιπλέον πεδία του τύπου movie.....	58
<b>Εικόνα 12:</b> Παρεχόμενα πεδία από την κλάση του Ubercart.....	59
<b>Εικόνα 13:</b> Το User Interface των Views.....	59
<b>Εικόνα 14:</b> Οι custom Views της εργασίας.....	60
<b>Εικόνα 15:</b> Η view Blu-ray.....	61
<b>Εικόνα 16:</b> Δημιουργία στοιχείου μενού.....	62
<b>Εικόνα 17:</b> Ορισμός κλάσεων για τα πεδία της view Blu-ray.....	63
<b>Εικόνα 18:</b> Κώδικας CSS για τα πεδία της view Blu-ray.....	63
<b>Εικόνα 19:</b> Τελικό αποτέλεσμα της view Blu-ray.....	64
<b>Εικόνα 20:</b> Η view Movies_categories.....	65
<b>Εικόνα 21:</b> Το block της view Movies_categories.....	66
<b>Εικόνα 22:</b> Τελικό αποτέλεσμα της view Movies_categories.....	66
<b>Εικόνα 23:</b> Η view The_front_page.....	67
<b>Εικόνα 24:</b> Το Slideshow της view The_front_page.....	67
<b>Εικόνα 25:</b> Ο CSS κώδικας για το Slideshow display.....	68
<b>Εικόνα 26:</b> Ο CSS κώδικας για τα display Page και Offers.....	69
<b>Εικόνα 27:</b> Τελικό αποτέλεσμα της view The_front_page.....	70
<b>Εικόνα 28:</b> Η view Top_rated_movies1.....	71
<b>Εικόνα 29:</b> Τελικό αποτέλεσμα της view Top_rated_movies.....	72
<b>Εικόνα 30:</b> Το add on Firebug του Firefox.....	73
<b>Εικόνα 31:</b> Ένα τυπικό Web template system.....	74
<b>Εικόνα 32:</b> Το region.tpl.php template του Drupal.....	75
<b>Εικόνα 33:</b> Τα πεδία του τύπου movie σε μορφή πίνακα.....	76
<b>Εικόνα 34:</b> Πρώτο τμήμα κώδικα του αρχείου movie-node.tpl.php.....	77
<b>Εικόνα 35:</b> Η συνάρτηση rawurlencode() της PHP.....	78
<b>Εικόνα 36:</b> Δεύτερο τμήμα κώδικα του αρχείου movie-node.tpl.php.....	79
<b>Εικόνα 37:</b> Ο κώδικας CSS για το movie-node.tpl.php.....	80
<b>Εικόνα 38:</b> Τελικό αποτέλεσμα του movie-node.tpl.php.....	81
<b>Εικόνα 39:</b> Αρχείο .info της custom μονάδας book_form.....	82
<b>Εικόνα 40:</b> Πρώτο τμήμα κώδικα μονάδας book_form.....	83
<b>Εικόνα 41:</b> Δεύτερο τμήμα κώδικα μονάδας book_form.....	84
<b>Εικόνα 42:</b> Τελικό αποτέλεσμα της μονάδας book_form.....	85
<b>Εικόνα 43:</b> Φόρμα επικοινωνίας του eVideoclub.....	85

## Λίστα πινάκων

<b>Σχήμα 1:</b> Τυπικό σύστημα ηλεκτρονικού καταστήματος.....	11
<b>Πινάκας 1:</b> Σχετικές μεθοδολογίες υλοποίησης.....	12
<b>Σχήμα 2:</b> Η βάση της λειτουργίας όλων των CMS.....	16
<b>Σχήμα 3:</b> Τυπικό διάγραμμα ροής εργασίας σε ένα CMS .....	20
<b>Σχήμα 4:</b> Εκδόσεις του Drupal .....	24
<b>Σχήμα 5:</b> Τα πέντε βασικά επίπεδα του Drupal .....	26
<b>Σχήμα 6:</b> Drupal's Stack Technology.....	31
<b>Σχήμα 7:</b> Σχέση μονάδων πυρήνα, custom, contributed μονάδων και hook .....	32
<b>Σχήμα 8:</b> Σχέση μονάδων πυρήνα, custom και community μονάδων και API .....	33
<b>Σχήμα 9:</b> Στάδια theming του περιεχομένου μέχρι την παρουσίαση του .....	35
<b>Σχήμα 10:</b> Σύλλογή αρχείων σε ένα τυπικό Drupal theme .....	36
<b>Σχήμα 11:</b> Διασύνδεση ενός βασικού θέματος με δύο υποθέματα .....	37
<b>Σχήμα 12:</b> Σύστημα αρχείων και καταλόγων του Drupal.....	38
<b>Σχήμα 13:</b> Ο κατάλογος sites του Drupal.....	38
<b>Σχήμα 14:</b> Επισκόπηση της διαδικασίας αποστολής του μενού.....	40
<b>Σχήμα 15:</b> Καθορισμός του database αρχείου που θα περιληφθεί με βάση την τιμή της \$db_url.....	42
<b>Σχήμα 16:</b> Κύκλος ζωής μιας συνεδρίας στο Drupal -Session life cycle .....	44
<b>Σχήμα 17:</b> Πως το Drupal χειρίζεται τις φόρμες.....	46
<b>Σχήμα 18:</b> Σχέση δεδομένων πληροφοριών.....	47
<b>Σχήμα 19:</b> Δικαιώματα εγγεγραμμένου χρήστη και μη.....	50
<b>Σχήμα 20:</b> Δικαιώματα administrator και webmaster.....	51
<b>Σχήμα 21:</b> Διαδικασία παραγγελίας πιστοποιημένου χρήστη.....	52
<b>Σχήμα 22:</b> Κατηγοριοποίηση προϊόντων ανά χαρακτηριστικό.....	53
<b>Σχήμα 23:</b> Πεδία βασικού τύπου περιεχομένου movie.....	53



## 1. Εισαγωγή

Ηλεκτρονικό κατάστημα (e-shop) είναι ο όρος που χρησιμοποιείται για να αναφερθεί κάποιος σε ένα Διαδικτυακό τόπο μέσω του οποίου πραγματοποιούνται πωλήσεις διαφόρων ειδών. Η Επανάσταση της Πληροφορικής άλλαξε σημαντικά τον τρόπο ζωής των πολιτών, επιφέροντας μια σειρά αλλαγών, που επηρεάζουν και τις εμπορικές επιχειρήσεις.

Αυτές οι αλλαγές μπορούν να αποτελέσουν σημαντικό όπλο στα χέρια των επιχειρήσεων που θέλουν να ανταποκριθούν στις απαιτήσεις ενός νέου περιβάλλοντος που χαρακτηρίζεται από συνεχώς μεταβαλλόμενες συνθήκες, διεθνοποίηση και εντατικοποίηση του ανταγωνισμού, κ.α. Οι επιχειρήσεις που θα "επιβιώσουν" στον ανταγωνισμό είναι αυτές που στον παρόντα χρόνο θα κάνουν τις στρατηγικές επιλογές για την ενσωμάτωση των νέων τεχνολογιών στις πρακτικές τους. Παλαιότερα, η ενσωμάτωση αυτή περιλάμβανε μόνο την ηλεκτρονική παρουσίαση των καταστημάτων, όχι όμως και όλων των ειδών τους και, πολύ περισσότερο, δεν υπήρχε δυνατότητα άμεσης παραγγελίας κάποιου είδους. Το ηλεκτρονικό επιχειρεί (E-Business) αναφέρεται στην πραγματοποίηση επιχειρηματικών συναλλαγών μέσω του [Internet](#) και είναι η προσαρμογή του κλασικού επιχειρηματικού μοντέλου στην νέα ηλεκτρονική πραγματικότητα ή την ανάπτυξη νέου επιχειρηματικού μοντέλου με αντικείμενο μόνο το Διαδίκτυο.

Έτσι, αναπτύσσονται διεθνώς τα ηλεκτρονικά καταστήματα, που προσφέρουν ημερησίως χιλιάδες προϊόντα που υπόσχονται χαμηλότερες τιμές. Ανάλογα με τα προσφερόμενα είδη, ο μελλοντικός πελάτης μπορεί να αναζητήσει ανάμεσα σε πολλά ομοειδή το συγκεκριμένο είδος που επιθυμεί, να μάθει την τιμή και τον χρόνο αποστολής (εάν το παραγγείλει), να το δει σε εικόνες (ορισμένες φορές και σε βίντεο) και να κάνει και σχετικές συγκρίσεις τιμών. Οι τιμές στα ηλεκτρονικά καταστήματα είναι φθηνότερες, γιατί ένα τέτοιο κατάστημα δεν διατηρεί σημεία πώλησης με υψηλό ενοίκιο, δεν απασχολεί αριθμητικά το ίδιο προσωπικό με ένα συμβατικό και παραμένει "ανοικτό" σε 24ωρη βάση και για 365 μέρες ετησίως. Ο μέλλον πελάτης μπορεί ακόμη να βρει και να παραγγείλει είδη που δεν υπάρχουν στα συμβατικά καταστήματα της πόλεως ή της χώρας του και μπορεί να πληρώσει μέσω της πιστωτικής του κάρτας ή με την χρήση της αντικαταβολής ή [Paypal](#).

## 1.1 Περίληψη

Στην παρούσα πτυχιακή εργασία καταγράφονται όλα τα βήματα που ακολουθηθήκαν για την ανάλυση τη σχεδίαση την υλοποίηση και την λειτουργία ενός ηλεκτρονικού καταστήματος. Περιληπτικά τα στάδια ανάπτυξης του λογισμικού περιγράφονται ως εξής.

Από την ανάλυση των απαιτήσεων πρόεκυαν οι παρακάτω ανάγκες.

Από την πλευρά του χρηστή (front end).

- 1) Ασφαλή είσοδος/έξοδος στο/από το σύστημα.
- 2) Ασφαλείς συναλλαγές μεταξύ χρηστών και συστήματος.
- 3) Απλότητα στη χρήση.
- 4) Λειτουργικότητα.
- 5) Σαφής διαχωρισμός περιεχομένου.
- 6) Κατηγοριοποίηση του περιεχομένου.
- 7) Ευδιάκριτη μορφοποίηση του περιεχομένου.
- 8) Απλή και σύνθετη αναζήτηση περιεχομένου.

Από την πλευρά του διαχειριστή (back end).

- 1) Πλήρης διαχείριση-έλεγχος του συστήματος.
- 2) Πλήρης διαχείριση χρηστών.
- 3) Πλήρης διαχείριση παραγγελιών.
- 4) Δυνατότητα ενημέρωσης του συστήματος.
- 5) Απλότητα στην δημιουργία περιεχομένου.
- 6) Δυνατότητα εμπλουτισμού του περιεχομένου.
- 7) Αυτόματη κατηγοριοποίηση περιεχομένου.
- 8) Δυνατότητα ενημέρωσης του περιεχομένου.

Για την σχεδίαση της εφαρμογής πρόεκυψε η απαίτηση της ύπαρξης αντικειμενοστραφούς γλώσσας προγραμματισμού, χαρακτηριστικό που προσφέρει πλήρως η PHP. Κάθε τύπος περιεχομένου θα πρέπει να κατηγοριοποιείται κάτω από μια κλάση, έτσι ώστε να απλοποιείτε η διαχείριση του. Επίσης απαραίτητη είναι η ύπαρξη μιας σχεσιακής βάσης δεδομένων για την αποθήκευση την ανάκληση την ταξινόμηση και γενικά την διαχείριση του περιεχομένου. Τυπικό παράδειγμα αποτελεί η ανοιχτού κώδικα σχεσιακή βάση δεδομένων MySQL. Για την κάλυψη της λειτουργικότητας και της μορφοποίησης του περιεχομένου της εφαρμογής δεν υπάρχει άλλος δρόμος πέραν των τεχνικών JavaScript και CSS.

Τέλος η υλοποίηση της εφαρμογής επιλέχτηκε να γίνει με το ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου Drupal, το οποίο υπερκαλύπτει όλες τις προκαθορισμένες απαιτήσεις κατά την ανάλυση την σχεδίαση την υλοποίηση αλλά και την συντήρηση της εφαρμογής.

## 1.2 Κίνητρο για την διεξαγωγή της εργασίας

Κίνητρο για την διεξαγωγή της εργασίας αυτής αποτέλεσε ο τρόπος κατασκευής μιας web εφαρμογής, ενός ηλεκτρονικού καταστήματος, κάποιου βαθμού δυσκολίας, ο οποίος αυξάνει ανάλογα των υπηρεσιών και των λειτουργιών που προσφέρει η εφαρμογή, από το μηδέν. Ένας οδηγός, δηλαδή, κατασκευής ενός ηλεκτρονικού καταστήματος του οποίου οι απαιτήσεις συμπίπτουν με αυτές της προηγούμενης παραγράφου.

## 1.3 Σκοπός και στόχοι εργασίας

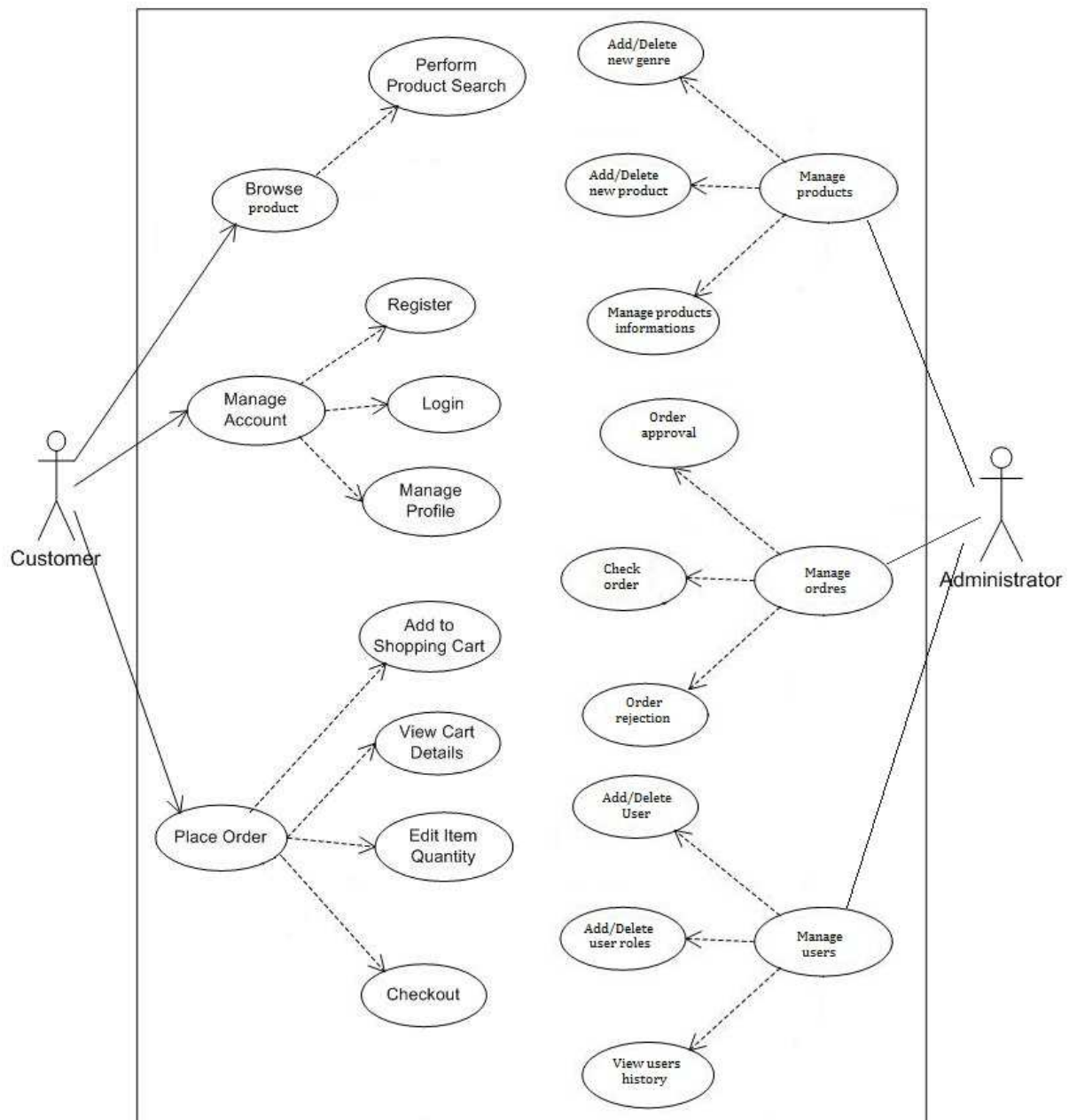
Σκοπός της εργασίας αυτής είναι η κατασκευή μιας web εφαρμογής, ενός ηλεκτρονικού καταστήματος, με τη χρήση των πιο σύγχρονων τεχνολογιών στην επιστήμη της πληροφορικής. Κύριος στόχος της εργασίας αυτής είναι η παροχή υπηρεσιών πώλησης και ενοικίασης προϊόντων μέσω διαδικτύου με έναν απλό, σαφή και αποτελεσματικό τρόπο. Η απλότητα της εφαρμογής στην χρήση αποτελεί έναν άλλο στόχο καθώς επίσης και η κατηγοριοποίηση των προϊόντων ανά είδος. Τέλος, στόχο αποτελεί και η παρουσίαση των διαφόρων τεχνικών, του κώδικα της εφαρμογής, αλλά και του συστήματος διαχείρισης περιεχομένου που χρησιμοποιήθηκαν για την ολοκλήρωση της εργασίας.

## 1.4 Δομή εργασίας

- Στο πρώτο κεφάλαιο γίνεται μια περιληπτική παρουσίαση του θέματος της πτυχιακής εργασίας, των απαιτήσεων της εφαρμογής των τεχνικών και των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της καθώς και τους στόχους και τον σκοπό της εργασίας.
- Στο δεύτερο κεφάλαιο γίνεται μια περιληπτική παρουσίαση των εργαλείων και των τεχνικών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Επίσης περιγράφονται η μέθοδο ανάπτυξης και ανάλυσης της εφαρμογής.
- Στο τρίτο κεφάλαιο περιγράφεται το σχέδιο δράσης που ακολουθήθηκε για την εκπόνηση της πτυχιακής εργασίας και οι σημαντικοί στόχοι. Επίσης περιγράφονται οι εισαγωγικές έννοιες των εργαλείων ανάπτυξης της εφαρμογής που χρησιμοποιήθηκαν καθώς και το απαραίτητο χρονοδιάγραμμα.
- Στο τέταρτο κεφάλαιο παρουσιάζεται το κυρίως μέρος της πτυχιακής εργασίας, δηλαδή η ανάλυση του προβλήματος και των απαιτήσεων του συστήματος ο σχεδιασμός υλοποίησης του συστήματος και η υλοποίηση του.
- Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα της εφαρμογής που αναπτύχθηκε τα συμπεράσματα και τυχόν μελλοντικές επέκτασης της εφαρμογής.
- Στο έκτο κεφάλαιο καταγράφεται η βιβλιογραφία και η πηγές από όπου αντλήθηκαν χρήσιμες πληροφορίες για την ανάλυση τον σχεδιασμό και την υλοποίηση της εφαρμογής.

## 2. Μεθοδολογία υλοποίησης

Ένα τυπικό σύστημα ηλεκτρονικού καταστήματος φαίνεται στο σχήμα 1. Είναι το ίδιο σύστημα που ακολουθεί και το ηλεκτρονικό κατάστημα που υλοποιήσαμε. Οι ενέργειες που μπορούν να πραγματοποιηθούν από την πλευρά του χρήστη είναι η περιήγηση στο περιεχόμενο η δημιουργία λογαριασμού στο σύστημα και η δημιουργία παραγγελίας στο σύστημα. Από την πλευρά του διαχειριστή οι ενέργειες που μπορούν να πραγματοποιηθούν είναι η διαχείριση των προϊόντων των παραγγελιών και των χρηστών.



Σχήμα 1: Τυπικό σύστημα ηλεκτρονικού καταστήματος.

## 2.1 Μέθοδος ανάλυσης και ανάπτυξης της πτυχιακής

Από την ανάλυση της εφαρμογής πρόεκυψε η απαίτηση της ύπαρξης αντικειμενοστραφούς γλώσσας προγραμματισμού, χαρακτηριστικό που προσφέρει πλήρως η PHP. Κάθε τύπος περιεχομένου θα πρέπει να κατηγοριοποιείται κάτω από μια κλάση, έτσι ώστε να απλοποιείτε η διαχείριση του. Επίσης απαραίτητη είναι η ύπαρξη μιας σχεσιακής βάσης δεδομένων για την αποθήκευση την ανάκληση την ταξινόμηση και γενικά την διαχείριση του περιεχομένου. Τυπικό παράδειγμα αποτελεί η ανοιχτού κώδικα σχεσιακή βάση δεδομένων MySQL. Για την κάλυψη της λειτουργικότητας και της μορφοποίησης του περιεχομένου της εφαρμογής δεν υπάρχει άλλος δρόμος πέραν των τεχνικών JavaScript και CSS. Όλα αυτά προσφέρονται ολοκληρωμένα από το σύστημα διαχείρισης περιεχομένου Drupal. Για την υλοποίηση του απαιτητού καλαθιού, shopping cart, που δίνει την δυνατότητα για ηλεκτρονικές αγορές χρησιμοποιήθηκε το Ubercart. Με αλλά λόγια το πρόβλημα που καλούμαστε να επιλύσουμε θα αντιμετωπιστεί με την κατασκευή μιας δυναμικής ιστοσελίδας.

Μεθοδολογία	Αναφορά στην βιβλιογραφία
Drupal	<a href="http://drupal.org/home">http://drupal.org/home</a> <a href="http://el.wikipedia.org/wiki/Drupal">http://el.wikipedia.org/wiki/Drupal</a>
Ubercart	<a href="http://www.ubercart.org/">http://www.ubercart.org/</a>
Modules	<a href="http://drupal.org/project/modules">http://drupal.org/project/modules</a>
Themes	<a href="http://drupal.org/project/themes">http://drupal.org/project/themes</a>
PHP	<a href="http://us2.php.net/">http://us2.php.net/</a>
MySQL	<a href="http://www.mysql.com/">http://www.mysql.com/</a>
JavaScript	<a href="http://www.w3schools.com/js/default.asp">http://www.w3schools.com/js/default.asp</a>
jQuery	<a href="http://jquery.com/">http://jquery.com/</a>
CSS	<a href="http://www.w3schools.com/css/default.asp">http://www.w3schools.com/css/default.asp</a>

**Πινάκας 1: Σχετικές μεθοδολογίες υλοποίησης.**

### 2.1.1 Αλγόριθμοι

#### Taxonomy

Προκειμένου το περιεχόμενο μιας web εφαρμογής να μπορεί να ταξινομηθεί κατά βούληση και με βάση τα στοιχεία ταξινόμησης από τον κάθε χρήστη είναι απαραίτητος ένας αλγόριθμος ταξινόμησης. Το Drupal παρέχει τον αλγόριθμο αυτό μέσω της μονάδας (module) **Taxonomy**. Το taxonomy module είναι ο αλγόριθμος κατηγοριοποίησης περιεχομένου της web εφαρμογής.

### 2.1.2 Θεωρίες

#### Drupal

Το **Drupal** είναι ένα αρθρωτό **σύστημα διαχείρισης περιεχομένου** (Content Management System) ανοικτού/ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού **PHP**. Το Drupal, όπως πολλά σύγχρονα CMS, επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστότοπου και αυτούς που συνεισφέρουν. Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal ορισμένες φορές περιγράφεται ως «υποδομή για εφαρμογές ιστού», καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

#### Ubercart

Το **Ubercart** είναι ένα ανοιχτού κώδικα e-commerce σύστημα διαχείρισης παραγγελιών προϊόντων για ηλεκτρονικά καταστήματα το οποίο είναι και πλήρως συμβατό με το Drupal. Το Ubercart μεγιστοποιεί την ισχύ του πυρήνα του Drupal παρέχοντας την δυνατότητα για οποιοδήποτε ιστότοπο, υλοποιημένο σε Drupal, να μετατραπεί σε ηλεκτρονικό κατάστημα με ένα πλήρες σύστημα διαχείρισης πωλήσεων και μεθόδων πληρωμών κατηγοριοποίηση προϊόντων και ένα λειτουργικό καλάθι αγορών.

#### Modules

Οι μονάδες του Drupal είναι διαφορά addons που συνδέονται στον ήδη υπάρχοντα πυρήνα του. Ως προς την δομή δεν είναι τίποτα παραπάνω από αρχεία κώδικα γραμμένα σε PHP τα οποία ενσωματώνονται με τον υπόλοιπο κώδικα του Drupal. Ως προς την λειτουργικότητα είναι ανεκτίμητα για τον έξυπνο λόγο του ότι επιτρέπουν στο κάθε web developer να επεκτείνει την λειτουργικότητα και την λειτουργία του ήδη υπάρχοντα κώδικα με όποιον τρόπο επιθυμεί. Παραδείγματος χάρι εάν υπάρχει ανάγκη για την δημιουργία κάποιου φόρμας που δεν παρέχεται από τον πυρήνα τότε η κατασκευή μιας μονάδας και η ενσωμάτωσή της στον πυρήνα δίνει την λύση.

#### Themes

Οι θεματικές παραλλαγές του Drupal χρησιμοποιούν τυποποιημένα μορφότυπα που μπορούν να δημιουργηθούν από μηχανές σχεδιασμού θεματικών παραλλαγών ακόμη και εκτός Drupal. Οι περισσότερες θεματικές παραλλαγές είναι γραμμένες στην μηχανή PHPTemplate ή, λιγότερο όμως, στην XTemplate μηχανή. Το σύστημα θεματικών παραλλαγών του Drupal χρησιμοποιεί μια template μηχανή για τον διαχωρισμό των HTML/CSS από την PHP.

## **PHP**

Η PHP είναι μια γενικής χρήσης Server-side γλώσσα προγραμματισμού σχεδιασμένη για Web development και την παραγωγή δυναμικών ιστοσελίδων. Είναι μια από τις πρώτες Server-side γλώσσες προγραμματισμού που μπορεί να ενσωματωθεί σε ένα HTML source αρχείο αντί να καλείται ένα εξωτερικό αρχείο για την επεξεργασία των δεδομένων. Ο κώδικας γίνεται interpret από ένα Web Server με εγκατεστημένη την PHP μονάδα επεξεργασίας, π.χ. Apache, όπου και παράγει την ιστοσελίδα με το αποτέλεσμα που ζητήθηκε.

## **MySQL**

Η MySQL είναι η πιο δημοφιλείς ανοιχτού κώδικα σχεσιακή βάση δεδομένων στον κόσμο, που λειτουργεί ως Server παρέχοντας πολλαπλή πρόσβαση χρηστών σε έναν αριθμό βάσεων δεδομένων. Λειτουργεί κάτω από την όρους την GNU General Public Licence καθώς επίσης και κάτω από διαφορές ιδιοκτήτες συμφωνίες. Συστήματα ανοιχτού κώδικα που απαιτούν μια πλήρη χαρακτηριστικών βάση δεδομένων χρησιμοποιούν την MySQL.

## **JavaScript**

Η JavaScript είναι μια prototype-based γλώσσα προγραμματισμού, δυναμική και weakly-typed που χρησιμοποιεί κλάσης και υποστηρίζει αντικειμενοστρέφεια. Χρησιμοποιείτε κυρίως στην client-side μορφή της όπου υλοποιείτε ως μέρος του φυλλομετρητή με σκοπό να εμπλουτίσει το διεπαφή του χρήστη με το δυναμικό ιστότοπο.

## **jQuery**

Η jQuery είναι μια cross-browser βιβλιοθήκη της JavaScript, σχεδιασμένη να απλοποιεί τον client-side προγραμματισμό της HTML. Η jQuery είναι ανοιχτού κώδικα και κάτω από διπλή άδεια την MIT άδεια χρήσης και την GNU General Public Licence. Η σύνταξη της είναι σχεδιασμένη έτσι ώστε να κάνει ευκολότερη την πλοήγηση σε ένα έγγραφο-αρχείο ,επιλέγοντας DOM στοιχεία, αλλά και για την δημιουργία γραφικών, την διαχείριση events-γεγονότων καθώς και για την ανάπτυξη Ajax εφαρμογών.

## **CSS**

Η CSS είναι μια γλώσσά για την μορφοποίηση φύλλων στυλ, που χρησιμοποιούνται για να περιγράψουν την σημασιολογία που παρουσιάζεται(εμφάνιση σχήμα και διάταξη)σε ένα αρχείο γραμμένο σε μια γλώσσα υπερκείμενου, όπως η HTML. Η πιο κοινή εφαρμογή της είναι στο να δίνει στυλ σε ιστοσελίδες γραμμένες σε HTML και XHTML,αλλά η CSS μπορεί επίσης να εφαρμοστεί και σε κάθε είδος XML αρχείων.

### 3. Σχέδιο δράσης για την εκπόνηση της εργασίας

#### 3.1 State of the Art

##### 3.1.1 Εισαγωγή στο Σύστημα Διαχείρισης Περιεχομένου

Ο ρυθμός ανανέωσης της πληροφορίας μεταβάλλεται συνεχώς από τις αρχές του 21ου αιώνα. Πλοηγός της νέας κοινωνίας της πληροφορίας αποτελεί το Διαδίκτυο, που παρουσιάζει ολοένα και μεγαλύτερη διεισδυτικότητα στα σύγχρονα νοικοκυριά. Οι χιλιάδες χρήστες πλέον απαιτούν «φρέσκες» πληροφορίες, με ανανέωση τουλάχιστον ανά ώρα ή και ανά λεπτά, για παράδειγμα στην περίπτωση παρακολούθησης του χρηματιστηρίου. Η απαίτηση αυτή συνοδεύεται ταυτόχρονα από την ανάγκη για την ύπαρξη μία ευέλικτης πλατφόρμας για την παρουσίαση αυτών των πληροφοριών. Τα δύο αυτά χαρακτηριστικά, σύμφυτα της ανάπτυξης του Διαδικτύου, επηρεάζουν μία μεγάλη γκάμα οργανισμών, όχι απαραίτητα κερδοσκοπικών. Για παράδειγμα, μία ηλεκτρονική εφημερίδα χρειάζεται εξίσου το κοινό της, όπως και μία εμπορική επιχείρηση, για να επιβιώσει αρχικά και για να μπορεί να ασκεί επιρροή στην σύγχρονη πραγματικότητα αφετέρου.

Η αυτοματοποίηση των διαδικασιών δημιουργίας των πληροφοριών, που αποτελούν το περιεχόμενο του Διαδικτύου, δημοσίευσης τους και παρουσίασης τους συνιστά το επόμενο βήμα στις προηγούμενες απαιτήσεις. Ο μεγάλος όγκος της πληροφορίας σε συνδυασμό με την απαιτούμενη τεχνική γνώση δεν επέτρεπε στους οργανισμούς να επιτύχουν την ισορροπία ανάμεσα σε ένα εύχρηστο και ελκυστικό περιβάλλον παρουσίασης και σε ένα συνεχώς ανανεώσιμο περιεχόμενο, που θα τους εξασφάλιζε μία σταθερή βάση επισκεψιμότητας στη ιστοσελίδα τους. Όταν δε έμπαινε και ο παράγοντας του ελέγχου της ροής της πληροφορίας από πολλαπλά άτομα, η κατάσταση γινόταν ακόμη πιο δύσκολη. Αποτέλεσμα ήταν η δημιουργία μεγάλων ιστότοπων με καλή σχεδίαση, αλλά ξεπερασμένο χρονικά περιεχόμενο, ή με κακή σχεδίαση χωρίς μεγάλα περιθώρια ευελιξίας, αλλά με υπέρ-ανανεωμένο περιεχόμενο.

Η έλλειψη τεχνικών γνώσεων από τα στελέχη του οργανισμού οδηγούσε τις επιχειρήσεις σε δημιουργία γραφείων ή σε εκμίσθωση ειδικευμένων εταιριών για την διατήρηση των ιστοσελίδων τους. Εκτός από το φανερό κόστος της κίνησης αυτής, η λύση της δημιουργίας ενός ειδικού γραφείου παρουσίαζε σημαντικά προβλήματα. Λίγα άτομα με τεχνικές γνώσεις επιμερίζονταν τον τεράστιο όγκο των πληροφοριών του ιστοχώρου, ενώ επιμερίζονταν ταυτόχρονα και όλες τις λειτουργίες, από την εύρεση του περιεχομένου, την επεξεργασία του, την δημοσίευση του και την αποθήκευση του για μελλοντική χρήση. Συνέπεια ήταν να μην μπορεί το γραφείο πολλές φορές να διαχειριστεί τον τεράστιο όγκο των πληροφοριών, αυτές να δημοσιεύονται με καθυστέρηση και να μην υπάρχει πολυφωνία και πλούτος περιεχομένου. Ιδιαίτερα, αν η ιστοσελίδα ήταν μεγάλη, τότε πολλές φορές το περιεχόμενο της διαμοιράζονταν σε πολλά γραφεία, με αποτέλεσμα έναν ιστοχώρο με έλλειψη διασύνδεσης και χωρίς πολλές φορές καμία συνοχή.

Το τοπίο λοιπόν ήταν γόνιμο για την δημιουργία των ηλεκτρονικών εργαλείων, που θα έδιναν λύση στο πρόβλημα της επιτυχημένης ηλεκτρονικής παρουσίας των οργανισμών στο διαδίκτυο. Τα CMS επιτρέπουν στους οργανισμούς να δημιουργούν, αλλά και να εισάγουν έτοιμο πολυμεσικό υλικό. Να πιστοποιούν τους χρήστες του συστήματος και να επιμερίζουν ξεχωριστούς ρόλους στον καθένα στον κύκλο της λειτουργίας τους. Επίσης, επιτρέπουν τον προσδιορισμό εργασιών ροής του περιεχομένου, συχνά σε συνδυασμό με την λειτουργία των ειδοποιήσεων τον έλεγχο και την επαναφορά παλαιότερου υλικού της ιστοσελίδας. Το βασικότερο, όμως, χαρακτηριστικό που προσφέρουν είναι η δυνατότητα διαχωρισμού του περιεχομένου από την παρουσίαση της ιστοσελίδας.



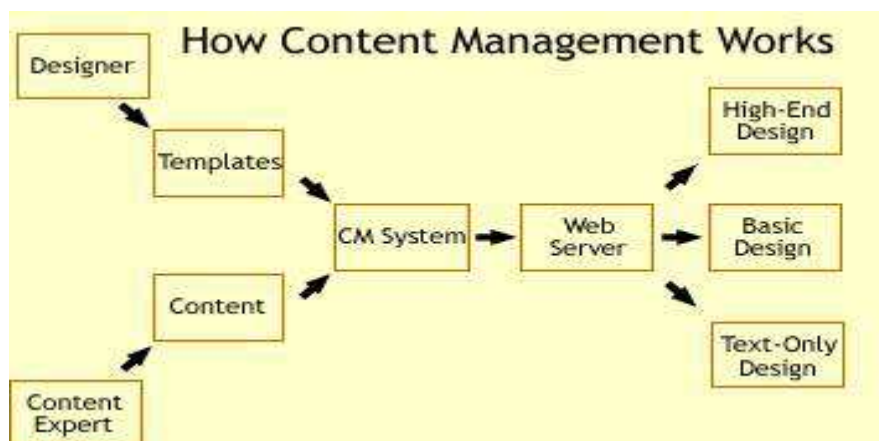
### 3.1.2 Ιστορικά στοιχεία

Ο όρος Content Management System (CMS) αρχικά χρησιμοποιήθηκε για να δηλώσει τα συστήματα δημοσίευσης ιστοσελίδων στο Διαδίκτυο γενικότερα, καθώς επίσης και για τα προγράμματα διαχείρισης περιεχομένου ευρύτερα. Τα πρώτα Συστήματα Διαχείρισης Περιεχομένου αναπτύσσονταν εσωτερικά στους οργανισμούς από το τεχνικό τους τμήμα, καθώς ήταν απαραίτητα για την δημοσιοποίηση ενός μεγάλου όγκου υλικού, από ηλεκτρονικά περιοδικά και εφημερίδες των επιχειρήσεων μέχρι την δημοσίευση και αποστολή των εταιρικών newsletters.

Το 1995, η εταιρία CNET αποφάσισε να επεκτείνει το εσωτερικό σύστημα διαχείρισης περιεχομένου, που χρησιμοποιούσε, για την δημοσίευση ηλεκτρονικού υλικού και να δημιουργήσει την ξεχωριστή εταιρία Vignette. Στόχος της ήταν να εκμεταλλευτεί εμπορικά τα CMS. Στην διάρκεια της δεκαετίας που ακολούθησε η αγορά εξελίχθηκε και σήμερα υπολογίζεται ότι υπάρχουν περί τις 500 εφαρμογές CMS κάθε είδους. Η αγορά εξελίσσεται συνεχώς αναγκάζοντας τους οργανισμούς να ενημερώνονται συνεχώς για τις εξελίξεις και να μετακινούνται στα συστήματα, που πλέον καλύπτουν ακόμη περισσότερο τις ανάγκες τους.

### 3.1.3 Ορισμός-Περιγραφή

Το Content Management System (CMS) είναι μία μορφή λογισμικού για ηλεκτρονικούς υπολογιστές, που αυτοματοποιεί τις διαδικασίες δημιουργίας, οργάνωσης, ελέγχου και δημοσίευσης περιεχομένου σε μία πληθώρα μορφών. Τα περισσότερα CMS έχουν την δυνατότητα να διαχειριστούν περιεχόμενο στις εξής μορφές: κείμενα, εικόνες, βίντεο, Java animation, πρότυπα σχεδίασης, βάσεις δεδομένων κ.α. Πολλές φορές ένα CMS επιτρέπει και την ομαδική δημιουργία κειμένων και άλλου υλικού, για αυτό συχνά χρησιμοποιείται, για παράδειγμα, στα εκπαιδευτικά προγράμματα πολλών εταιριών. Τα CMS χρησιμοποιούνται συχνά και για την αποθήκευση, τον έλεγχο, την διαχείριση και την δημοσίευση εκδόσεων, ο προσανατολισμός των οποίων εξαρτάται από τον φορέα, στον οποίο ανήκει το περιεχόμενο. Έτσι, μπορεί οι εκδόσεις αυτές να αποτελούνται από ειδησεογραφικά άρθρα, εγχειρίδια λειτουργίας, τεχνικά εγχειρίδια, οδηγίες πωλήσεων έως και εμπορικό διαφημιστικό υλικό.



Σχήμα 2: Η βάση της λειτουργίας όλων των CMS

Ένα Web Content Management System ή Web Publishing System είναι η μορφή λογισμικού, που παρέχει επιπρόσθετες δυνατότητες, για την διευκόλυνση των απαραίτητων εργασιών δημοσίευσης ηλεκτρονικού περιεχομένου σε μία ιστοσελίδα. Τα Web CMS έχουν την μεγαλύτερη διείσδυση στους οργανισμούς σήμερα, για αυτό και θα αποτελέσουν τον κορμό της παρούσας εργασίας.

Αποτελούν ένα συνδυασμό μία μεγάλης βάσης δεδομένων, ενός συστήματος αρχειοθέτησης και άλλων στοιχείων λογισμικού, τα οποία χρησιμοποιούνται για την αποθήκευση και την μετέπειτα ανάκτηση των δεδομένων, καθώς επίσης χρησιμοποιούνται και για τις διάφορες ξεχωριστές λειτουργίες των CMS. Αυτά τα συστήματα ηλεκτρονικής δημοσίευσης, λοιπόν, γίνεται φανερό ότι διαφέρουν από τις απλές βάσεις δεδομένων υπό την έννοια ότι μπορούν να καταλογογραφήσουν κείμενο, ηχητικά αποσπάσματα, αποσπάσματα βίντεο ή εικόνες.

Οι χρήστες των Web CMS μπορούν να εντοπίσουν σχετικό υλικό στην βάση δεδομένων, ψάχνοντας με κριτήριο μία λέξη-κλειδί, τον συγγραφέα του κειμένου, την ημερομηνία δημιουργίας του αρχείου κτλ. Έτσι, μπορούν να αποτελέσουν πλέον τα Web CMS μία πύλη πληροφοριών, ή οποία μπορεί να χρησιμεύσει σαν ραχοκοκαλιά για την διαχείριση δεδομένων του ιδιοκτήτη της ιστοσελίδας. Για παράδειγμα, θα μπορούσε να χρησιμεύσει αποθηκεύοντας κάθε άρθρο, που δημοσιεύτηκε σε μία ηλεκτρονική εφημερίδα τα τελευταία τρία χρόνια, και δημιουργώντας ένα ευρετήριο. Έτσι δημιουργεί στην ουσία αυτόματα ένα αρχείο της εφημερίδας εύχρηστο και προσβάσιμο σε κάθε συντάκτη, χωρίς να χρειάζεται να εκτυπώνεται κάθε άρθρο και να διατηρείται ένα ογκώδες και απροσπέλαστο αρχείο.

Ταυτόχρονα, πέρα από τις δυνατότητες σχετικά με την διαχείριση βάσεων δεδομένων, τα λογισμικά αυτά επιτρέπουν στον καθένα να συνεισφέρει πληροφορίες σε μία ιστοσελίδα με την χρήση μίας Γραφικής Διασύνδεσης Χρήστη (Graphical User Interface- GUI). Η διασύνδεση αυτή βασίζεται σε προκατασκευασμένα πρότυπα της ιστοσελίδας και παρέχει μία πλατφόρμα για την εισαγωγή δεδομένων σε κάθε τμήμα της ιστοσελίδας αυτής, χωρίς να είναι απαραίτητη η γνώση εξειδικευμένων γλωσσών προγραμματισμού. Επομένως, μπορούν πλέον οι συντάκτες των ιστοσελίδων να διαχωριστούν από τους τεχνικούς και να εισάγουν απευθείας δεδομένα. Διαχωρίζεται δηλαδή το περιεχόμενο από την παρουσίαση της ιστοσελίδας, καθώς και το καθένα αντιστοιχεί σε διαφορετικού τύπου επεξεργασίας που αποτελεί ένα από τα βασικά πλεονεκτήματα των CMS.

Τα Web CMS μπορούν ακόμη να διανείμουν υλικό σε πελάτες και εταιρικούς συνεργάτες πέρα από τα όρια ενός οργανισμού, παραδείγματος χάρι με την αυτόματη αποστολή newsletters στους πελάτες και την αυτόματη σύνταξη Δελτίων Τύπου και στατιστικών στοιχείων και την ηλεκτρονική αποστολή τους σε συνεργαζόμενες επιχειρήσεις ή ΜΜΕ. Ο πυρήνας, όμως, ενός CMS είναι η διαχείριση του περιεχομένου σε όλο τον κύκλο ζωής της πληροφορίας, δηλαδή από την παραγωγή της μέχρι την δημοσίευση της, αλλά και την μετέπειτα αποθήκευση της.

Τα CMS, επομένως, είναι όλα βασισμένα στην ίδια ιδέα (βλ. σχήμα 2): η διαχείριση περιεχομένου επιτρέπει στους σχεδιαστές να επικεντρωθούν στην σχεδίαση με το χτίσιμο προτύπων (templates). Από την άλλη, οι συντάκτες χτίζουν το περιεχόμενο σε ξεχωριστό περιβάλλον, ο κεντρικός διακομιστής παίρνει το περιεχόμενο, το εισάγει στο σωστό template και το στέλνει όλο μαζί, καθαρά περιτυλιγμένο, στους τελικούς χρήστες.

### **3.1.4 Γενικά και Εξειδικευμένα Πλεονεκτήματα**

Στα γενικά πλεονεκτήματα θα μπορούσαμε να τοποθετήσουμε την μείωση των εξόδων για την διατήρηση μίας ιστοσελίδας και την αύξηση του εισοδήματος χάριν στην επιτυχημένη παρουσία της ιστοσελίδας αυτής. Ακόμη, σημαντικό πλεονέκτημα είναι η δυνατότητα ιεράρχησης και ροής της διαδικασίας δημιουργίας και δημοσίευσης αντικειμένων στην ιστοσελίδα με την χρήση των CMS. Συνεπώς, ιδιαίτερα στην δημιουργία ενός ιστοχώρου, όπου πολλά άτομα θα έχουν πρόσβαση, ώστε να εισάγουν υλικό και να διατηρήσουν ενημερωμένη την ιστοσελίδα, χρειάζεται ένας έλεγχος των σταδίων, που θα ακολουθήσει η πληροφορία για να δημοσιευτεί. Ακόμη, με την βοήθεια των CMS μπορεί να αυξηθεί κατακόρυφα η ποιότητα μίας ιστοσελίδας με την χρήση υψηλής ποιότητας προτύπων σχεδίασης, που θα δίνουν μία εντυπωσιακή εικόνα για τον ιδιοκτήτη τους. Επίσης, τα πρότυπα αυτά μπορούν και να προσδίδουν την ταυτότητα και τον χαρακτήρα του.

Στα γενικά πλεονεκτήματα ενός Συστήματος Διαχείρισης Περιεχομένου, θα μπορούσαν να τοποθετηθούν και οι λιγότερες ανάγκες εκπαίδευσης, που απαιτεί. Με τις έτοιμες φόρμες εισαγωγής, μορφοποίησης και προεπισκόπησης, που προσφέρουν, δεν απαιτούνται πλέον ειδικές γνώσεις προγραμματισμού και σχεδίασης ιστοσελίδων. Με απλές γνώσεις χρήσης ηλεκτρονικών υπολογιστών, που είναι πλέον απαραίτητες στους εργαζόμενους κάθε τομέα της παραγωγής, αλλά και γενικότερα στην καθημερινή ζωή, μπορεί κάποιος να δημιουργήσει ένα εντυπωσιακό, περιεκτικό και ενημερωμένο ιστοχώρο. Σύμφωνα με την προηγούμενη διατύπωση, μία από τις βασικές συνέπειες των CMS θα είναι η μείωση του τεχνικού τμήματος, που απαιτείται για την διαχείριση μίας ιστοσελίδας σε ένα μικρό αριθμό τεχνικών, που θα χρειάζονται για την σωστή λειτουργία και συντήρηση των CMS. Επομένως, αυξάνεται ακόμη περισσότερο το κέρδος από την λειτουργία τους.

Πιο εξειδικευμένα πλεονεκτήματα από την χρήση ενός Content Management System μπορεί να έχει ένας οργανισμός βραχυπρόθεσμα και ανάλογα με το είδος του CMS, που χρησιμοποιεί. Πρώτον, ένας οργανισμός μπορεί να αποκεντρώσει την διατήρηση του περιεχομένου της ιστοσελίδας του, μειώνοντας τις οποιεσδήποτε καθυστερήσεις. Πλέον τα βήματα, που ακολουθούνται, μειώνονται και απλουστεύονται, ενώ η δημιουργία του περιεχομένου μπορεί να διανεμηθεί σε πολλούς. Εξαιτίας της ομοιομορφίας του προτύπου σχεδίασης που προσφέρει το κάθε CMS, πλέον μπορούν να δημιουργηθούν συνεκτικοί, αλλά και πολύ πλούσιοι σε περιεχόμενο ιστοχώροι, αποτέλεσμα της εργασίας πολλών διαφορετικών ανθρώπων και όχι λίγων τεχνικά καταρτισμένων.

Σε ένα δεύτερο επίπεδο, ο διαχωρισμός της λειτουργικότητας και της παρουσίασης της ιστοσελίδας από την δημοσίευση και το περιεχόμενο αντίστοιχα, μπορεί να συνεισφέρει σημαντικά σε έναν οργανισμό. Μπορεί να βοηθήσει στην καλύτερη ιεράρχηση των υπεύθυνων για την δημιουργία και διαχείριση της ιστοσελίδας, καθώς επίσης και στην εστίαση του κάθε υπεύθυνου συγκεκριμένα σε κάποιους τομείς της ιστοσελίδας. Αποτέλεσμα είναι ο καλύτερος καταμερισμός της εργασίας, ώστε να προκύψουν τα μέγιστα δυνατά αποτελέσματα. Για παράδειγμα, ο διευθυντής πωλήσεων σε μία επιχείρηση μπορεί να έχει την δική του ενότητα στην ιστοσελίδα της επιχείρησης, όπου δημοσιεύει τους ισολογισμούς, τους προϋπολογισμούς και τα μελλοντικά επιχειρηματικά σχέδια της επιχείρησης. Από την άλλη, ο υπεύθυνος τύπου της επιχείρησης έχει στην διάθεση του επίσης την δική του ενότητα, ώστε να δημοσιεύει Δελτία Τύπου, ειδήσεις σχετικά με την επιχείρηση, νέες καμπάνιες διαφημιστικές κ.τ.λ.

Αμφότερες αυτές οι ενότητες παρουσιάζουν μία ομοιομορφία, χωρίς να είναι φανερή η διαφορετική ταυτότητα του υπεύθυνου, ενώ στην πρώτη σελίδα υπάρχουν σύντομες καταχωρήσεις με υπερσυνδέσεις προς όλα όσα καταχωρούνται εσωτερικά. Παραδείγματος χάρη, μπορεί να υπάρχει ένα ημερολόγιο στο οποίο μπορεί να προστίθεται αυτόματα η καταχώρηση ενός γεγονότος, όταν αυτό καταχωρείται στην ενότητα του γραφείου τύπου. Έτσι, ένας οργανισμός μπορεί να χρησιμοποιήσει τα καλύτερα στελέχη του για κάθε τμήμα της ιστοσελίδας του, χωρίς να κινδυνεύει η εικόνα της ιστοσελίδας και η λειτουργικότητα της.

Εμφανή είναι και τα οφέλη από την παροχή πληροφοριών σε τακτά χρονικά διαστήματα, βασική δυνατότητα που προσφέρουν τα CMS στους οργανισμούς. Αποτέλεσμα είναι να αυξάνονται θεαματικά οι επισκέπτες, που επισκέπτονται την ιστοσελίδα ή επιστρέφουν σε αυτή, καθώς μέχρι τώρα δεν μπορούσαν να εντοπίσουν τις άμεσες πληροφορίες, που χρειάζονταν. Επιπλέον, εκτός από την αύξηση των επισκεπτών, αυξάνεται και η συχνότητα επισκεψιμότητας της ιστοσελίδας, αφού πλέον ο ίδιος επισκέπτης την επισκέπτεται συχνότερα, για να μπορέσει να βρει νέες πληροφορίες. Σε αυτήν την περίπτωση, ο κερδοσκοπικός οργανισμός κερδίζει από την δημιουργία πολλές φορές ενός μεγάλου πελατολογίου μέσω του Διαδικτύου, ενώ ο μη κερδοσκοπικός οργανισμός από την αύξηση της επιρροής του. Βασικό στοιχείο εδώ είναι ότι σχεδόν όλα τα CMS παρέχουν στατιστικά στοιχεία σχετικά με την επισκεψιμότητα, την συχνότητα επισκεψιμότητας και τις επιλογές των επισκεπτών της ιστοσελίδας.

Επιπροσθέτως, ένα από τα πιο σημαντικά πλεονεκτήματα αποτελεί η δυνατότητα πολλαπλών δημοσιεύσεων της πληροφορίας σε διάφορα κανάλια. Ως εκ τούτου μπορεί ένας οργανισμός να δημοσιεύσει αυτόματα περιεχόμενο σε διάφορα σημεία στην κεντρική σελίδα του, σε διάφορα τμήματα του δικτυακού τόπου του, αλλά πλέον μπορεί πολύ γρήγορα και αυτόματα να δημοσιεύσει υλικό και σε διάφορες συνεργαζόμενες ιστοσελίδες άλλων οργανισμών. Η παγκοσμιοποίηση του 21ου αιώνα και του Διαδικτύου μπορεί να οδηγήσει λόγω του προηγούμενου παραδείγματος σε υψηλές επενδύσεις στις μετοχές τις εταιρίες στην Ελλάδα μετά από την ανάγνωση αυτού του άρθρου διεθνώς.

Η παροχή προσωποποιημένων υπηρεσιών αποτελεί την κορωνίδα στις υπηρεσίες, που προσφέρουν τα CMS, καθώς στον σύγχρονο εξατομικευμένο κόσμο το νέο μοντέλο της πληροφόρησης βασίζεται πλέον στις επιθυμίες και τα ιδιαίτερα χαρακτηριστικά του κάθε ατόμου. Είναι απαραίτητο πια στην ιστοσελίδα κάθε μεγάλου οργανισμού να υπάρχει μία περιοχή μελών, είτε αυτοί αποτελούν τα μέλη του οργανισμού, είτε αποτελούν τους επισκέπτες της ιστοσελίδας, που επιθυμούν να ενημερώνονται σχετικά με τον οργανισμό, να έχουν οικονομικές σχέσεις μαζί του ή να συμμετάσχουν στην βελτιστοποίηση του. Τα CMS παρέχουν την δυνατότητα για την δημιουργία τέτοιων υπηρεσιών, συμβάλλοντας στην παροχή καλύτερων υπηρεσιών του οργανισμού προς τους επισκέπτες - υποψήφιους πελάτες - της ιστοσελίδας του και μεγαλύτερη ικανοποίηση από τα μέλη του είτε εσωτερικά είτε εξωτερικά.

Τέλος, πολύ σημαντικό πλεονέκτημα, που θα έπρεπε να αναπτυχθεί διεξοδικότερα, είναι το ζήτημα του κόστους, που θα εξοικονομήσει ένας οργανισμός από την χρήση των CMS. Ήδη έχει αναφερθεί η μείωση του κόστους, εξαιτίας της μείωσης του τεχνικού προσωπικού, που απαιτείται για την διαχείριση της ιστοσελίδας. Η εξοικονόμηση, όμως για την εταιρία δεν περιορίζεται μόνο σε αυτόν τον τομέα. Επιπροσθέτως, θα μειωθεί το κόστος για την δημιουργία του εταιρικού branding μιας επιχείρησης και των εξόδων μάρκετινγκ, όταν πρόκειται για έναν κερδοσκοπικό οργανισμό, ή των εξόδων διαφήμισης και πρόσβασης σε ενδιαφερόμενους για την πληροφόρησή τους, όταν πρόκειται για μη κερδοσκοπικό οργανισμό. Τα CMS θα βελτιώσουν την παραγωγικότητα του εργατικού δυναμικού του οργανισμού, που σχετίζεται με την διαχείριση της ιστοσελίδας και θα μειώσουν κατακόρυφα τις τεχνικές γνώσεις, που απαιτούνται γενικότερα για την διαχείριση των ιστοσελίδων. Επομένως, θα μειωθούν τα έξοδα για την εκπαίδευση των μελών του οργανισμού και θα αυξηθούν τα οφέλη.

### 3.1.5 Βασικά Χαρακτηριστικά

**Βάση Δεδομένων Περιεχομένου:** πρόκειται για μία βάση δεδομένων, η οποία συγκεντρώνει και ιεραρχεί όλο το περιεχόμενο, το οποίο πρόκειται να δημοσιευτεί στην ιστοσελίδα. Οι λύσεις των Συστημάτων Διαχείρισης Περιεχομένου έχουν, όπως είναι φυσικό, την δυνατότητα να διαχειρίζονται μία πολύ μεγάλη ποικιλία περιεχομένου, καθώς επίσης και διάφορες μορφές του περιεχομένου αυτού. Μπορούν να διαχειρίζονται διάφορες μορφές κειμένων, αρχείων (PDF, Word, Excel, PowerPoint, Zip), άρθρα, Δελτία Τύπου, εικόνες, streaming ήχου και βίντεο, HTML, γραφικά, υπερσυνδέσεις κ.α.

**Βάση Δεδομένων Ατόμων:** πρόκειται για μία βάση δεδομένων όλων των ατόμων που σχετίζονται με την ιστοσελίδα, την οποία διαχειρίζεται το CMS. Αυτά τα άτομα μπορεί να είναι επισκέπτες, μέλη, εγγεγραμμένοι στα newsletters της ιστοσελίδας, εθελοντές κ.α. Σημειώνεται εδώ ότι δεν παρέχουν όλα τα CMS αυτήν την δυνατότητα, καθώς πολλές φορές βασίζονται στην βάση δεδομένων της Εξυπηρέτησης πελατών, με την οποία πολλές φορές τα CMS μπορούν να συνεργαστούν. **Βάση Διαχείρισης Χρηστών:** πρόκειται για μία βάση δεδομένων, που αποτελείται από τα στοιχεία όλων των διαχειριστών και των συντακτών περιεχομένου της ιστοσελίδας, που διαχειρίζεται το CMS. Σε αυτήν την βάση αποθηκεύονται οι κωδικοί των χρηστών αυτών, καθώς επίσης οι συσχετισμένοι ρόλοι τους και τα καθήκοντα τους. Πληροφοριακή Αρχιτεκτονική (Information Architecture - IA): πρόκειται για τον χάρτη πλοήγησης της ιστοσελίδας.

Ένα CMS επιτρέπει στον διαχειριστή του να εγκαταστήσει και να διαχειριστεί την Πληροφοριακή Αρχιτεκτονική και να ρυθμίσει την παρουσίαση των σχετικών μενού πλοήγησης.

Σχεδίαση Παρουσίασης: πρόκειται για την οπτική και την αίσθηση της ιστοσελίδας, όπως αυτές δημιουργούνται μέσω της γραφικής σχεδίασης της. Η παρουσίαση της ιστοσελίδας χωρίζεται σε δύο μέρη: Α) Πλαίσιο: αναφέρεται στην εμφάνιση των δομικών χαρακτηριστικών της σελίδας, όπως για παράδειγμα της κεφαλίδας, του υποσέλιδου, της αριστερής, κεντρικής και δεξιάς στήλης, καθώς επίσης και των κύριων στοιχείων πλοήγησης. Β) Γραφική Προσέγγιση: αναφέρεται στην χρωματική παλέτα, τα είδη και τα μεγέθη των γραμματοσειρών και τα γραφικά στοιχεία, που βρίσκονται σε κοινή χρήση σε όλη την ιστοσελίδα, όπως για παράδειγμα το φόντο. Η παρουσίαση μπορεί να σχεδιαστεί από ένα πρόγραμμα γραφικού σχεδιασμού, το οποίο μπορεί να είναι ενσωματωμένο στο CMS, ή να χρειάζεται χειροκίνητο προγραμματισμό σε γλώσσες, όπως η HTML, CSS και άλλες γλώσσες για την δημιουργία script. Σε κάθε περίπτωση, δημιουργούνται από τους προγραμματιστές του CMS πρότυπα παρουσίασης (packages), οι οποίες μπορούν να εφαρμοστούν σε όλη ή σε μέρος της ιστοσελίδας.

Δημιουργία Περιεχομένου: Εργαλεία φορμών και προγραμματισμού HTML σε μορφή WYSIWYG (Αυτό Που Βλέπεις Είναι Αυτό Που Παίρνεις) βοηθούν τους χρήστες του CMS να προσθέσουν υλικό χωρίς να χρειάζονται να χρησιμοποιήσουν τεχνικούς πόρους. Τα WYSIWYG εργαλεία βοηθούν τους συντάκτες περιεχομένου όχι μόνο να προσθέσουν περιεχόμενο στην ιστοσελίδα χωρίς να χρειάζονται πολλές τεχνικές γνώσεις, αλλά και να παράγουν ένα άρτιας σχεδίασης τελικό προϊόν. Ένα CMS συνήθως περιλαμβάνει μία μεγάλη ποικιλία λειτουργιών, όπως για παράδειγμα: την εισαγωγή γραφικών, μορφοποίηση κειμένου (γραμματοσειρά, μέγεθος, χρώμα, υπογράμμιση, πλάγια κ.α.), δημιουργία πινάκων, ορθογραφικό έλεγχο κ.α. Η λειτουργία προεπισκόπησης επιτρέπει φυσικά στον χρήστη να ελέγξει ξανά το περιεχόμενο μέσα στα πλαίσια της παρουσίασης του, πριν αυτό δημοσιευτεί στην ιστοσελίδα. Αυτό είναι ένα από τα βασικά πλεονεκτήματα των CMS.

Εργαλεία Ροής: πρόκειται για εργαλεία, που βοηθούν την αυτοματοποίηση της διαδικασίας της ροής του περιεχομένου κατά την διάρκεια της παραγωγής του. Σε ένα CMS εμφανίζονται συνήθως τρία στάδια (βλ. σχήμα 3): ο ρόλος του σχεδιαστή, ο ρόλος του συντάκτη/διορθωτή και ο ρόλος του εκδότη, που δίνει την έγκριση του. Μία νέα ή μία αναμορφοποιημένη σελίδα περνάει από κάθε στάδιο, πολλές φορές περισσότερες από μία φορές, μέχρι να δημοσιευτεί. Μικρότεροι ή λιγότερο σύνθετοι οργανισμοί χρησιμοποιούν συχνά μία απλούστερη προσέγγιση ενός σταδίου, για τη δημοσίευση του περιεχομένου. Ο κάθε συντάκτης, δηλαδή, δημοσιεύει ο ίδιος το περιεχόμενο του απευθείας στην ιστοσελίδα, ενσωματώνοντας στο πρόσωπο του και τους τρεις ρόλους.



**Σχήμα 3: Τυπικό διάγραμμα ροής εργασίας σε ένα CMS**

Φόρμες Βάσεων Δεδομένων: πρόκειται για φόρμες, που εμφανίζονται στην δημοσιευμένη ιστοσελίδα και χρησιμοποιούνται για την επί τόπου συγκέντρωση στοιχείων από τους επισκέπτες της. Αυτές οι φόρμες χειρίζονται τις βασικές ανάγκες στην συλλογή δεδομένων, όπως μία σελίδα για την εγγραφή εθελοντών.

Τα περισσότερα CMS προσφέρουν την δυνατότητα σε χρήστες χωρίς πολλές τεχνικές γνώσεις να ρυθμίσουν αυτές τις φόρμες. Εργαλεία Αναζήτησης: πρόκειται για εργαλεία, που επιτρέπουν την αναζήτηση χαρακτηριστικών στοιχείων τόσο σε όλο το μήκος τον δικτυακό τόπο, όσο και σε κάποια συγκεκριμένη περιοχή, που καθορίζεται από τον χρήστη. Επίσης, αφορά τα εργαλεία αναζήτησης κειμένου από τους επισκέπτες της ιστοσελίδας, τα οποία την κάνουν πιο προσβάσιμη και εύχρηστη. Τα καλύτερα εργαλεία αναζήτησης ερευνούν στα κείμενα και στις σελίδες του δικτυακού τόπου και προσφέρουν λειτουργίες σύνθετης αναζήτησης.

Τέλος, πρέπει να σημειωθεί ότι, για να βελτιώσουν τα αποτελέσματα της αναζήτησης, οι διαχειριστές χρησιμοποιούν συχνά ειδικά εργαλεία, τα οποία τεμαχίζουν ή κατηγοριοποιούν τα κείμενα, τα αρχεία και τις εικόνες, διευκολύνοντας με αυτό τον τρόπο την αναζήτηση τους. Εργαλεία Ενσωμάτωσης: πρόκειται για πολύ μικρές εφαρμογές, που υποστηρίζουν την γρήγορη διασύνδεση ανάμεσα στα CMS και στα συστήματα διαχείρισης οικονομικών δεδομένων, όπως είναι, για παράδειγμα, της λογιστικής, της διαχείρισης μελών και δωρεών, των τραπεζικών συναλλαγών και του ηλεκτρονικού εμπορίου.

Αναφορές Ιστοσελίδας: πρόκειται για στατιστικά στοιχεία που συγκεντρώνονται από ένα CMS, ώστε ο διαχειριστής να έχει καλύτερη επίβλεψη. Οι αναφορές αυτές αναλύουν την καθημερινή κίνηση της ιστοσελίδας, τις σελίδες που συνάντησαν το μεγαλύτερο ενδιαφέρον από τους επισκέπτες, την προέλευση των επισκεπτών, την μέση διάρκεια των επισκέψεων στην ιστοσελίδα. Ακόμη, αναφέρουν τον πιο συχνό όρο που αναζητήθηκε από τα εργαλεία αναζήτησης, πια μέθοδος αναζήτησης χρησιμοποιήθηκε περισσότερο, αλλά και άλλα στατιστικά στοιχεία..

Σχεδόν κάθε εργαλείο από τα προηγούμενα είναι διαθέσιμο στα περισσότερα open source CMS. Παρόλα αυτά, η προηγούμενη συλλογή εργαλείων αποτελεί την αρχική σύνθεση ενός CMS, που απευθύνεται σε αρχάριους στον χώρο ή σε οργανισμούς, που επιζητούν μία απλά αξιοπρεπή παρουσία στον χώρο του Διαδικτύου. Περισσότερο εξελιγμένες και σύνθετες λύσεις, καθώς επίσης και χαρακτηριστικά, τα οποία δεν είναι απαραίτητα σε κάθε ιστοσελίδα, παρέχονται από εξειδικευμένα CMS. Τα χαρακτηριστικά αυτά βρίσκονται ενσωματωμένα στο CMS, δηλαδή για τον χειρισμό τους είναι υπεύθυνος και πάλι ο διαχειριστής του προγράμματος, μοιράζονται τον ίδιο πίνακα ελέγχου και έχουν κοινή βάση δεδομένων με τα βασικά χαρακτηριστικά, στα CMS στα οποία προσφέρονται.

### 3.1.6 Είδη Συστημάτων Διαχείρισης Περιεχομένου

Τα Content Management Systems διακρίνονται σε ορισμένες κατηγορίες ανάλογα με ορισμένα βασικά χαρακτηριστικά τα οποία παρουσιάζουν. Μπορούν, λοιπόν, να κατηγοριοποιηθούν ανάλογα με το είδος του παρόχου τους και ανάλογα με το που βρίσκεται ο χώρος αποθήκευσης και διαχείρισης της βάσης δεδομένων και του CMS.

#### ASP και Licensed (με βάση το χώρο αποθήκευσης και διαχείρισης)

Στα Application Service Provider (ASP) CMS, δηλαδή Υποστήριξης Παρόχου Υπηρεσίας, ο κατασκευαστής τους φιλοξενεί όλα τα δεδομένα και το λογισμικό στους server της εταιρίας του. Με αυτόν τον τρόπο απαλείφονται τα έξοδα για μία ακριβή αγορά λογισμικού και hardware του συστήματος, που θα φιλοξενεί το CMS. Παράλληλα μειώνονται και οι ανάγκες για τεχνικούς πόρους, όπως για παράδειγμα για συντηρητές του δικτύου των υπολογιστών. Τέλος, βασικότερο πλεονέκτημα ενός τέτοιου είδους συστήματος είναι η συνεχής εξέλιξη, καθώς ο πάροχος προωθεί διαρκώς νέες λειτουργίες του προϊόντος και ανανεώσεις στους πελάτες του, προσφέροντας έτσι το χαρακτηριστικό της άμεσης ανανέωσης και πρωτοπορίας της ιστοσελίδας.

Στα CMS με παροχή άδειας (Licensed), ο πάροχος του πουλάει το προϊόν, δηλαδή παρέχει άδεια χρήσης του, δεν εμπλέκεται στην όλη διαδικασία λειτουργίας του και ο χρήστης είναι πλέον υπεύθυνος, ώστε να το εγκαταστήσει, να το ρυθμίσει και να το συντηρήσει. Διαχειριστής σε αυτήν την περίπτωση είναι το τεχνικό τμήμα του οργανισμού. Η προσέγγιση αυτών των CMS εξασφαλίζει ότι φιλοξενείς και διαχειρίζεσαι τα δικά σου δεδομένα. Επίσης, τα Licensed είναι ιδανικά για οργανισμούς, οι οποίοι διατηρούν ήδη στις εγκαταστάσεις του κάποιο είδος υπηρεσίας παρόμοιας, όπως για παράδειγμα το σύστημα Διαχείρισης Εξυπηρέτησης Πελατών (CRM), οπότε θα ήταν πιο φθηνό να συντηρούν ταυτόχρονα και ένα CMS.

### Commercial, Open source, Managed Open Source (με βάση το είδος του παρόχου)

**Commercial:** πρόκειται για λογισμικό, που προέρχεται είτε από κερδοσκοπικές είτε από μη κερδοσκοπικές εταιρίες. Οι πάροχοι αυτοί αναπτύσσουν κατά κύριο λόγο το λογισμικό, το οποίο στην συνέχεια πουλάνε και υποστηρίζουν τεχνικά. Στην σημερινή εποχή, οι εμπορικές αυτές λύσεις είναι πιο συχνές από τις ελεύθερες λύσεις των open source CMS. **Open Source:** πρόκειται για μία λύση CMS, που δημιουργείται και συντηρείται από έναν ανεπίσημο και ανιδιοτελή συνεργάτη μίας κοινότητας χρηστών. Στην συνέχεια, το λογισμικό αυτό διανέμεται για συγκεκριμένο σκοπό στα μέλη αυτής της κοινότητας. Για αυτά τα ανοιχτά λογισμικά θα πρέπει σαφώς στο κόστος τους να συμπεριληφθεί και τα έξοδα τεχνικής υποστήριξης τους, τα οποία σαφώς και είναι αυξημένα σε αυτό το μοντέλο. Ακόμη, θα πρέπει να προστεθεί το εσωτερικό hardware και λογισμικό και το τεχνικό προσωπικό που χρειάζεται για να συντηρηθεί αυτό το σύστημα, όπως είναι για παράδειγμα οι προγραμματιστές, οι οποίοι εγκαθιστούν τις ανανεώσεις και εξελίσσουν τις λειτουργίες του προγράμματος.

**Managed Open Source:** πρόκειται για έναν συνδυασμό της εμπορικής και της ελεύθερης προσέγγισης, όπου ένας πάροχος υιοθετεί μία open- source λύση σαν την βασική του πλατφόρμα και στην συνέχεια προσφέρει την λύση αυτή σε άλλους σε συνδυασμό με συμπληρωματικές υπηρεσίες τεχνικής υποστήριξης. Αυτή η λύση ουσιαστικά σχεδόν δεν υπάρχει σήμερα στην κοινότητα των μη-κερδοσκοπικών παρόχων. Παρόλα αυτά, καθώς οι λύσεις open- source ωριμάζουν, οι ειδικοί περιμένουν ότι θα εμφανιστούν πολύ πιο έντονα. Όσον αφορά τη διάκριση των CMS σε σχέση με τον τρόπο παράδοσης τους, έχουμε δύο μορφές λογισμικού. Υπάρχουν εκατοντάδες επιλογές από CMS και των δύο κατηγοριών και η κάθε μία από αυτές διαφέρει στην υλοποίηση, στο κόστος και στην εξυπηρέτηση.

### **3.1.7 Κριτήρια Επιλογής CMS**

Ένα CMS αποτελεί για τους περισσότερους οργανισμούς, οποιουδήποτε μεγέθους, μία αγορά κεφαλαίου. Επειδή, λοιπόν, οι λύσεις που προσφέρονται στην διαχείριση περιεχομένου είναι πολλαπλές και πολλές φορές πολύπλοκες και εξειδικευμένες, υπάρχουν κάποιοι παράγοντες που πρέπει να λάβει κανείς υπόψη πριν αγοράσει, κατεβάσει από το Διαδίκτυο και εγκαταστήσει ένα CMS. Τα κριτήρια επιλογής του λογισμικού αυτού, επομένως θα πρέπει να είναι τα εξής:

**Open Source ή Commercial:** στην επιλογή αυτή σημαντικό παράγοντα παίζει η έννοια κόστος. Στην περίπτωση του open-source λογισμικού, αυτό παρέχεται «δωρεάν». Στην πραγματικότητα, όμως, κρύβει κόστη σχετικά με την τεχνική υποστήριξη του. Τα ερωτήματα, που πρέπει να απαντηθούν είναι, ποιος θα υποστηρίξει τεχνικά το λογισμικό και ποιος θα δημιουργεί νέες λειτουργίες και θα εγκαθιστά τις ανανεώσεις. Χρειάζεται, άρα, μεγάλη προσοχή, καθώς υπάρχουν πολλές βιώσιμες open-source λύσεις, αλλά καλό θα ήταν πάντα να συνυπολογίζεται το συνολικό κόστος.

ASP ή Licensed: στην επιλογή αυτή σημαντικό παράγοντα παίζει το που θα εγκατασταθεί το λογισμικό και η βάση δεδομένων. Υπάρχουν οργανισμοί, που προτιμούν να έχουν τον άμεσο έλεγχο της ιστοσελίδας και των δεδομένων τους, και να φιλοξενούν για το λόγο αυτό το CMS στις εγκαταστάσεις τους. Άλλοι οργανισμοί, για να γλιτώσουν το διαχειριστικό κόστος, αναθέτουν την εγκατάσταση και την συντήρηση του CMS σε εξωτερικούς συνεργάτες. Τα ερωτήματα, που τίθενται, είναι: υπάρχει το απαραίτητο προσωπικό, για να αντιμετωπίσει τα προβλήματα που μπορεί να προκύψουν τις πλέον ακατάλληλες ώρες, όπως πολύ αργά το βράδυ; Υπάρχει ο εξοπλισμός, που χρειάζεται για την συντήρηση του λογισμικού, όπως για παράδειγμα στην περίπτωση φιλοξενίας και κάποιου άλλου συστήματος, με αντίστοιχες προϋποθέσεις και κόστος; Χρειάζεται, τέλος και σε αυτή την περίπτωση να λαμβάνεται υπόψη το γεγονός, ότι η διαχείριση του περιεχομένου στο Διαδίκτυο είναι από τις πιο χρονικά ευαίσθητες λειτουργίες, λόγω της φύσης του μέσου, που προσφέρεται για γρήγορη παροχή πληροφοριών και περιεχομένου.

Χρήστες- Συντάκτες Περιεχομένου: στο κριτήριο αυτό απαιτείται να υπολογιστεί ρεαλιστικά ο αριθμός των χρηστών, που θα συνεισφέρουν στην ιστοσελίδα. Παράλληλα, σημαντικό ρόλο παίζει και το είδος του περιεχομένου, για το οποίο ο καθένας από αυτούς θα είναι υπεύθυνος, και αν το περιεχόμενο αυτό θα δημοσιεύεται άμεσα στον δικτυακό τόπο ή θα περνάει από τον έλεγχο κάποιου αρχισυντάκτη. Οι επιλογές αυτές θα βοηθήσουν στην επιλογή CMS, τα οποία θα προσφέρουν δυναμική και ασφαλή λειτουργία ροής περιεχομένου και δυνατότητα διαχείρισης και ελέγχου ενός μεγάλου αριθμού συντακτών, αν αυτό χρειάζεται.

Είδη Περιεχομένου: το κριτήριο αυτό αφορά τις μορφές του υλικού, που θα εμφανίζεται στην ιστοσελίδα. Οι περισσότερες εφαρμογές διαχειρίζονται κείμενα, γραφικά και φωτογραφίες. Αν, όμως, στο δικτυακό τόπο δημοσιεύεται υλικό με πλούσια μορφοποίηση, όπως για παράδειγμα με πλάγια, έντονα, υπογραμμισμένα και μαρκαρισμένα γράμματα, ή πίνακες και λίστες με κουκίδες, συλλογές φωτογραφιών και ήχος ή βίντεο streaming, τότε, το CMS, που θα επιλεγεί, θα πρέπει να προσφέρει αυτές τις δυνατότητες.

Μονάδα Συσχετιζόμενων Συστημάτων: το κριτήριο αυτό αφορά την ύπαρξη κάποιας μορφής διασύνδεσης ανάμεσα στο CMS και στα υπόλοιπα συστήματα, όπως αυτό των χορηγιών από τους επισκέπτες της ιστοσελίδας (Donation System), του συστήματος εξυπηρέτησης πελατών και του συστήματος των ηλεκτρονικών πωλήσεων. Όλες αυτές οι διασυνδέσεις είναι διαθέσιμες από κάποιους εξειδικευμένους παρόχους, οι οποίοι συνεργάζονται με εταιρίες που σχεδιάζουν τα παραπάνω συστήματα, ώστε να υπάρχει η κατάλληλη συνεργασία. Παράλληλα, σημαντικό παράγοντας είναι και η εξέλιξη ενός δικτυακού τόπου, αφού καθώς αυτή ωριμάζει, οι σχέσεις μεταξύ των εμπλεκόμενων συστημάτων και βάσεων δεδομένων γίνεται σαφώς πιο πολύπλοκη.

Αναφορές: το κριτήριο αυτό αφορά το είδος των στατιστικών αποτελεσμάτων, που θα αναφέρει το CMS. Σε περίπτωση που απαιτούνται ιδιαίτερες αναφορές, όπως αυτές που χρειάζονται από τους υπεύθυνους των μελών, των χορηγών και της επικοινωνίας, τότε θα πρέπει να υποστηρίζονται από το επιλεγμένο CMS, ώστε να αξίζει η επένδυση σ' αυτό.

Επανασχεδίαση ή Μετακίνηση: το κριτήριο αυτό αφορά την πιθανότητα ανασχεδίασης του δικτυακού χώρου και την μετακίνηση στοιχείων, που θα χρησιμοποιηθούν από την παλιά ιστοσελίδα. Η δυνατότητα εύκολης μετακίνησης του κώδικα και των δεδομένων της ιστοσελίδας είναι πολύ σημαντική σε αυτήν την περίπτωση.

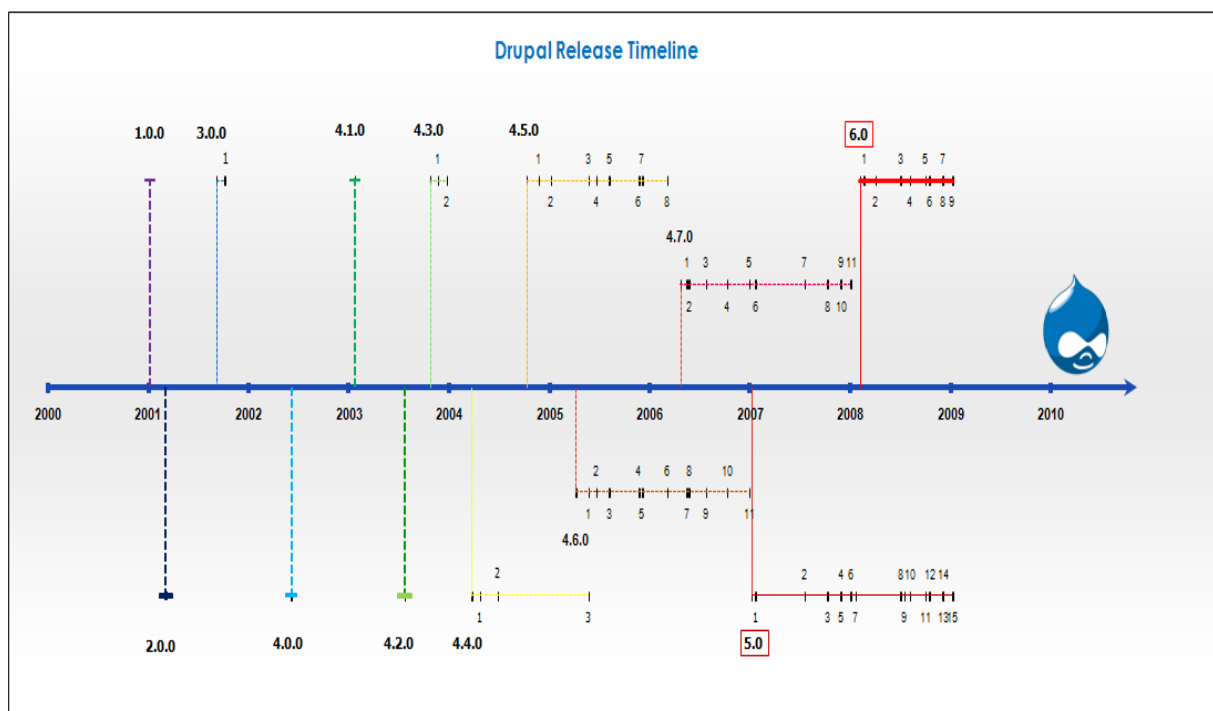
Πολύπλοκτη Εμφάνιση: το κριτήριο αυτό αφορά την υποστήριξη από το λογισμικό της πολύπλοκτης παρουσίασης του δικτυακού τόπου. Όταν η ιστοσελίδα περιέχει δυναμικά μενού πλοήγησης, στοιχεία Flash, ή άλλες σύνθετες γλώσσες γραφικού σχεδιασμού, χρειάζεται ένα πιο σύνθετο σύστημα διαχείρισης περιεχομένου.



### 3.1.8 Εισαγωγή στο Drupal

Το Drupal είναι ένα ελεύθερο και ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου (Content Management System) και σύστημα διαχείρισης πλαισίου εργασίας (Content Management Framework) γραμμένο σε PHP το οποίο διανέμεται κάτω από τους όρους της GNU General Public Licence. Χρησιμοποιείται σαν back-end system (ο τύπος οπου γίνεται η επεξεργασία των εισαγόμενων από τον χρήστη δεδομένων, μέσω φορμών) από το 2.1% όλων των web sites παγκοσμίως, όπως προσωπικών blogs πολιτικών και κυβερνητικών web sites, όπως του λευκού οίκου [whitehouse.gov](http://whitehouse.gov) και του [data.gov.uk](http://data.gov.uk). Η στάνταρ έκδοση του Drupal γνωστή ως Drupal core, περιέχει κοινά βασικά χαρακτηριστικά όπως τα υπόλοιπα συστήματα διαχείρισης περιεχομένου. Μερικά από αυτά είναι δημιουργία και διατήρηση λογαριασμών χρηστών, διαχείριση μενού, RSS feeds, επιλογή διαφόρων θεματικών παραλλαγών καθώς και πλήρη διαχείριση του συστήματος.

Το Drupal 1.0 αρχικά γράφτηκε από τον [Dries Buytaert](#) ως πινάκας ανακοινώσεων, και έγινε project ανοιχτού κώδικα το 2001. Από τότε έχουν δημοσιευτεί επτά εκδόσεις.



Σχήμα 4: Εκδόσεις του Drupal

Στην κοινότητα του Drupal με τον όρο πυρήνας-core εννοείτε οτιδήποτε υπάρχει εκτός του φάκελου sites, όπου είναι ο φάκελος για να τοποθετεί ο εκάστοτε διαχειριστής που κάνει εγκατάσταση το Drupal τα αρχεία του, είτε modules είτε themes, έτσι όλα τα άλλα αρχεία, τα οποία είναι εκτός του φακέλου sites, τα οποία έρχονται μαζί με την εγκατάσταση, αποτελούν τον πυρήνα του Drupal. Στην αρχική του παραμετροποίηση στο περιεχόμενο ενός Drupal web site μπορεί συμβάλει είτε ένας εγγεγραμμένος χρήστης είτε ένας μη εγγεγραμμένος. Το Drupal core περιέχει επίσης το ιεραρχικό σύστημα ταξινόμησης, το οποίο επιτρέπει την κατηγοριοποίηση ή το tagging του περιεχομένου με λέξεις κλειδιά για ευκολότερη και γρηγορότερη πρόσβαση.

Μερικά ενσωματωμένα core modules είναι τα εξής, εξειδικευμένα αναζήτηση, ιστολόγια (blogs), ψηφοφορίες (polls), σχόλια και forum, ασφάλεια και ειδοποίηση ενημερώσεων, προφίλ χρηστών, πολυεπίπεδο σύστημα πλοήγησης, περιορισμοί και έλεγχος πρόσβασης (IP address, ρολόι χρηστών), στατιστικά πρόσβασης και πολλά άλλα. Έκτος από τα ενσωματωμένα modules υπάρχουν και οι ενσωματωμένες θεματικές παραλλαγές-themes που εγκαθίσταται μαζί με τον πυρήνα του Drupal. Ένα άλλο χαρακτηριστικό του πυρήνα είναι το localization η εντόπιση γλώσσας δηλαδή.

Από το 2008 και μετά το Drupal είναι διαθέσιμο σε πάνω από 55 γλώσσες συμπεριλαμβανομένων και γλωσσών από τα δεξιά προς τα αριστερά, όπως τα αραβικά. Επίσης το Drupal περιέχει συναρτήσεις όπου εκτελούν εργασίες σχετικές με τις βάσεις δεδομένων όπως, απόδοση ονόματος προθέματος σε πινάκες πολλαπλών sites και παραγωγή καταλλήλων SQL ερωτήσεων. Τέλος, άλλα πολύ βασικό για κάθε είδους ελεύθερο λογισμικό, η κοινότητα που υποστηρίζει το Drupal αριθμοί πάνω από 648,000 μέλη και πάνω από 10,000 κατασκευαστές ιστοσελίδων.

### Πως τα καταφέρνει το Drupal

Οι άνθρωποι συχνά σκέφτονται μια ιστοσελίδα ως μια συλλογή από στατικές σελίδες, με ορισμένες λειτουργίες (όπως ένα blog, ή μια μηχανή ειδήσεων) ενσωματωμένες που την απαρτίζουν. Όταν πάνε να διαχειριστούν το site τους, σκέφτονται την σελίδες τους σαν μια δένδροειδή δομή όπου μπορούν απλά να την επεξεργαστούν. Από την άλλη πλευρά το Drupal αντιμετωπίζει κάθε τύπο περιεχομένου, ως ένα κόμβο. Στατικές σελίδες, blog spots, και ειδήσεις (μερικές πιθανές μορφές κόμβου) όλα αποθηκεύονται με τον ίδιο τρόπο, και η δομή πλοήγησης του site έχει σχεδιαστεί ξεχωριστά από την επεξεργασία των μενού, των views (κατάλογοι περιεχομένου), και των μπλόκς, πλευρικά περιεχόμενα που συχνά έχουν κάποιους συνδέσμους προς άλλες ενότητες του ιστότοπου.

Στο Drupal δεν υπάρχει ο διαχωρισμός που υπάρχει σε άλλα πρότυπα κωδικοποίησης σελίδων για παράδειγμα η XHTML παρέχει την δομή των σημαντικών πληροφοριών ενώ το CCS κανονίζει την τοποθέτηση τους. Στο Drupal ένας κόμβος περιέχει την δομή των σημαντικών πληροφοριών (όπως ο τίτλος, το περιεχόμενο, ο συγγραφέας κ.α.) που ανήκουν σε ένα blog spot ή σε ένα αντικείμενο ειδήσεων ενώ το menu system καθώς και το taxonomy και οι views δημιουργούν την αρχιτεκτονική των πληροφοριών. Τέλος το theme system μαζί με κάποια άλλα modules, προαιρετικά, ελέγχει το πώς θα φαίνεται το site στους επισκέπτες. Από την στιγμή όπου κάθε επίπεδο είναι χωριστό από το άλλο, μπορούμε να παρέχουμε έναν τελείως διαφορετικό τρόπο πλοήγησης και παρουσίασης του περιεχομένου μας σε διαφορετικούς χρήστες βάση των αναγκών και των ρόλων που έχουν. Τέλος η σελίδες μπορούν να ομαδοποιηθούν διαφορετικά, να έχουν προτεραιότητα με διαφορετική σειρά καθώς και ποικίλες λειτουργίες και περιεχόμενο μπορούν να εμφανίζονται ή όχι.

### Node: Το μυστικό της ευελιξίας του Drupal

Στην πιο απλή του μορφή ένας node-κόμβος είναι ένα σύνολο από σχετικές μεταξύ τους πληροφορίες. Όταν δημιουργείται ένα νέος blog spot, δεν δημιουργείται μόνο το κείμενο του σώματος (body text) αλλά επίσης και ο τίτλος του το περιεχόμενο του το link προς τον συγγραφέα, η ημερομηνία δημιουργίας τα taxonomy tags και λοιπά. Μερικά από αυτά τα στοιχεία θα εμφανιστούν από το επίπεδο της θεματικής παραλλαγής (theme layer) όταν ο κόμβος κληθεί-εμφανιστεί στην οθόνη. Τα υπόλοιπα είναι meta-data όπου ελέγχουν πότε ο κόμβος θα εμφανιστεί-όπως το taxonomy και η κατάσταση δημοσίευσης, αληθείς ή όχι.

Από την στιγμή που κάθε στοιχείο περιεχομένου είναι ένας κόμβος και περιέχει τις ίδιες βασικές πληροφορίες, καθένας μπορεί να διαχειριστεί με ένα στάνταρ τρόπο από το Drupal και τα modules. Αυτό επιτρέπει στους κατασκευαστές ιστοσελίδων να διαλέξουν που ακριβώς θα εμφανίζεται το περιεχόμενο αλλά και πως ακριβώς θέλουν να εμφανίζεται σε κάθε περίπτωση. Ο περισσότερος χρόνος ενός Drupal site κατασκευαστή ξοδεύετε στο τι είδους πληροφορίες θα αποθηκεύονται στους κόμβους και στη ρύθμιση των δομών μέσω τον οποίον θα εμφανίζονται. Στο Drupal δεν υπάρχει περιορισμός στο πως θα φαίνεται το περιεχόμενο του ιστότοπου. Μπορούν να οριστούν συστήματα πλοήγησης και custom θέματα, μπλοκ μικρά κομμάτια περιεχομένου και σχόλια. Τα σχόλια στο Drupal είναι κάτοικοι δεύτερης κατηγορίας σε σχέση με τους κόμβους. Τα σχόλια είναι μέρος του blog system και μπορούν να ενεργοποιηθούν σε τύπο κόμβου.

### Συνεργασία με τον πυρήνα

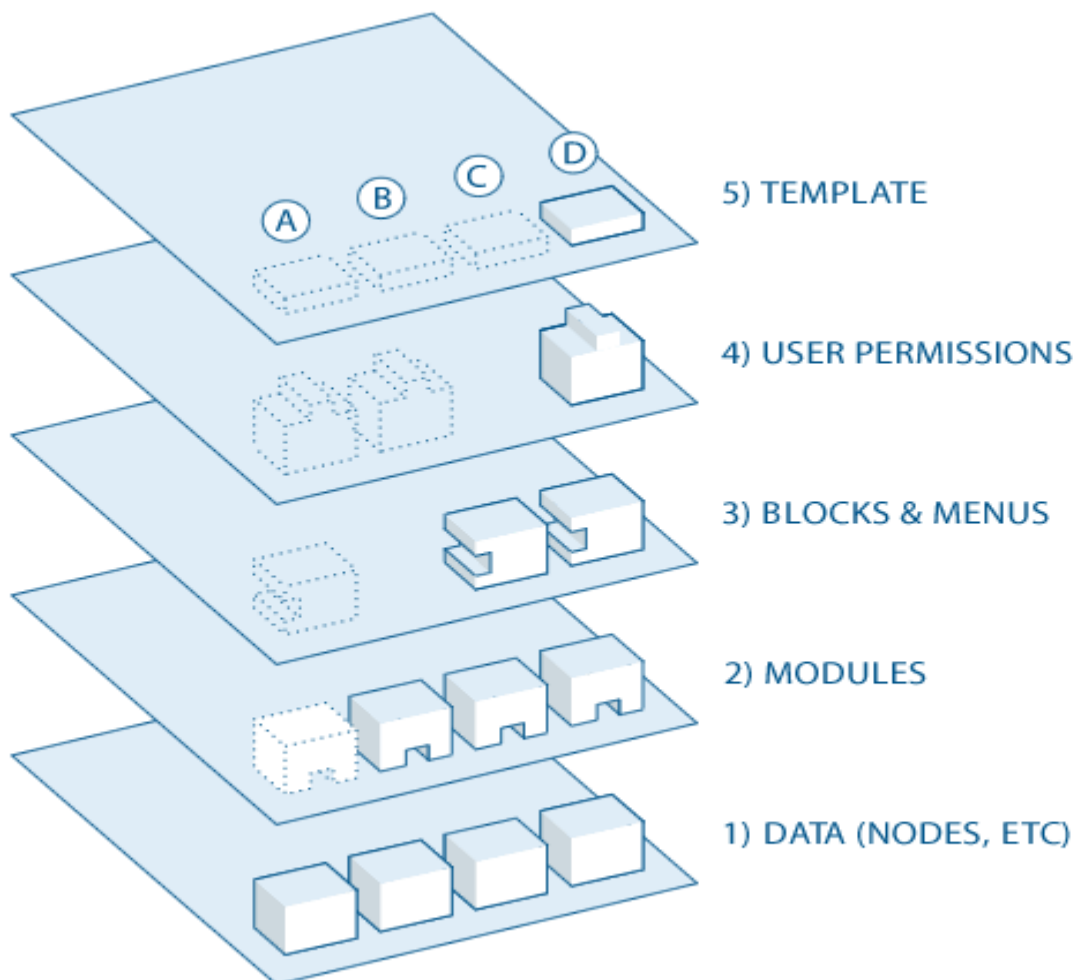
Δημιουργώντας ένα ενημερωτικό ιστότοπο που εκπέμπει από ένα προς πολλούς (one to many) είναι κάτι που τα περισσότερα CMS κάνουν με την απλή τους εγκατάσταση.

Ωστόσο το Drupal διαφέρει στο ότι όταν θέλουμε να ενδυναμώσουμε του χρήστες του ιστότοπου στο να δημιουργήσουν περιεχόμενο και να συνδέονται μεταξύ τους, μεταφερόμαστε δηλαδή από το ένα προς πολλούς στο πολλούς προς πολλούς (many to many) με το Drupal γίνεται εύκολα.

Με μερικά CMS μπορείς να κατασκευάσεις ένα blog και να εγκαταστήσεις κάποια plug-in και να διαχειριστείς μια κοινότητα χρηστών αλλά τι γίνεται όταν θέλουμε να δώσουμε χωριστά blogs σε κάθε ένα χρήστη, τη δυνατότητα να κατηγοριοποιούν το περιεχόμενό τους έτσι ώστε να εμφανίζεται κάθε blog με το δική του θεματική παραλλαγή καθώς και τη δυνατότητα να δημιουργούν θέματα που να εμφανίζονται σε όλα τα blogs ή ακόμη και top five λίστες; ακόμη αν θέλουμε να συμπεριλάβουμε όλα αυτά μέσα σε ένα forum ή σε ένα wiki-like περιβάλλον με κάθε χρήστη να έχει την δικιά του photo gallery; το Drupal έχει σχεδιαστεί από το μηδέν έτσι ώστε οι κατασκευαστές ιστοσελίδων να μπορούν να έχουν τον πλήρη έλεγχο στην δημιουργία περιεχομένου και ακόμη οι διαχειριστές στους χρήστες.

### Η ροή πληροφοριών στο Drupal

Για να πάμε βαθύτερα στο Drupal θα πρέπει να κατανοήσουμε πως διακινούνται οι πληροφορίες στο Drupal ανάμεσα στα επίπεδα του συστήματος. Υπάρχουν πέντε βασικά επίπεδα προς μελέτη.



**Σχήμα 5: Τα πέντε βασικά επίπεδα του Drupal**

- 1) Στην βάση του συστήματος υπάρχει οι συλλογή των κόμβων-nodes ή αλλιώς η δεξαμενή των δεδομένων. Πριν οτιδήποτε μπορεί να εμφανιστεί στον ιστότοπο, πρέπει να εισαχθεί ως δεδομένο.
- 2) Στο επόμενο επίπεδο βρίσκονται οι μονάδες-modules. Οι μονάδες είναι λειτουργικά plug in τα οποία είτε είναι μέρη του πυρήνα του Drupal (είναι φορτωμένα στο Drupal) είτε είναι ξεχωριστά στοιχεία τα οποία έχουν κατασκευαστεί και διανέμονται από την κοινότητα του Drupal. Μονάδες που έχουν κατασκευαστεί για να λειτουργούν πάνω στον πυρήνα του Drupal επιτρέπουν την παραμετροποίηση των δεδομένων των κόμβων, το <<στήσιμο>> ενός e-commerce, την ταξινόμηση και την εμφάνιση του περιεχομένου προγραμματιστικά(πράγμα που έχει γίνει στην παρούσα πτυχιακή εργασία) και πολλά άλλα. Υπάρχουν χιλιάδες διαφορετικές επιλογές στο γρήγορα αναπτυσσόμενο χώρο των συνεισφερόμενων μονάδων.
- 3) Στο επόμενο επίπεδο βρίσκονται τα block και τα μενού. Τα block συχνά παρέχουν την έξοδο από ένα module ή μπορούν να δημιουργηθούν έτσι ώστε να εμφανίζουν οτιδήποτε θέλουμε και μπορούν να τοποθετηθούν οπουδήποτε μέσα στο πρότυπο εξόδου(template layout) ή αλλιώς θέμα(theme-θεματική παραλλαγή). Τα block μπορούν να ρυθμιστούν ώστε να εμφανίζονται με διάφορους τρόπους καθώς και να εμφανίζονται μόνο σε συγκεκριμένες σελίδες ή μόνο σε συγκεκριμένους χρήστες.
- 4) Στο τέταρτο επίπεδο βρίσκονται τα δικαιώματα χρηστών. Εδώ είναι που οι ρυθμίσεις ορίζονται έτσι ώστε να καθοριστούν διαφορετικά είδη χρηστών που μπορούν να δουν ή να κάνουν κάτι. Δικαιώματα ορίζονται για διαφορετικούς ρόλους και ανάλογα χρήστες αντιστοιχούνται σε αυτούς τους ρόλους με σκοπό να αποκτήσουν τα ορισμένα δικαιώματα.
- 5) Στην κορυφή των επιπέδων βρίσκεται η θεματική παραλλαγή(θέμα-theme). Αυτό είναι κατασκευασμένο κυρίως από XHTML και CSS αναμιγμένα με μερικές PHP μεταβλητές έτσι ώστε το παραγόμενο από το Drupal περιεχόμενο να τοποθετείτε στο κατάλληλο σημείο. Σε κάθε θέμα επίσης περιλαμβάνεται και ένα σετ συναρτήσεων όπου μπορούν να χρησιμοποιηθούν για να παρακάμψουν (override) τις βασικές συναρτήσεις των μονάδων, με σκοπό να προσφέρουν πλήρη έλεγχο στο πως οι μονάδες θα παράγουν το mark up σε μια συγκεκριμένη έξοδο.

Αυτό το από κάτω προς τα πάνω κατευθυνόμενο διάγραμμα ελέγχει το πώς λειτουργεί το Drupal. Εάν υπάρχει κάποια νέα λειτουργία που θέλει κάποιος να προσθέσει απλά κάνει upload την μονάδα (module) του στο σύστημα. Αυτό επίσης είναι και το μεγάλο πλεονέκτημα στο ανοιχτό λογισμικό και κατά επέκταση και του Drupal.

### 3.1.9 Βασικές έννοιες στο Drupal

#### Πεδία –Fields

Ένα πεδίο είναι η μικρότερη μονάδα πληροφορίας στο Drupal. Ορισμένα παραδείγματα πεδίων μπορούν να περιέχουν ένα αριθμό(ακέραιο, πραγματικό, δυαδικό κ.α.) μια ημερομηνία ή ακόμη και κείμενο.

#### Κόμβοι-Nodes

Ένας κόμβος στο Drupal είναι ο γενικός όρος για ένα κομμάτι περιεχομένου σε ένα ιστότοπο. Ένας κόμβος αποτελείται από διάφορα πεδία. Να σημειωθεί ότι ο όρος “κόμβος” δεν σημαίνεται με την μαθηματική έννοια ως μέρος ενός δικτύου. Μερικά παραδείγματα κόμβων είναι τα εξής: Σελίδες σε βιβλία, η πρώτη σελίδα ενός ιστότοπου, θέματα συζητήσεων σε χώρους κοινωνικής δικτύωσης, εισαγωγές σε blogs άρθρα με νέα ειδήσεων κ.α. Κάθε κόμβος σε ένα ιστότοπο ανήκει σε κάποιον τύπο περιεχομένου-content type. Επίσης έχει πεδία όπως το Node ID, τον τίτλο, την ημερομηνία δημιουργίας, τον συγγραφέα του κόμβου(ένα χρήστη του ιστότοπου δηλαδή), ένα σώμα-Body(το οποίο μπορεί να αγνοηθεί για κάποιους τύπους δεδομένων) καθώς και κάποιες ακόμη ιδιότητες. Χρησιμοποιώντας κάποιες μονάδες όπως το kit κατασκευής περιεχομένου-Content Construction Kit (CCK) Module ή τη μονάδα ταξινόμησης περιεχομένου-Taxonomy Module μπορούν να προστεθούν περισσότερα πεδία σε ένα κόμβο.

#### Σχόλια-Comments

Τα σχόλια είναι ένας άλλος τύπος περιεχομένου που μπορεί να υπάρχει στον ιστότοπο εάν έχει ενεργοποιηθεί η μονάδα των σχολίων στον πυρήνα. Κάθε σχόλιο είναι τυπικά ένα μικρό κομμάτι περιεχομένου οπου ένας χρήστης υποβάλλει-submit ή επισυνάπτει-attached σε ένα συγκεκριμένο κόμβο.

#### Μονάδες-Modules

Μια μονάδα είναι λογισμικό (κώδικας) το οποίο επεκτείνει την λειτουργικότητα και της δυνατότητες του Drupal. Η μονάδες του πυρήνα περιέχονται με την βασική εγκατάσταση του Drupal και μπορούν να ενεργοποιηθούν η λειτουργίες τους χωρίς να εγκατασταθεί επιπρόσθετο λογισμικό. Οι προσφερόμενες μονάδες που είναι διαθέσιμες στο [Modules download section of drupal.org](http://drupal.org/modules) ,μονάδες δηλαδή που έχουν κατασκευαστεί από άτομα της κοινότητας του Drupal , εγκαθίστανται μέσα από την εγκατάσταση του Drupal. Επίσης μπορεί ο καθένας να δημιουργήσει την δική του μονάδα, αυτό όμως απαιτεί την βαθιά κατανόηση του Drupal ,του PHP προγραμματισμού και του Drupal module API

#### Περιοχή, Μπλοκ, Μενού-Region, Block, Menu

Οι σελίδες σε ένα Drupal ιστότοπο χωρίζονται σε περιοχές-regions που μπορούν να περιέχουν την επικεφαλίδα το υποσέλιδο τις πλευρικές στήλες και την βασική περιοχή περιεχομένου, σε ένα custom θέμα μπορούν να οριστούν επιπλέον περιοχές. Τα μπλοκ-block είναι διακριτά τμήματα πληροφοριών που μπορούν να εμφανίζονται σε διάφορες περιοχές της ιστοσελίδας του ιστότοπου. Τα μπλοκ μπορούν να έχουν την μορφή μενού (που είναι συνδεδεμένα με την πλοήγηση του ιστότοπου) την μορφή εξόδου από μια μονάδα ή δυναμικά και στατικά τμήματα πληροφορίας δημιουργημένα από τον χρήστη.

## Χρήστες, Δικαιώματα, Ρόλοι-Users, Permissions, Roles

Κάθε επισκέπτης σε ένα Drupal ιστότοπο, είτε έχει λογαριασμό και εισαχθεί στο σύστημα είτε είναι ανώνυμος για το Drupal θεωρείται ως χρήστης. Κάθε χρήστης έχει ένα μοναδικό αριθμητικό κωδικό-User ID και κάθε μη ανώνυμος χρήστης έχει επίσης ένα όνομα και μια διεύθυνση ηλεκτρονικού ταχυδρομείου. Οι ανώνυμοι χρήστες έχουν κωδικό χρήστη-User ID μηδέν (0). Ο χρήστης με τον κωδικό χρήστη-User ID ένα (1) ο οποίος αντιστοιχεί στον λογαριασμό χρήστη που δημιουργήθηκε κατά την εγκατάσταση του Drupal είναι ειδικός: αυτός ο χρήστης έχει τα δικαιώματα για να κάνει οτιδήποτε στο Drupal ιστότοπο. Σε άλλους χρήστες του ιστότοπου μπορούν να αποδοθούν δικαιώματα μέσω ρόλων. Για να γίνει αυτό θα πρέπει πρώτα να δημιουργηθεί ένας ρόλος, για παράδειγμα “διαχειριστές περιεχομένου” ή “μέλος”. Έπειτα θα πρέπει να παρασχεθούν κάποια δικαιώματα σε αυτό τον ρόλο που θα λένε στο Drupal τι μπορεί και τι δεν μπορεί να κάνει αυτός ο ρόλος στον ιστότοπο. Τέλος θα χορηγηθούν σε κάποιους χρήστες οι ρόλοι που δημιουργήθηκαν.

## Θεματική παραλλαγή(Θέμα)-Theme

Το θέμα ελέγχει το πώς ο ιστότοπος θα εμφανίζεται συμπεριλαμβανομένου των γραφικών του σχεδίου και των χρωμάτων. Ένα θέμα αποτελείται από ένα ή περισσότερα PHP αρχεία που ορίζουν την μορφή της HTML εξόδου που θα έχουν οι σελίδες του ιστότοπου καθώς και ένα ή περισσότερα CSS αρχεία που ορίζουν την μορφοποίηση των γραμματοσειρών τα χρώματα σε διάφορα τμήματα των σελίδων καθώς και άλλα στυλ.

## Ταξινόμηση-Taxonomy

Το Drupal χρησιμοποιεί ένα σύστημα-αλγόριθμο για να κατηγοριοποιεί το περιεχόμενο γνωστό ως Taxonomy το οποίο βρίσκεται στον πυρήνα του Drupal. Μπορούν να οριστούν λεξιλόγια (μια ομάδα από όρους ταξινόμησης-Taxonomy terms) και να προστεθούν όροι σε κάθε λεξιλόγιο. Τα λεξιλόγια μπορούν να είναι επίπεδα ή ιεραρχικά δομημένα, μπορούν να επιτρέπουν απλή ή πολλαπλή επιλογή επίσης μπορεί να οριστεί ελεύθερη απόδοση ετικετών (tagged) ή αλλιώς λέξεις κλειδιά (keywords). Κάθε λεξιλόγιο μπορεί να ενσωματωθεί σε ένα ή περισσότερους τύπους περιεχομένου και κατά αυτό τον τρόπο οι κόμβοι στον ιστότοπο να ομαδοποιηθούν σε κατηγορίες σε ετικέτες ή να κατηγοριοποιηθούν με όποιον τρόπο επιθυμείτε.

## Βάση Δεδομένων-Database

Το Drupal αποθηκεύει τις πληροφορίες σε μια βάση δεδομένων, κάθε τύπος δεδομένων έχει τον δικό του πίνακα στην βάση δεδομένων. Για παράδειγμα οι βασικές πληροφορίες για τους κόμβους του ιστότοπου αποθηκεύονται στον πίνακα Node. Αν χρησιμοποιείται η μονάδα CCK με σκοπό να προστεθούν επιπλέον πεδία στον κόμβο, οι πληροφορίες των πεδίων θα αποθηκευτούν σε διαφορετικούς πίνακες. Οι χρήστες, οι ρόλοι, τα σχόλια, τα δικαιώματα καθώς και άλλες ρυθμίσεις έχουν επίσης τους δικούς τους πίνακες στη βάση δεδομένων.

## Διαδρομή-Path

Όταν επισκεπτόμαστε ένα URL σε ένα Drupal ιστότοπο, το μέρος της URL μετά από την βασική διεύθυνση ονομάζεται διαδρομή-path. Όταν επισκεπτόμαστε μια διαδρομή σε ένα Drupal ιστότοπο, το Drupal αντιλαμβάνεται τι πληροφορίες πρέπει να στείλει στο φυλλομετρητή μέσω ενός ή περισσότερων ερωτημάτων προς την βάση δεδομένων.

Γενικά το Drupal επιτρέπει σε κάθε μονάδα που είναι ενεργοποιημένη στον ιστότοπο να ορίζει διαδρομές για τις οποίες θα είναι υπεύθυνη η αντίστοιχη μονάδα και μόλις επιλεγεί κάποια συγκεκριμένη διαδρομή το Drupal ρωτάει τη μονάδα για το τι θα εμφανίσει στην σελίδα. Για παράδειγμα αν έχουμε μια διεύθυνση σαν αυτή <http://drupal.org/node/19828> το path της διεύθυνσης είναι το “node/19828” και το base address το “drupal.org”. Η μονάδα που είναι υπεύθυνη για αυτή την διαδρομή είναι η μονάδα του πυρήνα του Drupal, Node, έτσι όταν γίνεται κάποια επίσκεψη στην σελίδα αυτή το Drupal δίνει τον έλεγχο στην μονάδα Node να καθορίσει το τι θα εμφανιστεί. Για να καθοριστεί μια διαδρομή σε μια συγκεκριμένη σελίδα σε έναν Drupal ιστότοπο, για την δημιουργία ενός link δηλαδή, κοιτάμε από τη μπάρα διευθύνσεων το URL της σελίδας. Εξ ορισμού το URL μετά από το base address θα ξεκινάει με “?q=” και αυτό μαζί με το path αποτελούν το κατάλληλο URL για να μετατραπεί σε link. Όμως όταν έχουν ενεργοποιηθεί τα Clean URLs’ η διεύθυνση θα έχει αρχειακή δομή, δηλαδή δεν θα χρειάζεται το “?q=”.

### **Τύποι περιεχομένου-Content types**

Ένας ιστότοπος μπορεί να περιέχει πολλούς τύπους περιεχομένου όπως σελίδες πληροφοριών, αντικείμενα ειδήσεων, ψηφοφορίες, δημοσιεύσεις blogs και λοιπά. Στο Drupal κάθε αντικείμενο περιεχομένου αποκαλείτε κόμβος-node, και κάθε κόμβος ανήκει σε ένα μόνο τύπο περιεχομένου, ο οποίος ορίζει διάφορες εξ ορισμού ρυθμίσεις για του κόμβους αυτού του τύπου, όπως το να δημοσιεύεται ο κόμβος αυτόματα ή να επιτρέπονται τα σχόλια στον κόμβο αυτό. Οι δύο βασική τύποι περιεχομένου στο Drupal είναι το page και το story. Με την εγκατάσταση και την ενεργοποίηση διάφορων μονάδων, ανάλογα την μονάδα δηλαδή, περισσότεροι τύποι περιεχομένου είναι διαθέσιμοι, καθώς επίσης μπορούν να δημιουργηθούν και από τον χρήστη κατά βούληση.

### **Διαφοροποίηση μεταξύ χρηστών-Differentiating between kinds of users**

Computer user: το πρόσωπο που εγκαθιστά το Drupal πρέπει να έχει πρόσβαση στον υπολογιστή που είναι εγκατεστημένο το Drupal. Αν το Drupal έχει εγκατασταθεί τοπικά τότε ο computer user είναι το άτομο που έκανε την εγκατάσταση. Αν το Drupal εγκατασταθεί σε έναν απομακρυσμένο server, όπως μια εταιρεία παροχής υπηρεσιών ιντερνέτ, ο computer user μπορεί να είναι ένας λογαριασμός από τους παρακάτω: Web hosting login, FTP login, Cpanel login ή Secure shell (SSH) login.

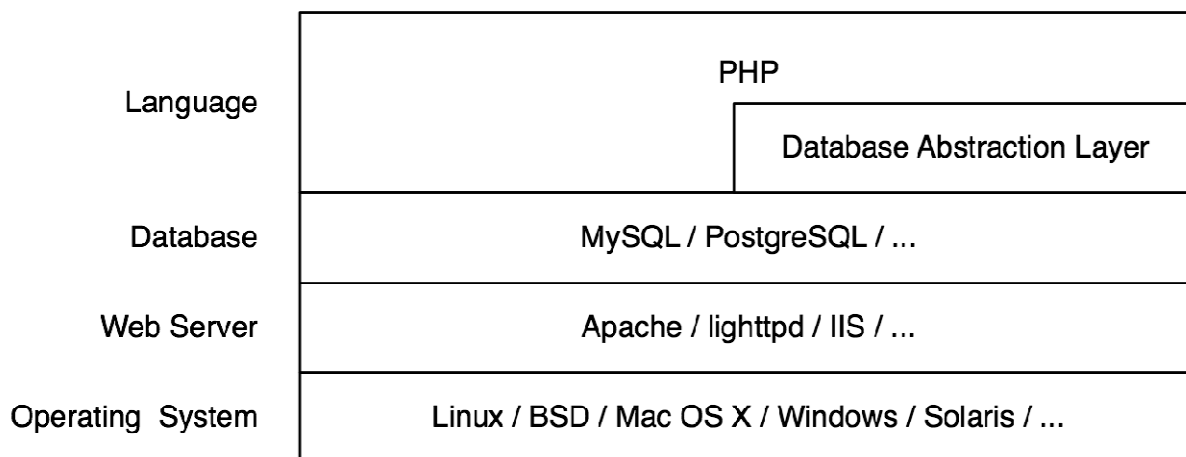
Database user: κάθε εγκατάσταση του Drupal απαιτεί την ύπαρξη μιας βάσης δεδομένων. Κάθε τυπική βάση δεδομένων περιέχει μηχανισμούς έλεγχου πρόσβασης και απαιτούν χρήστες με έγκυρα δικαιώματα ώστε να μπορούν να τροποποιούν την βάση. Μια Drupal εγκατάσταση πρέπει να έχει πλήρη έλεγχο σε μια βάση δεδομένων, έτσι όταν εγκαθίσταται για πρώτη φορά ένας Drupal ιστότοπος δημιουργείται ένας χρήστης της βάσης, είτε τοπικά από κάποιο χρήστη είτε από μια web hosting εταιρεία, με πλήρη δικαιώματα και έπειτα δίνονται στο Drupal το όνομα χρήστη και ο κωδικός έτσι ώστε το Drupal να έχει τον πλήρη έλεγχο της βάσης. Ο database user δεν είναι κάποιο πρόσωπο, είναι ένας λογαριασμός δημιουργημένος από το λογισμικό της βάσης με σκοπό να δώσει στο Drupal πλήρη έλεγχο της βάσης δεδομένων.

User/1: ο “User/1” γνωστός και ως “super-user account” είναι ο λογαριασμός που ζητείται να δημιουργηθεί άμεσα, μετά την εγκατάσταση, από το Drupal. Αυτός ο λογαριασμός είναι μοναδικός στον ιστότοπο και είναι διαφορετικός από όλους τους άλλους λογαριασμούς του Drupal διότι δεν έχει κανένα περιορισμό δικαιωμάτων. Ο User/1 σχετίζεται με ένα μόνο πρόσωπο το οποίο είναι υπεύθυνο για την ενημέρωση του ιστότοπου.

User/2: ο User/2 και όλοι οι άλλοι χρήστες θα πρέπει να σχετίζονται με ένα μόνο πρόσωπο στον Drupal ιστότοπο. Σε εγγεγραμμένους χρήστες μπορούν να ανατεθούν ρόλοι οι οποίοι έχουν διάφορα δικαιώματα με διαφορετική πρόσβαση στο περιεχόμενο του ιστότοπου.

### 3.1.10 Η στοίβα τεχνολογίας του Drupal και ο πυρήνας του-Technology Stack and the core

Οι βασικοί στόχοι στο σχεδιασμό του Drupal είναι δύο, πρώτον να είναι σε θέση να τρέξει καλά σε φθηνούς web hosting λογαριασμούς και δεύτερον να είναι σε θέση να αναβαθμιστεί σε μαζική διανομή ιστότοπων . Ο πρώτος στόχος προϋποθέτει την χρήση της πιο δημοφιλούς τεχνολογίας, και ο δεύτερος προϋποθέτει προσεκτικό και συμπαγή κώδικα.



**Σχήμα 6: Drupal's Stack Technology**

Το λειτουργικό σύστημα είναι σε τόσο χαμηλό επίπεδο στη στοίβα του Drupal που δεν ενδιαφέρεται πολύ γι 'αυτό. Το Drupal τρέχει με επιτυχία σε οποιοδήποτε λειτουργικό σύστημα που υποστηρίζει PHP. Ο πιο ευρέως χρησιμοποιούμενος web server με το Drupal είναι ο Apache αν και άλλοι web διακομιστές (συμπεριλαμβανομένου του Microsoft IIS) μπορούν να χρησιμοποιηθούν. Λόγω της μακράς ιστορίας του Drupal με τον Apache, το Drupal διανέμεται με το αρχείο .htaccess στον root κατάλογο του όπου διασφαλίζει την εγκατάσταση του Drupal. Επίσης υποστηρίζει τις Clean URL's, δηλαδή όσες διευθύνσεις δεν έχουν ερωτηματικά, συμπλεκτικά σύμβολα ή άλλους μη έγκυρους χαρακτήρες. Αυτό επιτυγχάνεται με τη χρήση της εντολής του Apache mod\_rewrite. Αυτό είναι ιδιαίτερα σημαντικό, διότι κατά τη μετάβαση από ένα σύστημα διαχείρισης περιεχομένου σε ένα άλλο οι διευθύνσεις URL του περιεχομένου δεν χρειάζεται να αλλάξουν.

Το Drupal επικοινωνεί με το επόμενο επίπεδο της στοίβας (τη βάση δεδομένων) μέσω ενός λεπτού αφαιρετικού επιπέδου. Αυτό το επίπεδο χειρίζεται την εξυγίανση των SQL ερωτήματα και καθιστά δυνατή τη χρήση βάσεων δεδομένων διαφορετικών προμηθευτών χωρίς να γράφεται ο κώδικας για κάθε τύπο βάσης χωριστά από την αρχή. Οι πιο δημοφιλείς και ελεγμένες βάσεις που δουλεύουν με το Drupal είναι οι MySQL και η PostgreSQL. Το Drupal είναι γραμμένο σε PHP. Η PHP είναι μια γενικής χρήσης Server-side γλώσσα προγραμματισμού σχεδιασμένη για Web development και την παραγωγή δυναμικών ιστοσελίδων.

#### Ο πυρήνας

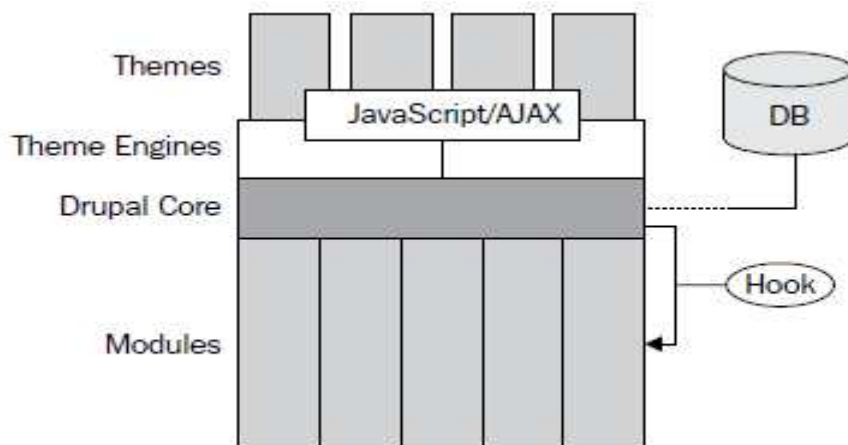
Ένα ελαφρύ framework αποτελεί τον πυρήνα του Drupal. Ο πυρήνας είναι υπεύθυνος για την παροχή των βασικών λειτουργιών και συναρτήσεων που θα χρησιμοποιηθούν για την υποστήριξη των τμημάτων του συστήματος. Ο πυρήνας περιλαμβάνει τον κώδικα που επιτρέπει στο σύστημα του Drupal ξεκινήσει όταν λαμβάνει ένα αίτημα (request), παρέχει μια βιβλιοθήκη με κοινής χρήσης συναρτήσεις που χρησιμοποιούνται με το Drupal, και μονάδες που παρέχουν βασικές λειτουργίες όπως η διαχείριση των χρηστών η ταξινόμηση το templating και άλλα.



### 3.1.11 Μονάδες και Hook system στο Drupal

Σε ένα Drupal ιστότοπο μπορούν αν υπάρχουν τριών ειδών μονάδες. Πρώτον οι μονάδες του πυρήνα που είναι φορτωμένες με το Drupal και αποδεκτές από τους κατασκευαστές του πυρήνα και την κοινότητα του Drupal. Δεύτερον οι συνεισφερόμενες μονάδες, αυτές που έχουν γραφτεί από την κοινότητα του Drupal και διανέμονται κάτω από την ίδια GNU Public Licence (GPL) άδεια καθώς και το Drupal. Και τέλος οι κατά απαίτηση μονάδες που κατασκευάζονται από του δημιουργούς του Drupal ιστότοπου. Μια μονάδα στο Drupal είναι μια συλλογή αρχείων που περιέχουν κάποια λειτουργικότητα και είναι γραμμένα σε PHP. Επειδή ο κώδικας της μονάδας εκτελείται στο πλαίσιο του ιστότοπου, μπορεί να χρησιμοποιήσει όλες τις λειτουργίες και να έχει πρόσβαση σε όλες τις μεταβλητές και τις δομές του πυρήνα του Drupal. Στην πραγματικότητα, μια μονάδα δεν είναι κάτι διαφορετικό από ένα κανονικό αρχείο PHP το οποίο μπορεί ανεξάρτητα να δημιουργηθεί και να ελεγχθεί και στη συνέχεια να χρησιμοποιηθεί για την οδήγηση πολλαπλών λειτουργιών. Η προσέγγιση αυτή επιτρέπει στον Drupal πυρήνα να καλέσει σε συγκεκριμένους χώρους ορισμένες λειτουργίες που ορίζονται στις μονάδες και να ενισχύσει έτσι τη λειτουργικότητα του πυρήνα.

Τα **Hooks** είναι το μέσο με το οποίο οι μονάδες μπορούν να αλληλεπιδράσουν με τον κώδικα του πυρήνα του Drupal. Δίνουν την δυνατότητα σε μια μονάδα να καθορίσει νέα URL και σελίδες μέσα στον ιστότοπο (`hook_menu`), να προσθέσετε περιεχόμενο σε σελίδες (`hook_menu`, `hook_footer`, κλπ.), να δημιουργήσουν προσαρμοσμένους πίνακες στη βάση δεδομένων (`hook_schema`), και πολλά άλλα. [Αυτή η σελίδα](#) απαριθμεί τα hooks που προβλέπονται στον πυρήνα, αλλά κάθε μονάδα μπορεί να καθορίσει τα δικά της hook. Για παράδειγμα, η μονάδα CCK ορίζει το `hook_field_info`, το οποίο μπορεί να χρησιμοποιηθεί από τις μονάδες που θέλουν να καθορίσουν ένα νέο τύπο πεδίου περιεχομένου. Οι περισσότερες μονάδες που ορίζουν κάποια hook θα παρέχουν επίσης και τεκμηρίωση γι' αυτά.



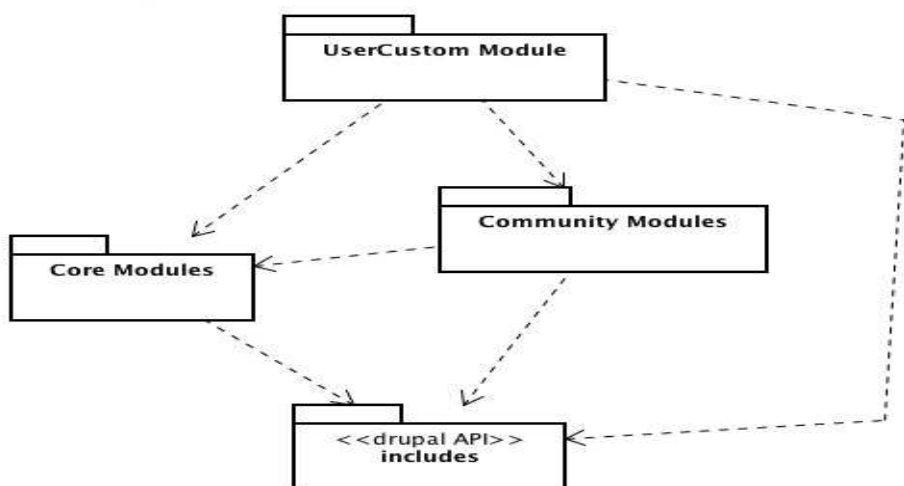
**Σχήμα 7: Σχέση μονάδων πυρήνα, custom, contributed μονάδων και hook**

Τα hook συμβαίνουν σε διάφορα σημεία κατά το νήμα της εκτέλεσης, όπου το Drupal ψάχνει από όλες τις ενεργοποιημένες μονάδες για διάφορα hook. Για παράδειγμα, όταν ο χρήστης επισκέπτεται μια σελίδα βοήθειας σε μια τοποθεσία του Drupal και καθώς το Drupal χτίζει τη σελίδα αυτή, θα δώσει σε κάθε μονάδα την ευκαιρία να παρουσιάσει την τεκμηρίωση της. Αυτό επιτυγχάνεται με τη σάρωση όλου του κώδικα της μονάδας για τις συναρτήσεις που έχουν το όνομα `mymodule_help` (`$` μονοπάτι, `$arg`), όπου "mymodule" είναι το όνομα του module που έχουμε κατασκευάσει, για παράδειγμα, της μονάδας block το hook βοήθειας καλείται `block_help` και της μονάδας node το hook καλείται `node_help`. Το hook μπορεί να παρέχει ορίσματα. Τα ορίσματα του `hook_help` είναι `$path` και `$arg` και επιτρέπουν στον κατασκευαστή να προσδιορίσει σε τι σελίδα ή σελίδες θα εμφανιστούν τα μηνύματα βοήθειας.

Ένα hook μπορεί να θεωρηθεί ως ένας ακροατής (listener) γεγονότων κατά την έννοια ότι ένα συμβάν (event) ενεργοποιεί μια ενέργεια (action). Ένα συμβάν στο Drupal, όπως η διαγραφή ενός κόμβου, θα προκαλέσει το hook "hook\_delete". Αν η μονάδα μας υλοποιεί το hook\_delete, τότε αυτή η συνάρτηση θα εκτελεστεί όταν ένας κόμβος διαγραφεί. Το σύστημα των μονάδων του Drupal βασίζεται στα hooks. Ένα hook είναι μια PHP συνάρτηση που ονομάζεται foo\_bar(), όπου το foo είναι το όνομα της μονάδας (με όνομα αρχείου foo.module) και bar είναι το όνομα του hook. Για κάθε hook ορίζεται ένα σετ παραμέτρων και τύπου επιστροφής. Για την επέκταση του Drupal μια μονάδα χρειάζεται να υλοποιήσει ένα hook.

### 3.1.12 Το Drupal API

Ένα Application Programming Interface (API) είναι μια διεπαφή μεταξύ των στοιχείων ενός μεγάλου συστήματος λογισμικού. Για κάθε μεμονωμένο στοιχείο του συστήματος ένα API χειρίζεται την επικοινωνία μεταξύ των στοιχείων και του πυρήνα. Με αυτό τον τρόπο ένα στοιχείο του συστήματος μπορεί να απομονωθεί από αντίστοιχες αλλαγές στον πυρήνα, μειώνοντας έτσι τις δοκιμές και το debugging για ένα στοιχείο και μόνο. Σε γενικές γραμμές, τα APIs θέτουν ένα πρότυπο για την διαχείριση των χαμηλού επιπέδου λειτουργιών και εισάγει σταθερότητα και ομοιομορφία στον κώδικα.



**Σχήμα 8: Σχέση μονάδων πυρήνα, custom και community μονάδων και API**

Το πιο κοινό παράδειγμα ενός API είναι το API της βάσης δεδομένων, που ενθυλακώνει τις λειτουργίες της βάσης δεδομένων από τον πυρήνα, κατά τέτοιο τρόπο ώστε ο πυρήνας να λειτουργεί ανεξάρτητα από το σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιείται. Το Drupal παρέχει ένα λεπτό στρώμα αφαιρετικότητας της βάσης δεδομένων έτσι ώστε να παρέχει στους προγραμματιστές τη δυνατότητα να υποστηρίζουν πολλούς διακομιστές βάσεων δεδομένων εύκολα. Σκοπός αυτού του στρώματος είναι να διατηρήσει τη σύνταξη και τη δύναμη της SQL όσο το δυνατόν περισσότερο, ενώ αφήνει στο Drupal τον έλεγχο των ερωτημάτων προς τη βάση δεδομένων που πρέπει να γράφονται διαφορετικά για διαφορετικούς διακομιστές και την παροχή βασικών ελέγχων ασφαλείας. Οι περισσότερες ερωτήσεις στο Drupal προς την βάση γίνονται με την κλήση της συνάρτησης db\_query() ή db\_query\_range().

Ένα άλλο παράδειγμα ενός API του Drupal είναι το Form API το οποίο παρέχει εξελιγμένες τεχνικές για τις φόρμες ενός Drupal ιστότοπου και επίσης επιτρέπει σχεδόν απεριόριστες δυνατότητες για την μορφοποίηση τους, την επικύρωση και την εκτέλεση των φορμών.

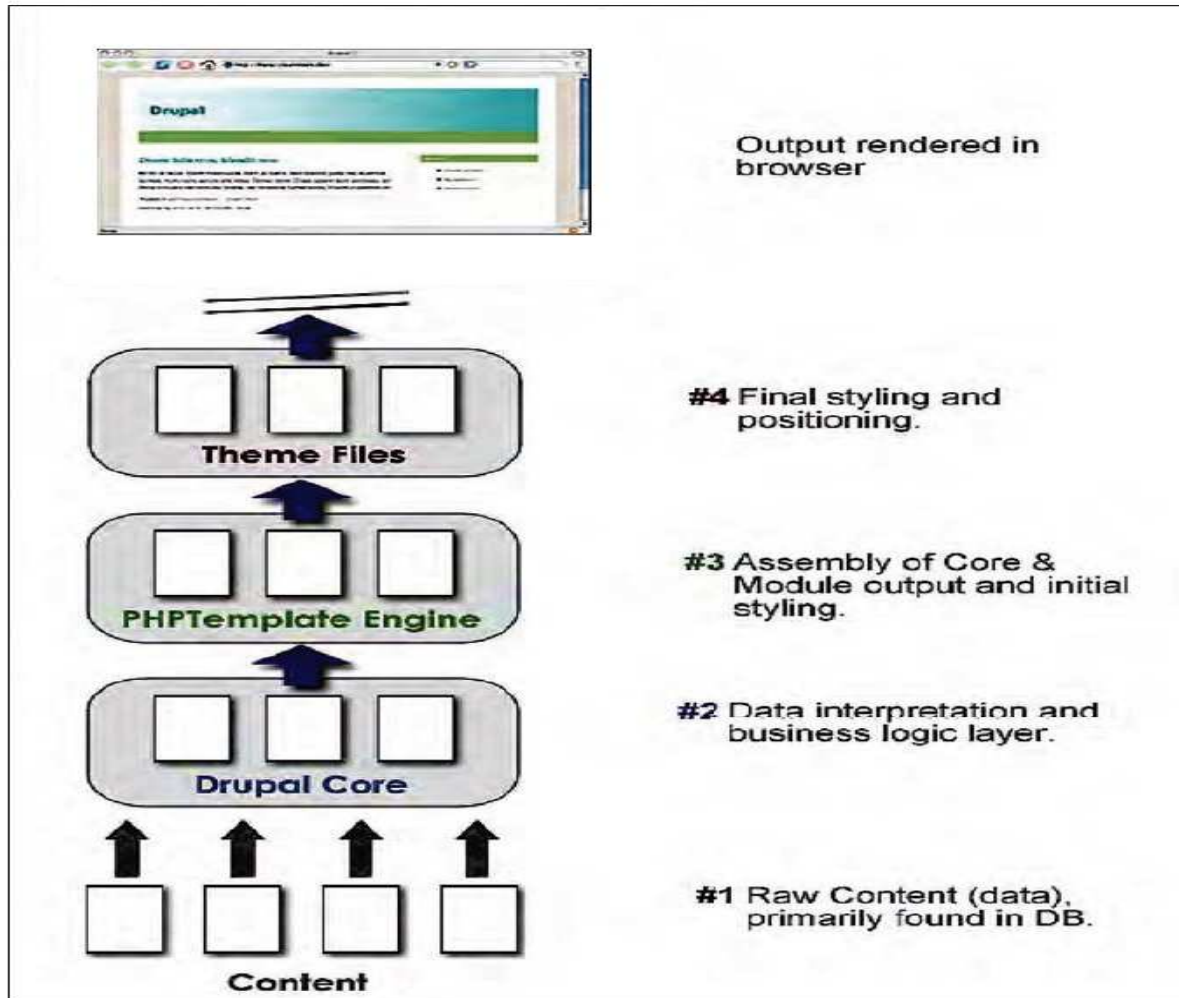
Ακόμα καλύτερα, οποιαδήποτε φόρμα (ακόμα και στον πυρήνα) μπορεί να μεταβληθεί με σχεδόν οποιοδήποτε τρόπο μπορεί κανείς να φανταστεί -στοιχεία μπορούν να αφαιρεθούν, να προστεθούν, και να αναδιαταχθούν. Ίσως το πιο σημαντικό είναι το ότι το API Form παρέχει ένα ασφαλές πλαίσιο για φόρμες, παρέχοντας προστασία από πολλές απειλές, και ο προγραμματιστής δεν έχει να κάνει σχεδόν τίποτα για να έχει αυτή την προστασία. Το API Form του Drupal δίνει σε μια μονάδα τον τρόπο να ορίσει μια φόρμα μέσω ενός πίνακα, μεταφράζοντας τον αργότερα σε HTML κώδικα, να χειριστεί την επικύρωση την υποβολή, και να εμφανίσει τη φόρμα σε άλλες μονάδες που μπορούν επίσης να την αλλάξουν. Η ιδέα του API είναι να εξαλείψει την γραφή HTML κώδικα για οποιαδήποτε κατασκευή φόρμας και να κάνει τη ροή της παρουσίασης, επικύρωσης και εκτέλεσης όσο το δυνατόν ασφαλέστερη και καθαρότερη. Μια φόρμα συνήθως μεταφράζεται σε HTML με το πέρασμα του πίνακα στην συνάρτηση `drupal_get_form`. Η συνάρτηση `drupal_get_form` ανακτά μια φόρμα από την συνάρτηση κατασκευής (constructor function) της φόρμας ή από την κρυφή μνήμη αν η φόρμα κατασκευάστηκε σε προηγούμενο φόρτωμα σελίδας (page load). Έπειτα η φόρμα προωθείται για επεξεργασία και για μετάφραση σε HTML και εμφάνιση αν είναι απαραίτητο.

Επίσης σημαντικό είναι το Node API. Κάθε τύπος περιεχομένου συντηρείται από ένα βασική μονάδα η οποία είναι είτε η μονάδα `node.module`, για τύπους περιεχομένου δημιουργημένους στη διεπαφή χρήστη, είτε από μονάδες που υλοποιούν το `hook_node_info()` (το hook αυτό επιτρέπει σε μια μονάδα να ορίσει ένα ή περισσότερους δικούς της τύπους περιεχομένου) για να ορίσουν ένα τύπο περιεχομένου. Το `hook_nodeapi` επιτρέπει στις μονάδες να επιδρούν σε ενέργειες που επηρεάζουν κάθε είδους κόμβους ανεξάρτητα από το αν κάποια μονάδα ορίζετε από τον κόμβο. Είναι συχνό φαινόμενο το `hook_nodeapi()` να χρησιμοποιείται με το `hook_form_alter()` (το hook αυτό πραγματοποιεί αλλαγές σε μια φόρμα πριν μεταφραστεί σε HTML). Οι μονάδες χρησιμοποιούν το `hook_form_alter()` για να προσθέσουν επιπλέον στοιχεία σε μια φόρμα στο node edit form και το `hook_nodeapi()` χρησιμοποιείται για να γράψει και να διαβάσει από και προς την βάση τις τιμές τις φόρμας. [The Drupal API reference](#).

### 3.1.13 Τι είναι θέμα και τι Μηχανή παραγωγής θεμάτων-Themes and Theme Engine

Στο πλαίσιο του Drupal, ο όρος θέμα σημαίνει μια συλλογή από αλληλένδετα αρχεία που είναι υπεύθυνα για την εμφάνιση και την αισθητική της ιστοσελίδας. Άλλα συστήματα διαχείρισης περιεχομένου (CMS) χρησιμοποιούν διαφορετικά ονόματα για τα αρχεία που επιτελούν την ίδια λειτουργία, ο πιο κοινός όρος που χρησιμοποιείται είναι το πρότυπο-template. Ένα θέμα περιέχει πολλά αρχεία που είναι γνωστά στους σχεδιαστές ιστοσελίδων, συμπεριλαμβανομένων των style sheets, εικόνες, και JavaScript. Ένα θέμα μπορεί να περιλαμβάνει επίσης ορισμένες επεκτάσεις αρχείων με τα οποία μπορεί να μην είναι τόσο εξοικειωμένοι, π.χ. `.theme`, ή `.tpl.php`. Το πρώτο χρησιμοποιείται από καθαρά θέματα PHP, η τελευταία επέκταση εμφανίζεται σε θέματα που χρησιμοποιούν την μηχανή PHPTemplate που είναι ενσωματωμένη με το Drupal.

Μια μηχανή παραγωγής θεματικών παραλλαγών είναι μια συλλογή από σενάρια (scripts) και αρχεία που χρησιμεύουν για την ερμηνεία της γλώσσας προγραμματισμού που χρησιμοποιείται και να επεξεργάζεται τις εντολές που περιέχονται σε αυτήν. Καθώς τα δεδομένα που προέρχονται από τη βάση δεδομένων και από εξωτερικές πηγές (εάν υπάρχουν), η μηχανή θέματος προσαρμόζει τα δεδομένα σε μία προκαθορισμένη μορφή για απεικόνιση. Υπάρχουν αρκετές δημοφιλείς μηχανές θέματος, καθεμία από της οποίες έχει σχεδιαστεί για να ερμηνεύσει διαφορετικές γλώσσες template. Το Drupal διανέμεται με τον μηχανή θέματος PHPTemplate. Η PHPTemplate είναι δημοφιλής για ποικίλους λόγους, το λιγότερο εκ των οποίων είναι ότι η γλώσσα που βασίζεται είναι η PHP μια προτιμώμενη επιλογή για πολλούς προγραμματιστές Web σήμερα.

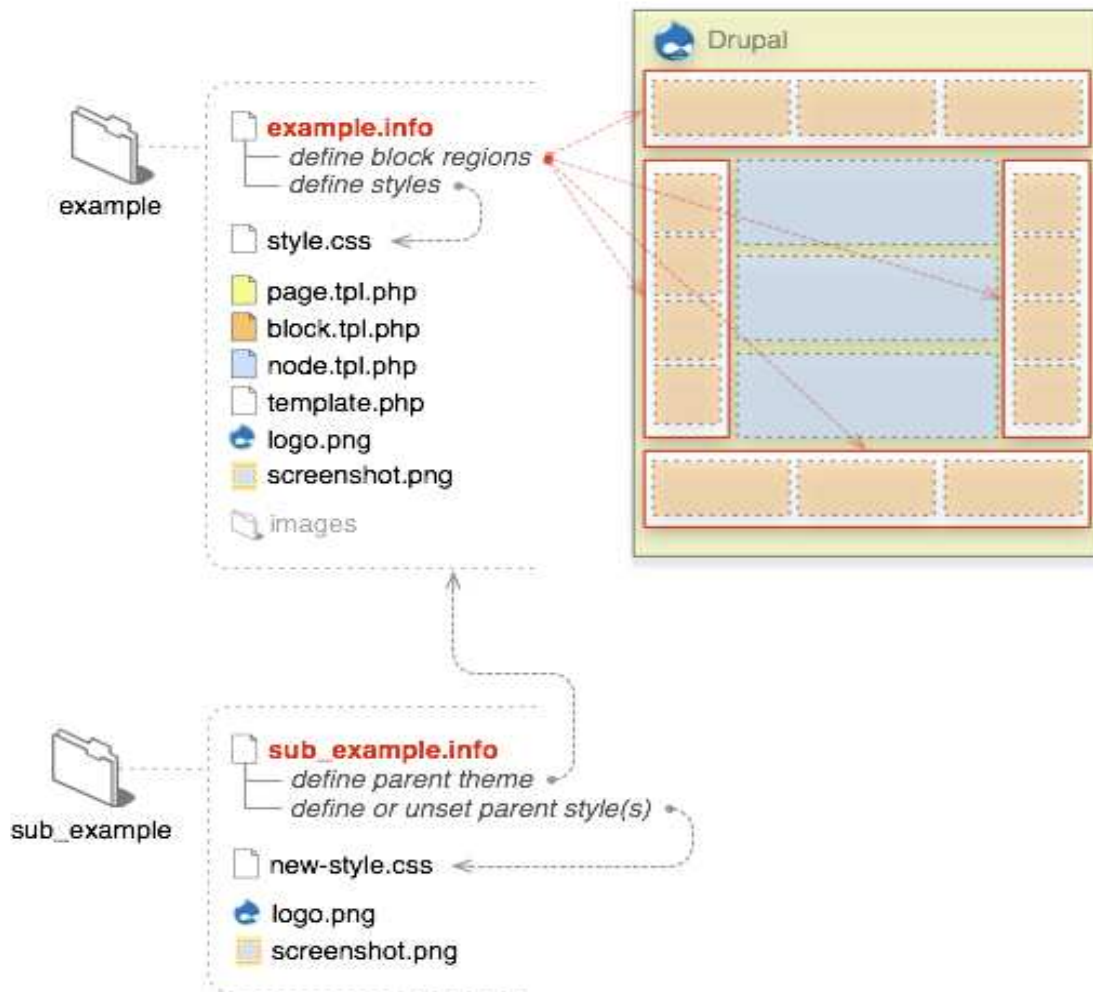


**Σχήμα 9: Στάδια theming του περιεχομένου μέχρι την παρουσίαση του**

- 1) Τα δεδομένα, ως επί το πλείστον, είναι αποθηκευμένο στη βασική τους μορφή στη βάση δεδομένων της Drupal εγκατάστασης. Η μορφοποίηση, αν υπάρχει, είναι παρούσα μόνο ως HTML ετικέτες που μπορεί να έχουν καθοριστεί στο περιεχόμενο από το συγγραφέα.
- 2) Το πρώτο σημαντικό βήμα στο δρόμο για την έξοδο συμβαίνει όταν ο πυρήνας του Drupal εξάγει και προ-επεξεργάζεται τα δεδομένα. Καμία πραγματική μορφοποίηση δεν συμβαίνει σε αυτό το επίπεδο. Οποιαδήποτε μορφοποίηση HTML που αναφέρεται στα στοιχεία που είναι αποθηκευμένα στη βάση απλά περνάει για την μετάφραση (interpreting) από το πρόγραμμα περιήγησης.
- 3) Το επόμενο βήμα στο δρόμο για την έξοδο βλέπει την μηχανή θέματος (theme engine) να αρχίζει να συγκεντρώνει την έξοδο από τον πυρήνα και τις μονάδες σε κάτι κοντά στην τελική του μορφή.
- 4) Το τελευταίο βήμα πριν από την έξοδο συμβαίνει όταν συγκεκριμένα αρχεία του θέματος επεξεργάζονται τα δεδομένα. Αυτό το τελευταίο στάδιο μπορεί να έχει ένα ευρύ φάσμα επιπτώσεων, από ελάχιστο έως πολύ σημαντικό. Η διακύμανση των επιπτώσεων εξαρτάται από το βαθμό στον οποίο ο συντάκτης του θέματος έχει δώσει συγκεκριμένες κατευθύνσεις για τη μορφοποίηση και αν ο συγγραφέας έχει επιλέξει να παρακάμψει τη μορφοποίηση της μηχανής θέματος ή των προεπιλεγμένων style sheets της εκάστοτε Drupal διανομής. Ότι και από τα δύο να συμβαίνει το περιεχόμενο παρουσιάζεται ανάλογα.

## Περίληψη των αρχείων ενός Drupal θέματος

Μια θεματική παραλλαγή είναι μια συλλογή αρχείων όπου ορίζουν το επίπεδο παρουσίασης. Κάθε χρήστης μπορεί να δημιουργήσει το δική του θεματική παραλλαγή (theme) ή υπό-θεματική παραλλαγή (sub-theme). Το μόνο που χρειάζεται είναι το .info αρχείο αλλά τα περισσότερα themes ή sub themes χρησιμοποιούν και άλλα αρχεία. Στο παρακάτω διάγραμμα παρουσιάζονται τα διάφορα αρχεία που υπάρχουν σε ένα τυπικό theme ή sub theme.



Σχήμα 10: Συλλογή αρχείων σε ένα τυπικό Drupal theme

### .info (required)

Αυτό που απαιτεί το Drupal για να δει ένα θέμα είναι το .info αρχείο. Ότι απαιτείται από το θέμα βρίσκεται μέσα σε αυτό αρχείο, όπως style sheets, JavaScript's, block regions και πολλά ακόμη περιγράφονται σε αυτό το αρχείο.

### template files (.tpl.php)

Αυτά τα templates χρησιμοποιούνται για την (x)HTML και τις PHP μεταβλητές. Σε ορισμένες περιπτώσεις μπορεί να εξάγει άλλους τύπους δεδομένων - XML RSS για παράδειγμα.

Κάθε .tpl.php αρχείο χειρίζεται την παραγωγή ενός συγκεκριμένου themeable μέρους δεδομένων, και σε ορισμένες περιπτώσεις μπορεί να χειριστεί πολλαπλά αρχεία .tpl.php. Είναι προαιρετικά, και εάν δεν υπάρχει κανένα σε κάποιο θέμα τότε θα παραχθεί η προεπιλεγμένη έξοδος.

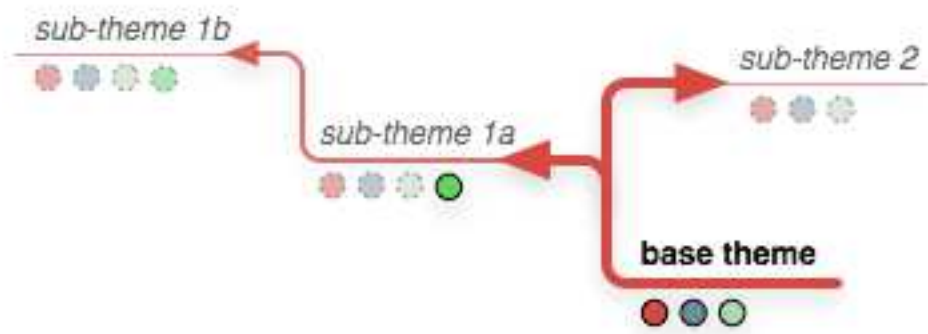
Τα αρχεία αυτά δεν πρέπει να έχουν περίπλοκη λογική. Στις περισσότερες περιπτώσεις, θα πρέπει να είναι καθαρές ετικέτες (x)HTML και PHP μεταβλητές.

#### template.php

Για όλες τις υπό όρους λογικές συνθήκες και την επεξεργασία των δεδομένων της εξόδου παραγωγής, υπάρχει το αρχείο template.php. Δεν απαιτείται, αλλά για να κρατήσει τα .tpl.php αρχεία τακτοποιημένα μπορεί να χρησιμοποιηθεί για τον έλεγχο του προεπεξεργαστεί για τη δημιουργία μεταβλητών πριν συγχωνευθούν με την HTML μέσα στα .tpl.php αρχεία. Προσαρμοσμένες συναρτήσεις, override συναρτήσεις θέματος ή οποιαδήποτε άλλη προσαρμογή των ακατέργαστων (raw) εξόδων θα πρέπει, επίσης, να γίνει εδώ.

#### sub themes

Στην επιφάνεια, ένα υπό-θέμα συμπεριφέρεται ακριβώς όπως ένα οποιοδήποτε άλλο θέμα. Οι μόνες διαφορές είναι ότι κληρονομούν τους πόρους από τα θέματα γονέων τους. Δεν υπάρχουν όρια στις δυνατότητες διασύνδεσης των επιμέρους θεμάτων με τους γονείς τους. Ένα υπό-θέμα μπορεί να είναι ένα παιδί ενός άλλου υπό-θέματος, και μπορεί να είναι διακλαδισμένα και οργανωμένα κατά απαίτηση. Αυτό είναι που δίνει στα υπό-θέματα μεγάλες δυνατότητες.

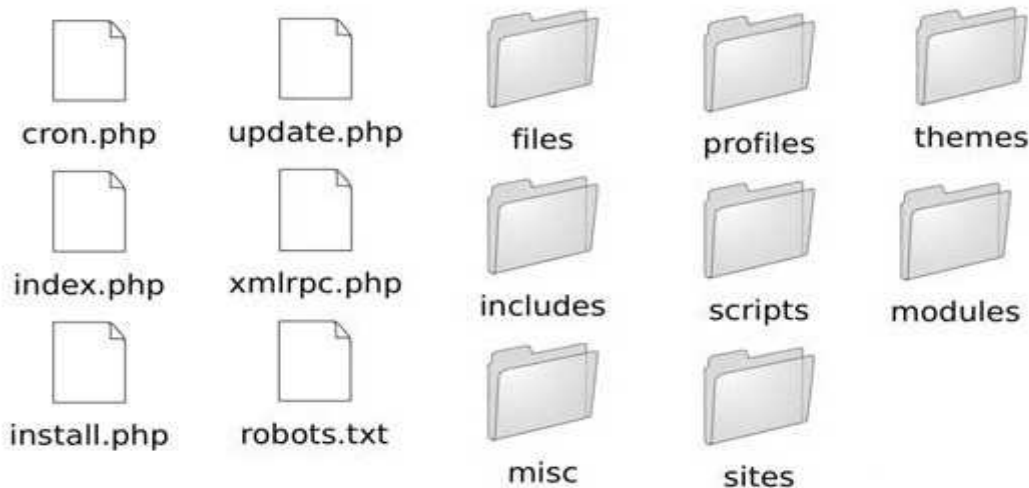


**Σχήμα 11: Διασύνδεση ενός βασικού θέματος με δύο υποθέματα**

### **3.1.14 Το σύστημα αρχείων του Drupal**

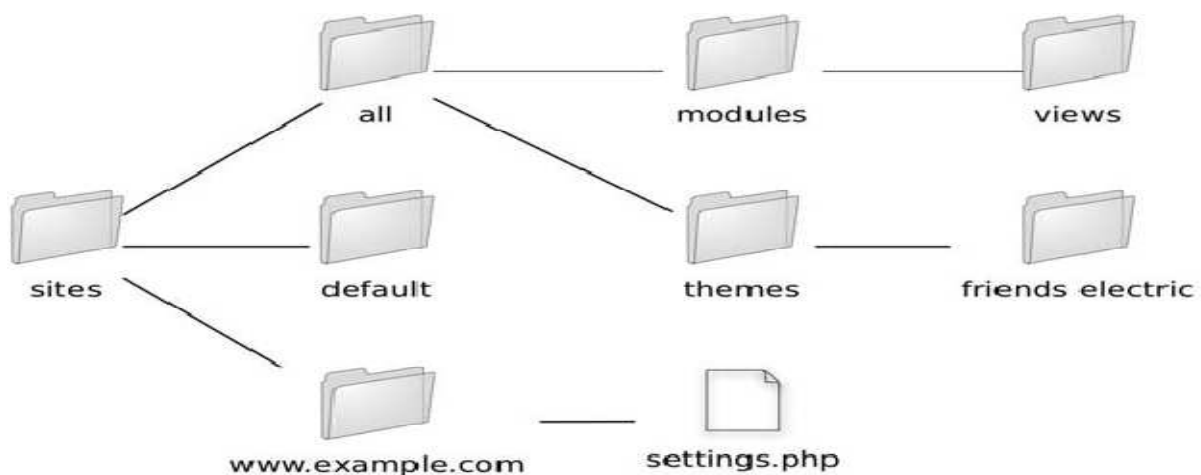
Η κατανόηση της δομής των προεπιλεγμένων καταλόγων της εγκατάστασης του Drupal βοηθάει στην αποσφαλμάτωση (debugging) του ιστότοπου και στην κατανόηση βέλτιστων πρακτικών, όπως το που πρέπει να εγκαθίσταστε κάθε μονάδα, διαμοιραζόμενη custom, αλλά και κάθε θέμα, κατασκευασμένο από εμάς ή από την κοινότητα, αλλά και το πώς μπορούμε να έχουμε διαφορετικά Drupal προφίλ. Μια τυπική εγκατάσταση του Drupal φαίνεται στο σχήμα 12. Ο φάκελος files δεν περιλαμβάνεται με Drupal από την αρχή, αλλά είναι απαραίτητος, για την αποθήκευση διαφόρων αρχείων όπως ένα προσαρμοσμένο λογότυπο, ενεργοποίηση avatar των χρηστών, ή για την αποθήκευση άλλων μέσων ενημέρωσης που σχετίζονται με τον ιστότοπο σας. Αυτός ο υποκατάλογος απαιτεί δικαιώματα ανάγνωσης και εγγραφής από τον web server όπου το Drupal τρέχει από πίσω. Ο φάκελος includes περιλαμβάνει βιβλιοθήκες των κοινών συναρτήσεων που χρησιμοποιεί το Drupal. Στο misc φάκελο αποθηκεύονται τα JavaScript αρχεία και διάφορα εικονίδια και εικόνες διαθέσιμες για την εγκατάσταση υλικού του Drupal. Ο φάκελος modules περιλαμβάνει τις βασικές μονάδες, με κάθε μονάδα στο δικό της φάκελο. Είναι καλύτερο να μην αγγίζετε τίποτα σε αυτό το φάκελο (επιπλέον μονάδες μπορούν να προστεθούν στον κατάλογο sites). Ο φάκελος profiles περιέχει τα διαφορετικά προφίλ για την εγκατάσταση ενός ιστότοπου.

Εάν υπάρχουν άλλα προφίλ, εκτός από το προεπιλεγμένο προφίλ σε αυτό υποκατάλογο, το Drupal θα ρωτήσει ποιό προφίλ θέλετε να εγκαταστήσετε κατά την πρώτη εγκατάσταση του Drupal ιστότοπου. Ο κύριος σκοπός του προφίλ εγκατάστασης είναι να ενεργοποιεί συγκεκριμένες μονάδες του πυρήνα και των διαμοιραζόμενων μονάδων αυτόματα. Ένα παράδειγμα θα μπορούσε να είναι ένα e-commerce προφίλ που αυτόματα μετατρέπει το Drupal ως πλατφόρμα ηλεκτρονικού εμπορίου.



**Σχήμα 12: Σύστημα αρχείων και καταλόγων του Drupal**

Ο φάκελος scripts περιέχει σενάρια για τον έλεγχο της σύνταξης, τον καθαρισμό του κώδικα, και το χειρισμό ειδικών περιπτώσεων με το cron Δεν χρησιμοποιείται κατά την διάρκεια του κύκλου ζωής αίτησης του Drupal. Ο φάκελος sites του σχήματος 12 περιέχει όλες τις τροποποιήσεις από τον χρήστη του Drupal, με τη μορφή των ρυθμίσεων, ενότητες και θέματα. Όταν προσθέτονται μονάδες στο Drupal είτε από τις διαμοιραζόμενες μονάδες είτε γράφοντας τη δική του ο χρήστης, αυτές αποθηκεύονται στη θέση sites/all/modules.



**Σχήμα 13: Ο κατάλογος sites του Drupal**

Αυτό κρατά όλες τις Drupal τροποποιήσεις του χρήστη σε ένα ενιαίο φάκελο. Μέσα στον κατάλογο sites θα πρέπει να υπάρχει ένας υποκατάλογος με το όνομα default που θα περιέχει το αρχείο των προεπιλεγμένων ρυθμίσεων για το Drupal ιστότοπο το αρχείο settings.php. Ο κατάλογος default συνήθως αντιγράφεται και μετονομάζεται στη διεύθυνση URL του ιστότοπου, έτσι ώστε το αρχείο ρυθμίσεων θα είναι στη διεύθυνση sites/www.example.com/settings.php. Τέλος ο φάκελος themes περιέχει τις μηχανές προτύπων και τα προεπιλεγμένα θέματα για το Drupal.

### 3.1.15 Η διαδικασία εξυπηρέτησης ενός αιτήματος από το Drupal-Serving a request

Έχοντας ένα θεωρητικό πλαίσιο για το τι συμβαίνει όταν μια αίτηση λαμβάνεται από το Drupal είναι χρήσιμη στην κατανόηση του πως δουλεύουν τα πράγματα πίσω από το Drupal.

**Ο ρόλος του Web Server:** Το Drupal τρέχει πίσω από έναν web server, συνήθως τον Apache. Αν ο web server “σέβεται” το αρχείο .htaccess του Drupal, ορισμένες PHP ρυθμίσεις αρχικοποιούνται και ενεργοποιούνται οι καθαρές διευθύνσεις URL. Το Drupal υποστηρίζει καθαρές διευθύνσεις URL, δηλαδή, διευθύνσεις URL που μοιάζουν με αυτές `http://example.com/foo/bar`. Η εντολή `mod_rewrite` στο αρχείο .htaccess του Drupal μεταφράζει τη διαδρομή σε `index.php?q=foo/bar`. Έτσι, στο εσωτερικό επίπεδο, το Drupal ασχολείται πάντα με το ίδιο μονοπάτι (που είναι αποθηκευμένο στο URL ερώτημα με παράμετρο το q) είτε οι καθαρές διευθύνσεις URL είναι ενεργοποιημένες είτε όχι. Σε αυτή την περίπτωση, η εσωτερική διαδρομή θα είναι `foo/bar`. Το εσωτερικό μονοπάτι ονομάζεται και Drupal μονοπάτι. Σε άλλους web servers, όπως το Microsoft IIS, οι καθαρές URL μπορεί να επιτευχθούν χρησιμοποιώντας μια Windows Internet Server Application Programming Interface (ISAPI) μονάδα, όπως η ISAPI\_Rewrite.

**Η Διαδικασία Bootstrap:** Το Drupal εκκινεί τον εαυτό του σε κάθε αίτημα περνώντας από μια σειρά φάσεων εκκίνησης. Αυτές οι φάσεις ορίζονται στην `bootstrap.inc` βιβλιοθήκη. Μερικές από αυτές περιγράφονται ως ακολούθως.

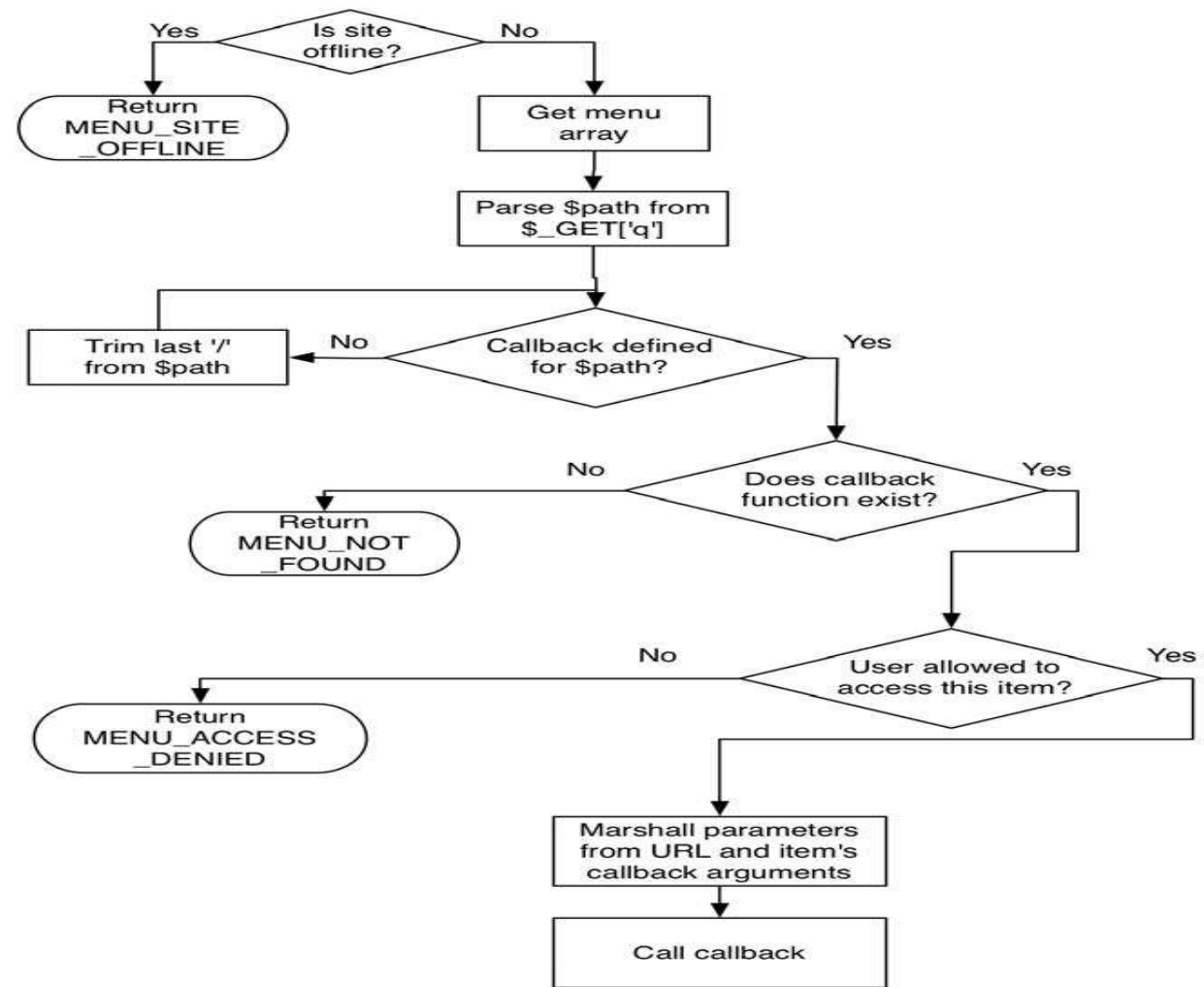
- 1) **Configuration:** Αυτή η φάση συμπληρώνει τον εσωτερικό πίνακα διαμόρφωσης του Drupal και καθορίζει το URL βάσης (`$base_url`) του δικτυακού τόπου. Το αρχείο `settings.php` αναλύεται μέσω της εντολής `include_once`, και κάθε μεταβλητή που καθορίστηκε να κάνει override εκεί, εφαρμόζεται.
- 2) **Προσωρινή Page Cache:** Σε καταστάσεις που απαιτούν υψηλό επίπεδο επεκτασιμότητας, ένα σύστημα προσωρινής αποθήκευσης μπορεί να χρειαστεί να κληθεί πριν από την σύνδεση με τη βάση δεδομένων. Η φάση προσωρινής αποθήκευσης επιτρέπει να συμπεριληφθεί (με την συνάρτηση `include()`) ένα αρχείο PHP που περιέχει μια συνάρτηση που ονομάζεται `page_cache_fastpath()`, η οποία αναλαμβάνει να επιστρέφει το περιεχόμενο στο πρόγραμμα περιήγησης. Η προσωρινή αποθήκευση ενεργοποιείται θέτοντας την μεταβλητή `page_cache_fastpath` σε TRUE, και το αρχείο που πρέπει να συμπεριληφθεί ορίζεται από τη ρύθμιση της μεταβλητής `cache_inc` στη διαδρομή του αρχείου.
- 3) **Μετάπειτα Page Cache:** Στη μετάπειτα page cache φάση, το Drupal φορτώνει αρκετό βοηθητικό κώδικα για να καθοριστεί το εάν θα “σερβιριστεί” ή όχι μια σελίδα από την page cache. Αυτό περιλαμβάνει συγχωνευμένες ρυθμίσεις από τη βάση δεδομένων στο πίνακα που δημιουργήθηκε κατά τη διάρκεια της φάσης διαμόρφωσης και τη φόρτωση ή την ανάλυση του κώδικα κάποιας μονάδας. Αν η σύνοδος δείχνει ότι η αίτηση δημιουργήθηκε από έναν ανώνυμο χρήστη και η προσωρινή αποθήκευση της σελίδας είναι ενεργοποιημένη, η σελίδα επιστρέφεται από τη μνήμη cache και η εκτέλεση σταματά.
- 4) **Βάση δεδομένων:** Κατά τη φάση της βάσης δεδομένων, ο τύπος της βάσης δεδομένων προσδιορίζεται, και μία αρχική σύνδεση γίνεται για να θα χρησιμοποιηθεί για τα ερωτήματα βάσης δεδομένων.
- 5) **Πρόσβαση:** Το Drupal επιτρέπει την απαγόρευση των hosts ανά-hostname/IP βασικής διεύθυνσης. Στη φάση της πρόσβασης, ένας γρήγορος έλεγχος για να δούμε αν η αίτηση προέρχεται από ένα απαγορευμένο υποδοχής? Αν ναι, δεν επιτρέπεται η πρόσβαση.
- 6) **Συνεδρία:** Το Drupal εκμεταλλεύεται τον ενσωματωμένο χειρισμό συνεδριάσεων της PHP, αλλά παρακάμπτει μερικούς handlers με το δικούς του για να μπορέσει να βάλει σε εφαρμογή βάσεις δεδομένων που υποστηρίζουν χειρισμό συνεδριάσεων. Οι συνεδρίες αρχικοποιούνται ή επανιδρύονται στη φάση της συνεδρίας.
- 7) **Διαδρομή:** Στη φάση της διαδρομής ο κώδικας που χειρίζεται τα μονοπάτια και τα ψευδώνυμα (alias) των μονοπατιών είναι φορτωμένος. Η φάση αυτή ενεργοποιεί τις αναγνώσιμες από τον άνθρωπο διευθύνσεις URL που πρέπει να επιλυθούν και διαχειρίζεται τα εσωτερικά Drupal μονοπάτια και τις αναζητήσεις.



- 8) **Full**: Στη φάση αυτή ολοκληρώνεται η διαδικασία εκκίνησης με την φόρτωση μιας βιβλιοθήκης κοινών συναρτήσεων όπως, υποστήριξης θεματικών παραλλαγών, την υποστήριξη για τη χαρτογράφηση επανάκλησης (callback), διαχείριση αρχείων, Unicode, PHP εργαλείων εικόνας, δημιουργία και επεξεργασία φορμών, δυνατότητα ταξινόμησης πινάκων αυτόματα, και το αποτέλεσμα σελιδοποίησης. Ο προεπιλεγμένος χειριστής σφαλμάτων του Drupal έχει οριστεί, το locale έχει οριστεί, και φορτώνονται όλες οι ενεργές μονάδες. Τέλος, το Drupal πυροδοτεί το init hook, έτσι ώστε οι μονάδες να έχουν την ευκαιρία να κοινοποιηθούν πριν από την επίσημη διεκπεραίωση της αίτησης που αρχίζει. Μόλις το Drupal ολοκληρώσει το bootstrapping, όλα τα συνθετικά μέρη του framework είναι διαθέσιμα. Είναι καιρός να αναλάβει το αίτημα του browser και να το περάσει στην PHP συνάρτηση που θα το χειριστεί. Η χαρτογράφηση μεταξύ των URL και των συναρτήσεων που τις χειρίζονται επιτυγχάνεται με τη χρήση ενός μητρώου επανάκλησης που φροντίζει τόσο την χαρτογράφηση των διευθύνσεων URL όσο και του ελέγχου πρόσβασης.

### 3.1.16 Το σύστημα πλοήγησης-The Menu System

Όταν ένα πρόγραμμα περιήγησης υποβάλλει μια αίτηση στο Drupal, δίνει μια διεύθυνση URL στο Drupal. Από αυτές τις πληροφορίες, το Drupal πρέπει να καταλάβει τι κώδικα να τρέξει και πώς να χειριστεί την αίτηση. Αυτό είναι κοινώς γνωστό ως dispatching. Το Drupal τεμαχίζει τη URL και κρατάει το τελευταίο μέρος, που ονομάζεται μονοπάτι. Για παράδειγμα, αν το URL είναι <http://example.com/?q=node/3>, το Drupal μονοπάτι είναι node/3.



Σχήμα 14: Επισκόπηση της διαδικασίας αποστολής του μενού

### Αντιστοίχιση URL σε συναρτήσεις

Το Drupal ζητά από όλες τις μονάδες να παρέχουν ένα πίνακα από στοιχεία μενού-δηλαδή, ένα μονοπάτι και κάποιες πληροφορίες σχετικά με αυτό το μονοπάτι. Ένα από τα κομμάτια των πληροφοριών που μια μονάδα πρέπει να δώσει είναι ένα callback (ένα callback είναι μια αναφορά σε ένα κομμάτι του εκτελέσιμο κώδικα που έχει περάσει ως όρισμα σε άλλο κώδικα). Ένα callback στα πλαίσια του Drupal είναι απλά το όνομα μιας συνάρτησης PHP που θα τρέξει όταν το πρόγραμμα περιήγησης ζητά μια συγκεκριμένη διαδρομή. Τα βήματα που ακολουθεί το Drupal όταν γίνεται κάποια αίτηση είναι τα εξής:

- 1) Αν η διαδρομή είναι ένα ψευδώνυμο για μια πραγματική διαδρομή, το Drupal βρίσκει την πραγματική διαδρομή και χρησιμοποιεί αντί αυτής. Για παράδειγμα, εάν ένας διαχειριστής έχει δώσει το ψευδώνυμο `http://example.com/?q=cats` στη διεύθυνση `http://example.com/?q=node/3`, το Drupal χρησιμοποιεί το `node/3` ως μονοπάτι.
- 2) Εκτελεί το `hook_menu()`, έτσι ώστε όλες οι μονάδες να μπορούν να παρέχουν τα callbacks τους.
- 3) Δημιουργεί ένα χάρτη από μονοπάτια (όπως `node/add`) σε callbacks (PHP συναρτήσεις όπως `node_page()`).
- 4) Αν η μονάδα `menu.module` είναι ενεργοποιημένη, εφαρμόζει οποιεσδήποτε αλλαγές ή προσθήκες που ο διαχειριστής του site έχει κάνει στο χάρτη (όπως το να παρακάμπτει τον τίτλο στοιχείου του μενού).
- 5) Χρησιμοποιεί τον χάρτη για να αναζητήσει την συνάρτηση επανάκλησης για τη ζητούμενη διεύθυνση URL, και την καλεί. Αν ζητούνται και ορίσματα στις συναρτήσεις επανάκλησης το Drupal τα στέλνει και αυτά μαζί.
- 6) Επιστρέφει το αποτέλεσμα της συνάρτησης ή ένα "Access denied" μήνυμα, αν ο χρήστης δεν μπορεί να έχει πρόσβαση στη διεύθυνση URL, ή ένα 404 μήνυμα, εάν η διαδρομή δεν αντιστοιχεί σε κάποια συνάρτηση.

### **3.1.17 Το abstract επίπεδο της βάσης δεδομένων του Drupal-The database abstract layer**

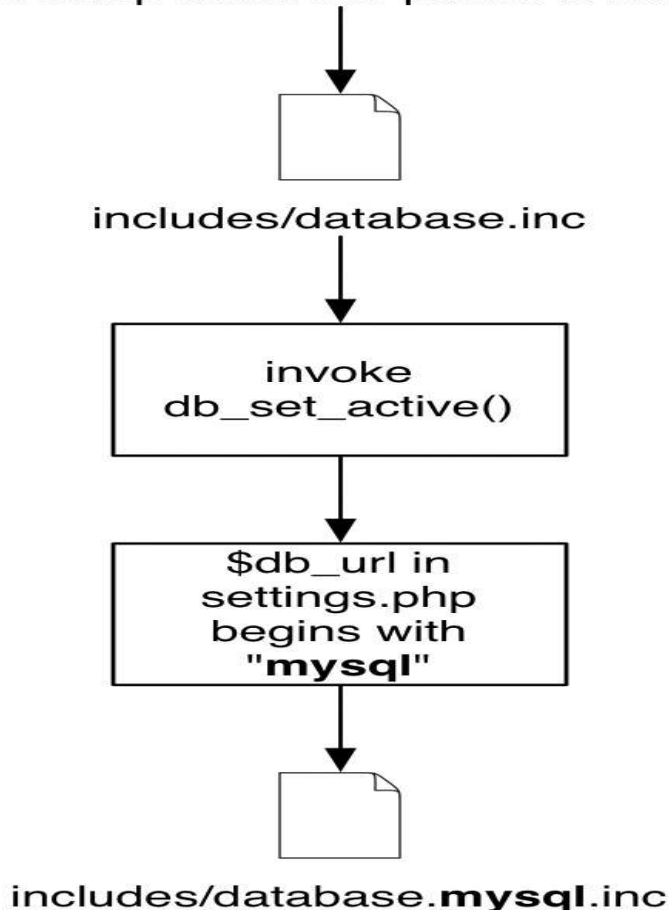
Το Drupal εξαρτάται από μια βάση δεδομένων για να λειτουργήσει σωστά. Μέσα στο Drupal, υπάρχει ένα ελαφρύ επίπεδο αφαιρετικότητας μεταξύ της βάσης δεδομένων και του κώδικα του Drupal άλλα και του custom κώδικα. Το Drupal γνωρίζει σε ποια βάση δεδομένων να συνδεθεί και τι το όνομα χρήστη και τι κωδικό πρόσβασης να χρησιμοποιήσει κατά τη δημιουργία μιας σύνδεσης στη βάση δεδομένων με αναζήτηση στο αρχείο `settings.php`. Η γραμμή που ορίζει τη σύνδεση στη βάση δεδομένων μοιάζει με αυτό: `$db_url = 'mysql://username:password@localhost/databasename'`; Αυτό το παράδειγμα είναι για τη σύνδεση με μια βάση δεδομένων MySQL. Για τη σύνδεση με μια βάση δεδομένων PostgreSQL το πρόθεμα σύνδεσης της συμβολοσειράς θα ήταν με `pgsql` αντί για `mysql`.

#### The database abstract layer:

Πολλές φορές χρειάζεται να αλλάξουμε από μια βάση δεδομένων σε μια άλλη βάση δεδομένων αλλά αυτό απαιτεί και την μετατροπή των εντολών από τη μια βάση στην άλλη. Με ένα επίπεδο αφαιρετικότητας δεν χρειάζεται πλέον να μετατρέπονται οι εντολές και τα ονόματα των συναρτήσεων για διαφορετικά συστήματα βάσεων δεδομένων, εφ' όσον όμως τα ερωτήματά (queries) αυτά είναι συμβατά με το πρότυπο ANSI SQL, έτσι δεν χρειάζεται να γραφτούν ξεχωριστά ερωτήματα για ξεχωριστά συστήματα. Για παράδειγμα, αντί να καλούνται η `mysql_query()` ή η `pg_query()` αντίστοιχα για την κάθε βάση δεδομένων, το Drupal χρησιμοποιεί την `db_query()`, η οποία διατηρεί το εμπορικό επίπεδο βάσης δεδομένων άγνωστο.

Το επίπεδο αφαιρετικότητας της βάσης δεδομένων του Drupal είναι ελαφρύ και εξυπηρετεί δύο βασικούς σκοπούς. Ο πρώτος είναι να κρατήσει τον κωδικό συμπαγή και να συνδέεται με οποιαδήποτε βάση δεδομένων. Ο δεύτερος είναι να αποστειρώνουν τα καταχωρημένα δεδομένα του χρήστη που τοποθετούνται σε ερωτήματα (queries) για την πρόληψη SQL εκχύσεων (SQL injections). Αυτό το επίπεδο χτίστηκε με βάση την αρχή ότι η γραφή SQL είναι πιο βολική από την εκμάθηση μιας νέας γλώσσας αφαιρετικού επιπέδου.

**Bootstrap database phase invoked**



**Σχήμα 15: Καθορισμός του database αρχείου που θα περιληφθεί με βάση την τιμή της \$db\_url**

Το Drupal καθορίζει τον τύπο της βάσης δεδομένων προς σύνδεση με τον έλεγχο της μεταβλητής \$ db\_url μέσα στο αρχείο settings.php. Για παράδειγμα, αν το \$ db\_url ξεκινά με mysql, τότε το Drupal θα συμπεριλάβει το αρχείο includes/database.mysql.inc. Αν αρχίζει με pgsql, το Drupal θα συμπεριλάβει το αρχείο includes/database.pgsql.inc (βλ. σχήμα 15). Το Drupal δημιουργεί αυτόματα μια σύνδεση με τη βάση δεδομένων ως μέρος της κανονικής διαδικασίας εκκίνησης του.

### 3.1.18 Συνεδρίες-Sessions

Το HTTP είναι ένα πρωτόκολλο χωρίς καταστάσεις (stateless-Στην επιστήμη των υπολογιστών, ένα πρωτόκολλο χωρίς καταστάσεις είναι ένα πρωτόκολλο επικοινωνίας που αντιμετωπίζει κάθε αίτηση ως ανεξάρτητη συναλλαγή που δεν έχει καμία σχέση με οποιαδήποτε προηγούμενη αίτηση, έτσι ώστε η επικοινωνία αποτελείται από ανεξάρτητα ζεύγη αιτήσεων και απαντήσεων) πράγμα που σημαίνει ότι κάθε αλληλεπίδραση μεταξύ του web browser και του server γίνεται μόνο μια φορά. Πώς γίνεται όμως να παρακολουθείται ένας χρήστης, όταν αυτός πλοηγείται μέσα από μια σειρά από ιστοσελίδες σε μια ιστοσελίδα;

Εδώ αναλαμβάνουν οι συνεδρίες. Ξεκινώντας με την έκδοση 4, η PHP προσφέρει ενσωματωμένη υποστήριξη για συνεδρίες μέσω της οικογένειας συναρτήσεων συνεδριών. Ας δούμε πώς το Drupal χρησιμοποιεί τις συνεδρίες της PHP.

Όταν ένα πρόγραμμα περιήγησης αιτείται μια σελίδα από ένα Drupal ιστότοπο για πρώτη φορά, η PHP δημιουργεί ένα cookie για το πρόγραμμα περιήγησης που περιέχει ένα προεπιλεγμένο τυχαίο 32-χαρακτήρων ID, που ονομάζεται PHPSESSID. Αυτό γίνεται με την προσθήκη μιας γραμμής στις κεφαλίδες απάντησης της HTTP που στέλνονται στον browser την πρώτη φορά που επισκέπτεται την ιστοσελίδα.

```
HTTP/1.1 200 OK
Date: Wed, 17 Jan 2007 20:24:58 GMT
Server: Apache/1.3.33 (Darwin) PHP/5.1.6
Set-Cookie: PHPSESSID=3sulj1mainvme55r8udcc6j2a4; expires=Fri, 09 Feb 2007 23:58:19 GMT; path=/
Last-Modified: Wed, 17 Jan 2007 20:24:59 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
```

### **Εικόνα 1: Κεφαλίδα απάντησης της HTTP με το PHPSESSID προς τον browser**

Στις επόμενες επισκέψεις στο site, το πρόγραμμα περιήγησης εμφανίζει το cookie στον διακομιστή περιλαμβάνοντας αυτό σε κάθε αίτημα HTTP.

```
GET / HTTP/1.1
User-Agent=Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-US; rv:1.8.1.1) Gecko/20061204 Firefox/2.0.0.1
Cookie: PHPSESSID=3sulj1mainvme55r8udcc6j2a4
```

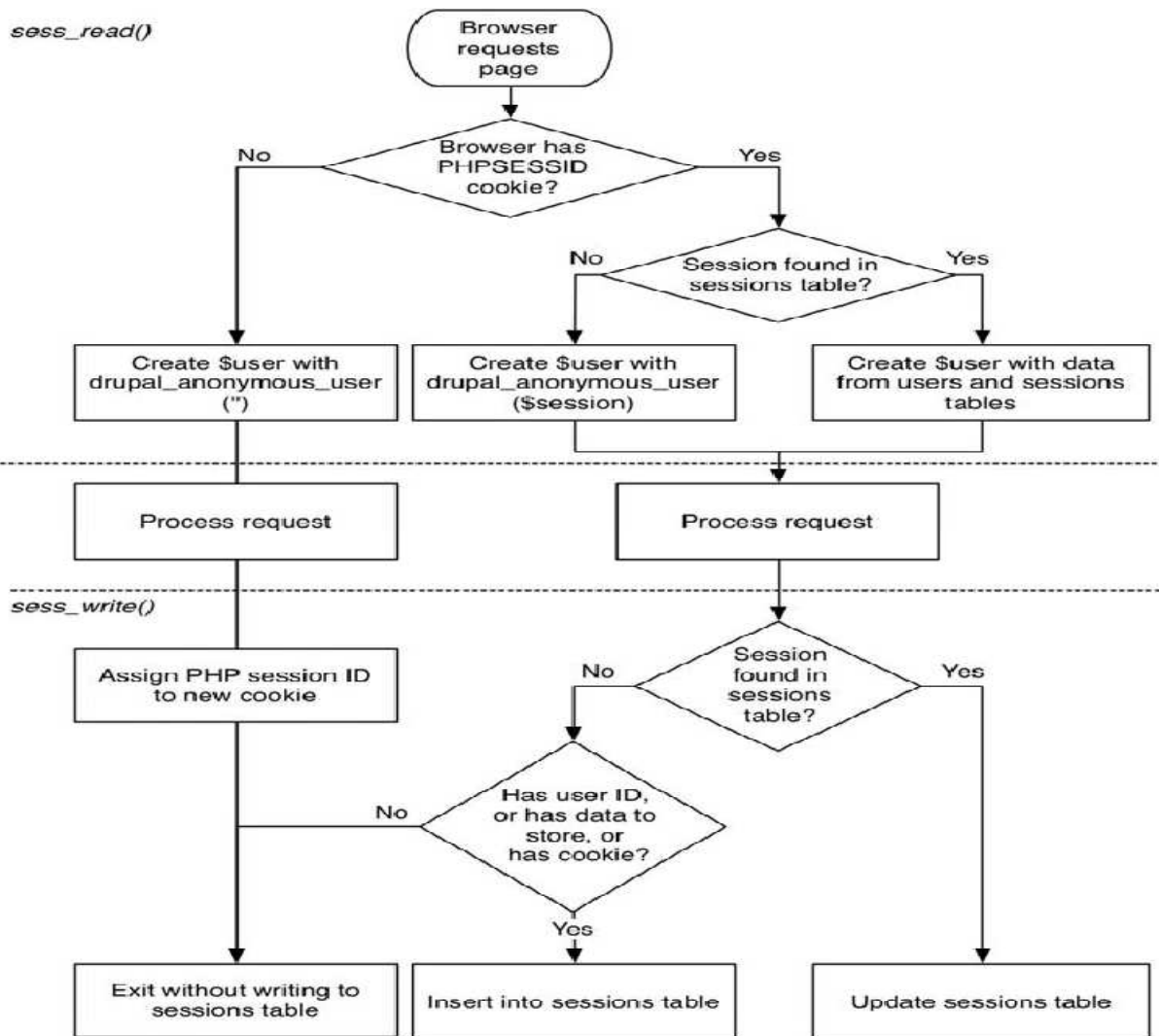
### **Εικόνα 2: Αίτηση HTTP προς τον server με το PHPSESSID που αποδόθηκε από πριν**

Αυτό επιτρέπει στην PHP να παρακολουθεί ένα πρόγραμμα περιήγησης καθώς επισκέπτεται την ιστοσελίδα. Το 32-χαρακτήρων ID, γνωστό ως αναγνωριστικό περιόδου (session ID), χρησιμοποιείται ως κλειδί στις πληροφορίες που το Drupal αποθηκεύει σχετικά με τη συνεδρία, και επιτρέπει στο Drupal να συνδέει συνεδρίες με μεμονωμένους χρήστες.

### Κύκλος ζωής μιας συνεδρίας-Session life cycle

Ο κύκλος ζωής μιας συνεδρίας ξεκινά όταν ο browser κάνει μια αίτηση στο διακομιστή. Κατά τη διάρκεια της φάσης του DRUPAL\_BOOTSTRAP\_SESSION μεταξύ των ρουτινών εκκίνησης του Drupal (βλέπε includes/bootstrap.inc) ο κωδικός για τις συνεδρίες ξεκινά. Εάν το πρόγραμμα περιήγησης δεν παρουσιάσει ένα cookie το οποίο είχε προηγουμένως λάβει από την ιστοσελίδα, το σύστημα διαχείρισης συνεδριάσεων της PHP θα δώσει στο πρόγραμμα περιήγησης ένα νέο cookie με ένα νέο PHP αναγνωριστικό συνεδρίας (PHP session ID). Αυτό το ID είναι συνήθως μια αναπαράσταση 32-χαρακτήρων ενός μοναδικού MD5 hash (Η MD5 Message-Digest Algorithm είναι μια ευρέως χρησιμοποιούμενη κρυπτογραφική συνάρτηση κατακερματισμού που παράγει μια 128-bit (16-byte) hash τιμή), αν και η PHP 5 επιτρέπει τη ρύθμιση των παραμέτρων της συνεδρίας. Στη συνέχεια το Drupal ελέγχει τον πίνακα συνεδριών για την ύπαρξη μιας σειράς με το αναγνωριστικό συνεδρίας ως κλειδί. Αν βρεθεί, τότε η συνάρτηση sess\_read() ανακτά τα δεδομένα της συνεδρίας και εκτελεί ένα SQL JOIN στη γραμμή από τον πίνακα συνεδριών και στην αντίστοιχη γραμμή από τον πίνακα των χρηστών.

Το αποτέλεσμα αυτού του συνδέσμου είναι ένα αντικείμενο που περιέχει όλα τα πεδία και τιμές από τις δύο σειρές. Αυτή είναι το καθολικό \$ user αντικείμενο που χρησιμοποιείται σε όλο το υπόλοιπο του Drupal. Έτσι, τα δεδομένα της συνεδρίας είναι επίσης διαθέσιμα με αναζήτηση στο \$ user αντικείμενο, ειδικά με το \$ user-> session. Αλλά τι θα συμβεί αν δεν υπάρχει χρήστης με όνομα χρήστη που να ταιριάζει με το όνομα χρήστη στη συνεδρία; Επειδή το Drupal συνεργάζεται με μια γραμμή στον πίνακα των χρηστών με User ID το 0, και επειδή σε μη εξουσιοδοτημένους χρηστές (“anonymous”) αποδίδεται το uid 0 στον πίνακα συνεδριών, το SQL .JOIN λειτουργεί πάντα.



Σχήμα 16: Κύκλος ζωής μιας συνεδρίας στο Drupal -Session life cycle

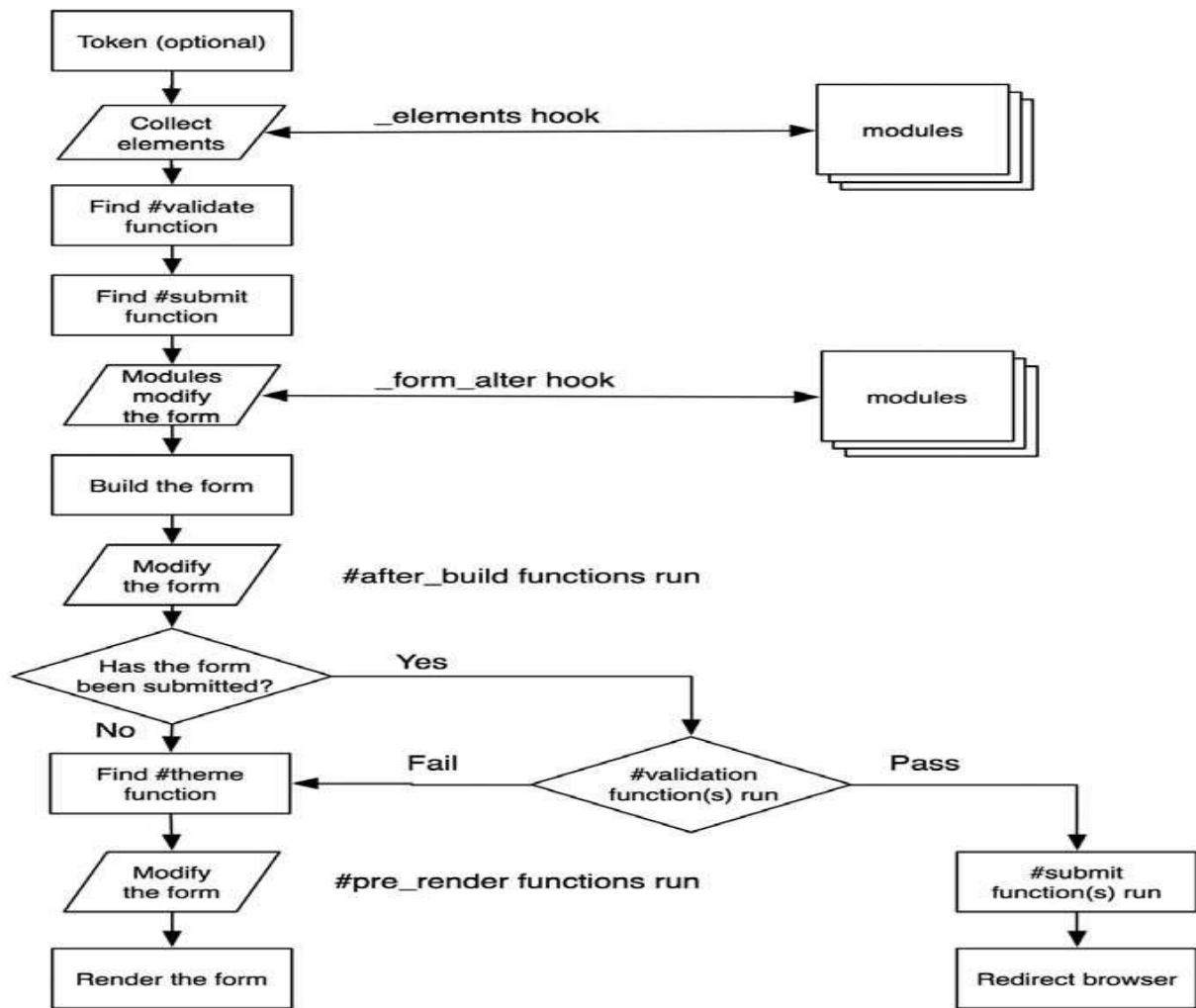
Όταν η ιστοσελίδα έχει παραδοθεί στο πρόγραμμα περιήγησης, το τελευταίο βήμα είναι να κλείσει η συνεδρία. Η PHP επικαλείται τη συνάρτηση sess\_write () από το includes/session.inc, γράφει οτιδήποτε είναι αποθηκευμένο στο \$\_SESSION (κατά τη διάρκεια της αίτησης) στον πίνακα συνεδριών. Η εξαίρεση σε αυτό είναι εάν ο αιτών δεν δέχεται cookies. Σε αυτή την περίπτωση καμία γραμμή δεν θα γραφτεί στον πίνακα συνεδριών. Ο λόγος για αυτό είναι να αποτρέψει τον πίνακα από γέμισμα με γραμμές που δημιουργούνται από προγράμματα ανίχνευσης ιστού, καθώς το μέγεθος του πίνακα μπορεί να επηρεάσει την απόδοση.

### 3.1.19 Το Form API του Drupal

Το Drupal διαθέτει μια διεπαφή προγραμματισμού εφαρμογών (API) για την παραγωγή, την επικύρωση και την επεξεργασία HTML φορμών. Το Form API δημιουργεί φόρμες σε ένα ένθετο πίνακα από ιδιότητες και τις τιμές. Ο πίνακας κατόπιν μεταφράζεται (rendered) από την μηχανή form-rendering σε κατάλληλο χρόνο, ενώ η σελίδα παράγεται. Υπάρχουν πολλές επιπτώσεις αυτής της προσέγγισης:

- 1) Αντί να παράγουμε εμείς τον HTML κώδικα εξόδου, δημιουργείται ένας πίνακας και αναλαμβάνει η μηχανή form-rendering να παράγει τον HTML κώδικα.
- 2) Από την στιγμή που έχουμε να κάνουμε με μια αναπαράσταση της φόρμας ως δομημένων δεδομένων, μπορούμε να προσθέσουμε, να διαγράψουμε, να αλλάξουμε τη σειρά των στοιχείων, αλλά και τη φόρμα. Αυτό είναι ιδιαίτερα χρήσιμο όταν θέλουμε να τροποποιήσουμε μια φόρμα που δημιουργήθηκε από μια διαφορετική μονάδα με ένα καθαρό και διακριτικό τρόπο.
- 3) Κάθε στοιχείο της φόρμας μπορεί να αντιστοιχηθεί σε κάθε συνάρτηση θέματος.
- 4) Επιπρόσθετη επικύρωση ή επεξεργασία μπορεί να προστεθεί σε οποιαδήποτε φόρμα.
- 5) Οι λειτουργίες με τις φόρμες προστατεύονται από επιθέσεις έκχυσης, όπου ένας χρήστης τροποποιεί μια φόρμα και στη συνέχεια προσπαθεί να την υποβάλει.
- 6) Η καμπύλη εκμάθησης για τη χρήση φορμών είναι λίγο απότομη!

Για να αλληλεπιδράσουμε με το Form API έξυπνα, είναι χρήσιμο να γνωρίζουμε πώς η μηχανή πίσω από το API λειτουργεί. Για να γίνει αυτό πρέπει να δούμε πως δουλεύει η συνάρτηση `drupal_get_form()`. Η συνάρτηση `drupal_get_form()` ξεκινά με την αρχικοποίηση της μεταβλητής `$form_values` (ο πίνακας που κρατάει τις τιμές που έχουν υποβληθεί) σε ένα κενό πίνακα και της μεταβλητής `$form_submitted` σε `FALSE`. Ένα από τα πλεονεκτήματα του συστήματος φορμών είναι ότι προσπαθεί να διασφαλίσει ότι η φόρμα που υποβάλλεται είναι στην πραγματικότητα η φόρμα που δημιουργήθηκε από το Drupal. Για να το κάνει αυτό το Drupal θέτει ένα ιδιωτικό κλειδί σε κάθε εγκατάσταση του Drupal. Μόλις το κλειδί δημιουργείται, αποθηκεύεται στον πίνακα μεταβλητών ως `drupal_private_key`. Ένα ψευδοτυχαίο διακριτικό με βάση το ιδιωτικό κλειδί αποστέλλεται στη φόρμα σε ένα κρυφό πεδίο και δοκιμάζεται όταν υποβάλλεται η φόρμα. Ένα κρυφό πεδίο που περιέχει το Form ID της τρέχουσας φόρμας στέλνεται στον browser ως μέρος της φόρμας. Αυτό το αναγνωριστικό αντιστοιχίζεται συνήθως με τη συνάρτηση που καθορίζει τη φόρμα και αποστέλλεται ως η πρώτη παράμετρος της `drupal_get_form()`. Για παράδειγμα, η `user_register()` συνάρτηση καθορίζει τη φόρμα εγγραφής χρήστη, και καλείται με αυτόν τον τρόπο: `$output = drupal_get_form('user_register');` Στη συνέχεια, καλείται η `element_info()`. Αυτή επικαλείται το `hook_elements()` σε όλες τις μονάδες που έχει εφαρμοστεί. Μέσα στον πυρήνα του Drupal, τα τυποποιημένα στοιχεία, όπως radio buttons και check boxes, ορίζονται από `system.module` εφαρμογή του `hook_elements()`. Οι μονάδες εφαρμόζουν αυτό το hook αν θέλουν να καθορίσουν τους δικούς τους τύπους στοιχείων. Μια συνάρτηση επικύρωσης για μια φόρμα μπορεί να αποδοθεί με τον καθορισμό της ιδιότητας `#validate` στη φόρμα σε ένα πίνακα με το όνομα της συνάρτησης ως κλειδί και με ένα πίνακα ως τιμή. Η συνάρτηση που χειρίζεται την υποβολή μιας φόρμας μπορεί να ανατεθεί με τον καθορισμό της ιδιότητας `#submit` στη φόρμα σε ένα πίνακα με το όνομα της συνάρτησης που θα χειριστεί την υποβολή της φόρμας ως το κλειδί και ένα πίνακα από τιμές που πρέπει να περάσει ως παραμέτρους. Αν η ιδιότητα `$form['#theme']` έχει οριστεί σε μια υπάρχουσα συνάρτηση, το Drupal απλά τη χρησιμοποιεί. Εάν όχι, η `theme_get_function()` καλείται προκειμένου να καθοριστεί εάν υπάρχει μια συνάρτηση θέματος διαθέσιμη για αυτή τη φόρμα.



Σχήμα 17: Πως το Drupal χειρίζεται τις φόρμες

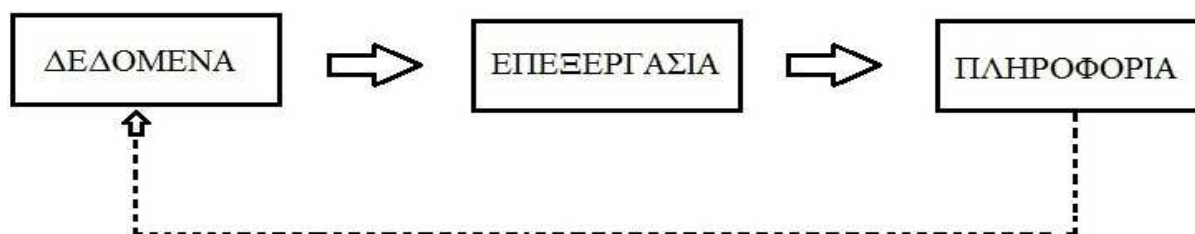
Για να μετατραπεί μια φόρμα δέντρου από ένα ένθετο πίνακα σε κώδικα HTML, ο μηχανισμός παραγωγής της φόρμας καλεί την συνάρτηση `drupal_render()`. Αυτή η αναδρομική συνάρτηση περνά μέσα από κάθε επίπεδο του δέντρου φόρμας και εκτελεί τις ακόλουθες ενέργειες:

- 1) Προσδιορίζει αν η ιδιότητα `#children` έχει οριστεί. Αν όχι, αποδίδει στα παιδιά αυτού του κόμβου του δέντρου ως εξής
  - Προσδιορίζει αν η συνάρτηση `#theme` έχει οριστεί για αυτό το στοιχείο.
  - Αν ναι, προσωρινά θέτει το `#type` του στοιχείου σε `markup`. Έπειτα περνάει αυτό το στοιχείο στη συνάρτηση θέματος και ξαναθέτει το στοιχείο στην τιμή που είχε πριν.
  - Εάν δεν έχει παραχθεί καθόλου περιεχόμενο κάθε ένα από τα παιδιά αυτού του στοιχείου γίνεται `render` με τη σειρά.
- 2) Εάν αυτό το στοιχείο δεν έχει ακόμη τυπωθεί, καλέστε το στοιχείο `renderer` για τον `#type` του στοιχείου αυτού. Αν ο τύπος `#type` του στοιχείου αυτού δεν έχει οριστεί, η προεπιλογή θα είναι `markup`.
- 3) Εισάγεται πρώτο το πρόθεμα `#prefix` και προσαρτεί το `#suffix` στο περιεχόμενο, και το επιστρέφει από τη συνάρτηση.

## 4. Κύριο Μέρος Πτυχιακής

### 4.1 Ανάλυση προβλήματος

Με τον όρο Πρόβλημα εννοείται μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της όμως δεν είναι γνωστή, ούτε προφανής. Στην παρούσα πτυχιακή εργασία το πρόβλημα είναι η κατασκευή ενός ηλεκτρονικού καταστήματος και πιο συγκεκριμένα μιας ηλεκτρονικής βίντεο λέσχης. Κάθε πρόβλημα έχει κάποια δεδομένα. Δεδομένα είναι τα στοιχεία του προβλήματος που θεωρούνται γνωστά ή δοσμένα και πάνω σε αυτά μπορούμε να στηριχτούμε για να την εύρεση του ζητούμενου. Τα δεδομένα είναι αδιαμφισβήτητα στοιχεία του προβλήματος που λειτουργούν σαν βάση για περαιτέρω σκέψεις και ενέργειες. Τα δεδομένα στην παρούσα πτυχιακή είναι το σύστημα διαχείρισης περιεχομένου Drupal με όλες τις επιμέρους, ήδη υπάρχουσες και μη, δυνατότητες (μονάδες, θέματα, δυνατότητες προγραμματισμού κ.α.) που προσφέρει καθώς και το περιεχόμενο που πρόκειται να χρησιμοποιηθεί. Τώρα από την επεξεργασία των δεδομένων με κατάλληλο τρόπο θα παραχθούν οι κατάλληλες πληροφορίες. Πληροφορία είναι κάθε γνωστικό στοιχείο που προέρχεται από την επεξεργασία των δεδομένων.



Σχήμα 18: Σχέση δεδομένων πληροφοριών

Η πληροφορία που παράγεται από την επεξεργασία των δεδομένων ενδέχεται να είναι και το τελικό ζητούμενο του προβλήματος. Το ζητούμενο είναι αυτό που ψάχνουμε να βρούμε στο πρόβλημα, δηλαδή το σημείο στο οποίο θέλουμε να φτάσουμε προκειμένου να βγούμε από την προβληματική κατάσταση. Τέλος σημαντικός είναι ο καθορισμός και η ανάλυση των απαιτήσεων. Αυτό ορίζεται ως η διαδικασία που ακολουθούμε προκειμένου να ανιχνεύσουμε τα δεδομένα που μας δίνει το πρόβλημα και να αποσαφηνίσουμε ποια είναι τα ζητούμενα που ψάχνουμε να βρούμε. Η ανάλυση των απαιτήσεων απαντά στο τι θα κάνει το ηλεκτρονικό κατάστημα αδιαφορώντας για το πως θα γίνει η υλοποίηση αυτού, τι τεχνικές υλοποίησης θα χρησιμοποιηθούν, τί γλώσσες προγραμματισμού κτλ.

Οι απαιτήσεις χωρίζονται σε λειτουργικές και μη λειτουργικές. Οι λειτουργικές απαιτήσεις του ηλεκτρονικού καταστήματος περιλαμβάνουν τα εξής:

- 1) Πλήρης διαχείριση-έλεγχος του συστήματος:
  - Δυνατότητα πλήρους διαχείρισης της βάσης δεδομένων του συστήματος.
  - Δυνατότητα πλήρους διαχείρισης των μονάδων του συστήματος.
  - Δυνατότητα συντήρησης του ιστότοπου.
- 2) Δυνατότητα ενημέρωσης του συστήματος:
  - Ενημέρωση του συστήματος σε θέματα ασφάλειας.
  - Ενημέρωση του συστήματος σε θέματα μονάδων.
  - Ενημέρωση της βάσης δεδομένων του συστήματος.
- 3) Ασφαλή είσοδος/έξοδος στο/από το σύστημα:
  - Χρήση του πρωτόκολλου SSL.
  - Διασφάλιση των προσωπικών κωδικών των χρηστών.
- 4) Πλήρης διαχείριση χρηστών:
  - Δυνατότητα εγγραφής/διαγραφής χρήστη.
  - Ανάθεση ρόλων σε χρήστες.
  - Ανάθεση δικαιωμάτων σε χρήστες ανάλογα με τον ρόλο τους.
  - Καταγραφή επισκέψεων χρήστη ανά σελίδα.
  - Διαχωρισμός χρηστών σε εγγεγραμμένους και μη.



- 5) Καλάθι αγορών:
  - Καταγραφή ποσότητας και τιμής προϊόντων και δυνατότητα αλλαγής παραγγελίας.
  - Φόρμα παραγγελίας.
  - Υπολογισμός αθροίσματος προϊόντων.
  - Δυνατότητα πληρωμών μέσω Paypal.
- 6) Πλήρης διαχείριση παραγγελιών:
  - Δυνατότητα επεξεργασίας παραγγελίας.
  - Ιστορικό παραγγελιών ανά χρήστη.
  - Ενημέρωση παραγγελιών.
- 7) Σαφής διαχωρισμός περιεχομένου:
  - Δυνατότητα δημιουργίας τύπων περιεχομένου.
  - Διαχωρισμός προϊόντων ανά κατηγορία.
- 8) Απλότητα στην δημιουργία περιεχομένου:
  - Φόρμα δημιουργίας περιεχομένου.
  - Μηχανισμός WYSIWYG.
  - Δυνατότητα προεπισκόπησης.
- 9) Δυνατότητα εμπλουτισμού του περιεχομένου:
  - Δυνατότητα εισαγωγής επιπρόσθετων πεδίων στη φόρμα δημιουργίας περιεχομένου.
  - Δυνατότητα εισαγωγής επιπρόσθετων χαρακτηριστικών στα προϊόντα.
- 10) Δυνατότητα ενημέρωσης του περιεχομένου:
  - Ανά πάσα χρονική στιγμή επεξεργασία του περιεχομένου.
  - Ανά πάσα χρονική στιγμή εισαγωγή διαγραφή περιεχομένου.
- 11) Αυτόματη κατηγοριοποίηση περιεχομένου:
  - Κατηγοριοποίηση του περιεχομένου κατά την υποβολή της φόρμας δημιουργίας του.
  - Κατηγοριοποίηση ανάλογα των λέξεων κλειδιών που εισάγονται στα πεδία της φόρμας.
- 12) Δυνατότητα ταξινόμησης περιεχομένου ανά λέξη κλειδί:
  - Κάθε προϊόν ανήκει σε μια κατηγορία ανάλογα τις λέξεις κλειδιά που κατέχει.
  - Κάθε χρήστης μπορεί να ταξινομή τα προϊόντα ανά κατηγορία βάση των λέξεων κλειδιών.
- 13) Ευδιάκριτη μορφοποίηση του περιεχομένου:
  - Στοίχιση περιεχομένου χωρίς επικαλύψεις.
  - Στοίχιση φωτογραφιών περιεχομένου.
  - Στοίχιση κειμένου περιεχομένου.
- 14) Απλότητα στη χρήση:
  - Όχι ογκώδη πολύπλοκα μενού.
  - Λίγα και απλά βήματα μέχρι την ολοκλήρωση κάθε ενέργειας στον ιστότοπο.
  - Άμεση πλοήγηση χωρίς ενδιάμεσες σελίδες.
- 15) Απλή και σύνθετη αναζήτηση περιεχομένου:
  - Παροχή μηχανής αναζήτησης για το περιεχόμενο του ιστότοπου.
  - Δυνατότητα επιλογής επιπρόσθετων χαρακτηριστικών αναζήτησης.

Στις μη λειτουργικές απαιτήσεις περιλαμβάνονται τα εξής:

- 1) Σχεδίαση θέματος και λογότυπου.
- 2) Φόρμα επικοινωνίας.
- 3) Slide show.
- 4) Δυνατότητα σχολιασμού περιεχομένου.
- 5) Τόπος δημόσιας συζήτησης.
- 6) Δυνατότητα εγγραφής στο newsletter.
- 7) Αξιολόγηση περιεχομένου.
- 8) RSS.
- 9) Χρήση των Widgets Datepicker και Tabs της jQuery

### 4.1.2 Απαιτήσεις του συστήματος

Οι απαιτήσεις του συστήματος διαχείρισης περιεχομένου Drupal διαφέρουν ανάλογα με το αν θα εγκατασταθεί τοπικά ή σε κάποιον απομακρυσμένο Server. Στην περίπτωση που εγκατασταθεί τοπικά



**Εικόνα 3: XAMPP**

στον υπολογιστή μας το Drupal, θα πρέπει πρώτα να κάνουμε την απαραίτητη προετοιμασία. Το Drupal όπως και πολλά άλλα open source CMS χρειάζεται αρχικά έναν web server για να τρέξει. Μπορούμε να χρησιμοποιήσουμε είτε τον Apache (1.3 ή 2.x), είτε τον IIS της Microsoft, αλλά όπως προτείνουν και οι δημιουργοί του Drupal η καλύτερη λύση είναι η πρώτη. Επίσης το Drupal χρειάζεται και την γλώσσα PHP, προτείνοντας κάποια έκδοση από την 5.2 και μετά. Τέλος το τρίτο απαραίτητο συστατικό που πρέπει να έχουμε στον υπολογιστή μας, είναι ένας database server. Το Drupal δουλεύει είτε με MySQL είτε με PostgreSQL, συστήνοντας ως καλύτερη λύση κάποια έκδοση της MySQL από την 4.1 και μετά. Και τα τρία βασικά συστατικά που χρειαζόμαστε (Apache, PHP, MySQL) είναι εργαλεία open source τα οποία μπορούμε να τα βρούμε δωρεάν στο δίκτυο. Αντί όμως να τα κατεβάζουμε το κάθε ένα χωριστά και να τα κάνουμε ξεχωριστά εγκατάσταση, μπορούμε να διαλέξουμε την λύση του XAMPP. Το XAMPP αποτελεί στην ουσία ένα πακέτο, το οποίο περιλαμβάνει τις τελευταίες εκδόσεις του Apache, της PHP και της MySQL, ενώ περιλαμβάνει επίσης και άλλα τρία χρήσιμα εργαλεία, που θα χρειαστούμε στην συνέχεια (PHPmyAdmin, Filezilla Server, Mercury Mail). Το XAMPP διατίθεται και αυτό δωρεάν από την σελίδα <http://www.apachefriends.org> για διάφορα λειτουργικά συστήματα (Linux, Windows, Solaris, Mac).

Στην περίπτωση που εγκατασταθεί σε ένα απομακρυσμένο sever, το Drupal, θα χρειαστεί ένας λογαριασμός στον απομακρυσμένο server ένας λογαριασμός στον database server και ένας ftp client. Όσον αφορά τον web server που είναι εγκατεστημένος ο ιστότοπος της παρούσας πτυχιακής χρησιμοποιήθηκαν τα εξής λογισμικά: Για την ασφαλή σύνδεση με τον web server και την ασφαλή διαχείριση της βάσης δεδομένων (καθώς και αντιγράφων ασφαλείας) χρησιμοποιήθηκε το PuTTY, ένα ελεύθερο και ανοιχτού κώδικα λογισμικό εξομοίωσης τερματικού που δρα ως SSH client. Για την μεταφορά αρχείων από και προς τον server χρησιμοποιήθηκε το WinSCP. Το WinSCP (Windows Secure CoPy) είναι ένα ελεύθερο και ανοιχτού κώδικα λογισμικό για μεταφορά αρχείων μέσω των πρωτοκόλλων SFTP, SCP και FTP client για τα Microsoft Windows. Η κύρια λειτουργία του είναι ασφαλής μεταφορά αρχείων μεταξύ ενός τοπικού και ενός απομακρυσμένου υπολογιστή.

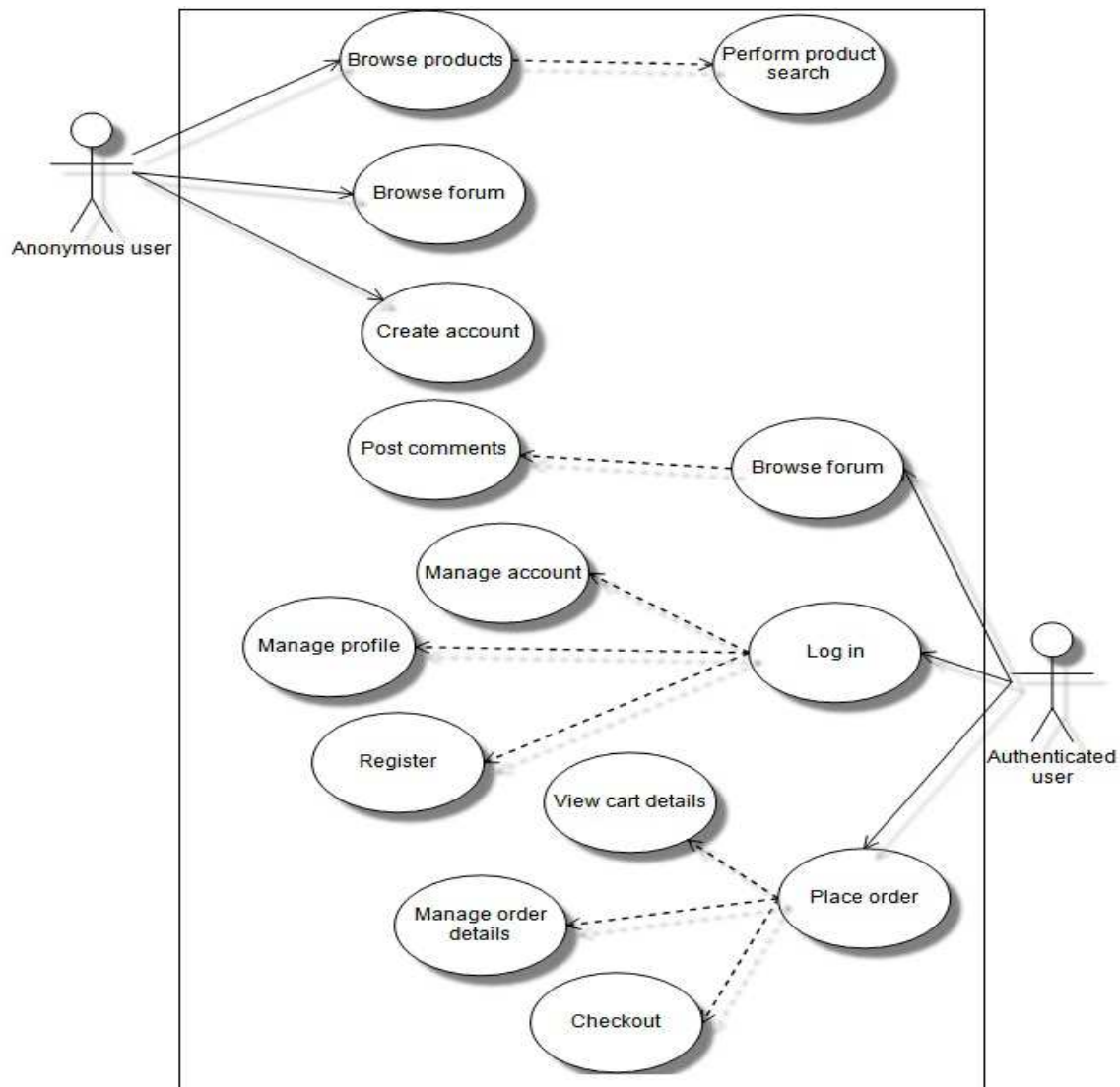


**Εικόνα 4: WinSCP -PuTTY**

Τελικό βήμα είναι η εγκατάσταση του Drupal. Το Drupal “ζει” στον δικτυακό τόπο <http://drupal.org>. Από εκεί κατεβάζουμε τα απαραίτητα αρχεία για την εγκατάσταση του Drupal. Ανάμεσα στις διάφορες εκδόσεις επιλέγεται η κατάλληλη για το project που θα υλοποιηθεί. Για το παρών project επιλέχτηκε η έκδοση 6.22 όμως πάντα υπάρχει μια ενημέρωση ασφάλειας σε μια νεότερη έκδοση έτσι είναι υποχρεωτικό να γίνεται συνεχώς αναβάθμιση του συστήματος σε επόμενες εκδόσεις. Επίσης η εγκατάσταση του καλαθιού αγορών Ubercart είναι απαραίτητη. Τέλος, όσον αφορά την διαδικασία εγκατάστασης όλων των εφαρμογών, συμπεριλαμβανομένου του Drupal και του Ubercart, υπάρχει αφθονία διαδικτυακών τόπων με αναλυτικά βήματα τόσο στα ελληνικά όσο και στα αγγλικά, ενδεικτικά αναφέρονται μερικοί: <http://drupal.org/documentation/install> <http://www.edutorials.gr> <http://drupal.org/documentation/install/windows> <http://learnbythedrop.com/drop/95> <http://mydrupal.gr/forum/10>

## 4.2 Σχεδιασμός υλοποίησης

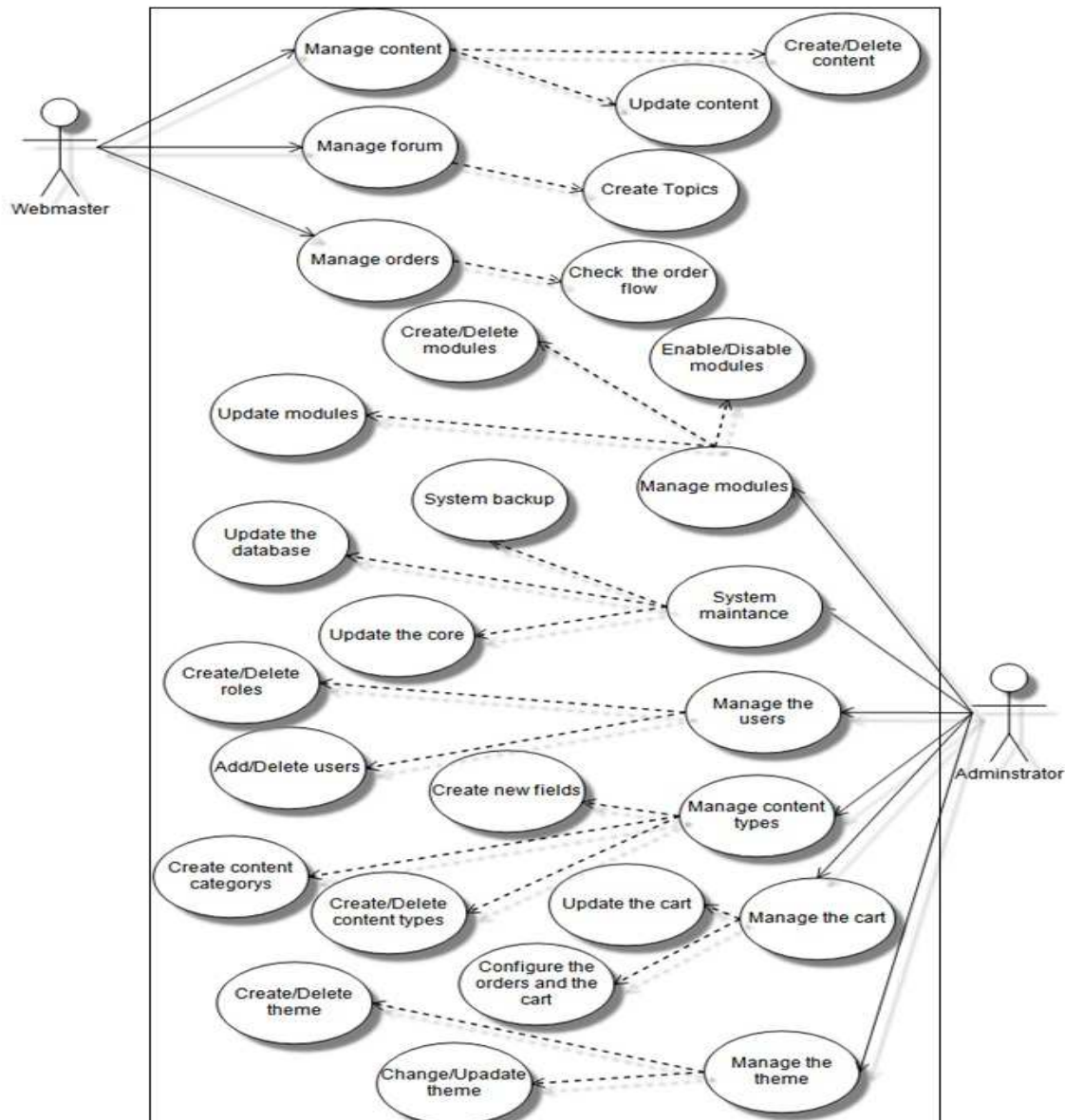
Κατά την φάση της σχεδίασης του συστήματος αναλύεται το πώς θα λειτουργεί το σύστημα. Στην παρούσα πτυχιακή εργασία το σύστημα είναι ένα ηλεκτρονικό κατάστημα. Το σύστημα διαχωρίζει τους χρήστες σε εγγεγραμμένους και μη. Κάθε κατηγορία χρηστών έχει διαφορετικά δικαιώματα. Κάθε μη εγγεγραμμένος χρήστης έχει δικαίωμα να περιηγηθεί μεταξύ των σελίδων του ηλεκτρονικού καταστήματος και να χρησιμοποιήσει τις λειτουργίες που προσφέρει το σύστημα αλλά όχι το δικαίωμα αγορών. Αυτό είναι δικαίωμα μόνο των εγγεγραμμένων χρηστών, αυτό είναι και το βασικό κριτήριο διαχωρισμού μεταξύ των χρηστών. Επίσης υπάρχουν συγκεκριμένοι ρόλοι για άλλες κατηγορίες χρηστών.



**Σχήμα 19: Δικαιώματα εγγεγραμμένου χρήστη και μη**

Υπάρχει ο ρόλος του webmaster (μπορεί να υπάρχουν και περισσότεροι από ένας) ο οποίος έχει τα δικαιώματα να κατηγοριοποιεί να ενημερώνει να διαγράφει και να εμπλουτίζει το περιεχόμενο του ηλεκτρονικού καταστήματος να διαχειρίζεται τις παραγγελίες του ηλεκτρονικού καταστήματος και τον τόπο δημόσιας συζήτησης. Υπάρχει ο ρόλος του administrator ο οποίος έχει τα δικαιώματα για την καθολική διαχείριση του συστήματος. Ο διαχειριστής μπορεί να συντηρεί τον ιστότοπο εφαρμόζοντας τις απαραίτητες ενημέρωσης τόσο για τον πυρήνα του συστήματος όσο και για τις μονάδες την βάση δεδομένων του καλαθιού αγορών καθώς και των θεματικών παραλλαγών.

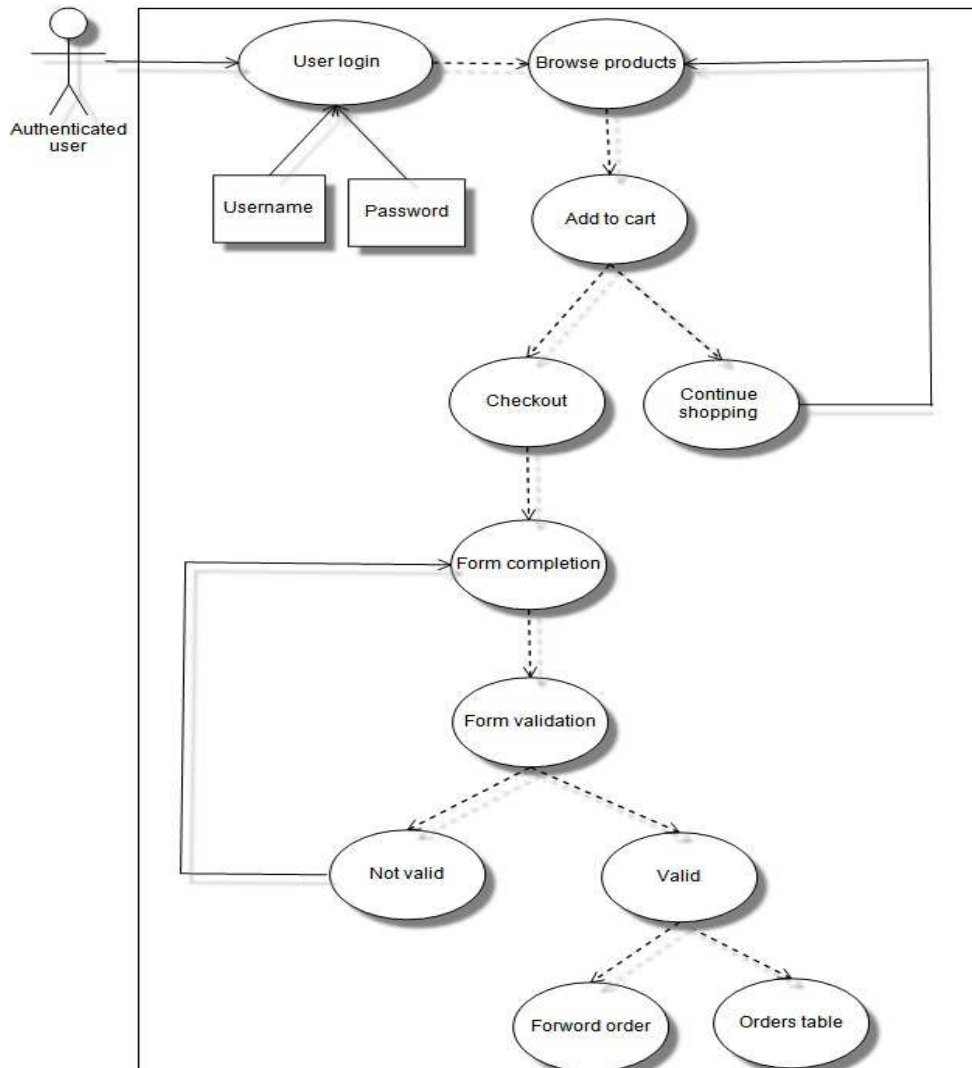
Ο ρόλος του διαχειριστή καλύπτει επίσης την δημιουργία μονάδων την ενημέρωση την ενεργοποίηση ή την απενεργοποίηση αυτών αλλά και την διαδικασία δημιουργίας αντιγράφων ασφαλείας τόσο των αρχείων του συστήματος όσο και της βάσης δεδομένων. ο διαχειριστής επίσης μπορεί να δημιουργεί νέους τύπους περιεχομένου καθώς και νέα πεδία σε ήδη υπάρχον τύπους. Είναι υπεύθυνος για την παραμετροποίηση του καταθίου αγορών αλλά και για την ενημέρωσή του. Τέλος ο διαχειριστής είναι αυτός ο οποίος δημιουργεί τους ρόλους που μπορεί να έχει κάθε χρήστης του συστήματος και κατά συνέπεια πλήρη έλεγχο στους ρόλους αυτούς. Υπάρχουν φυσικά πολλές ακόμη λειτουργίες για τις οποίες είναι υπεύθυνος ο διαχειριστής αλλά αυτές είναι οι πιο σημαντικές.



Σχήμα 20: Δικαιώματα administrator και webmaster

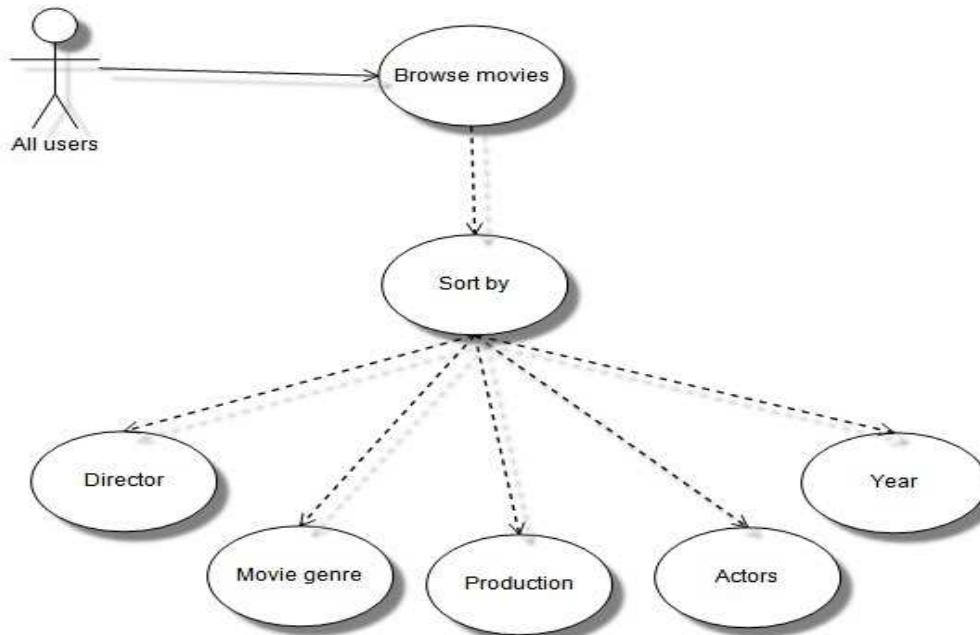
Η διαδικασία μιας παραγγελίας προϊόντων από τους πιστοποιημένους χρήστες είναι ένα σημαντικό θέμα όσον αφορά την αποτελεσματικότητα και την λειτουργία του ηλεκτρονικού καταστήματος. Το πρώτο βήμα της διαδικασίας είναι η ασφαλής και επιτυχής εισαγωγή του χρήστη στο σύστημα. Αυτό επιτυγχάνεται με την υποβολή του ονόματος και του συνθηματικού του χρήστη από την πλευρά του και με την πιστοποίηση του από την πλευρά του συστήματος. Έπειτα από την επιλογή κάποιου προϊόντος προς αγορά από τον χρήστη το προϊόν καταχωρείται στο καλάθι αγορών του συστήματος, έχοντας πάντα την δυνατότητα διαγραφής του ή την δυνατότητα εισαγωγής επιπλέον προϊόντων.

Επόμενο βήμα, και αφού έχουν επιλεγεί τα προς αγορά προϊόντα, είναι το “ταμείο”. Σε αυτό το βήμα συμπληρώνονται από το χρήστη όλα τα απαραίτητα πεδία της φόρμας παραγγελίας όπως όνομα διεύθυνση και τα λοιπά, επιλέγεται ο τρόπος πληρωμής, στην παρούσα πτυχιακή ο μόνος τρόπος πληρωμής είναι μέσω Paypal, και προσθέτονται τυχόν σχόλια για την παραγγελία. Επίσης αναφέρονται λεπτομερώς τα προϊόντα που έχουν επιλεγεί αλλά και η τελική τιμή τους. Στο επόμενο βήμα ελέγχονται τα πεδία της φόρμας ως προς την εγκυρότητά τους, αν δηλαδή έχουν συμπληρωθεί όλα τα απαραίτητα πεδία κατά έγκυρο τρόπο, δεν μπορεί το πεδίο για το όνομα για παράδειγμα να περιέχει αριθμητικούς χαρακτήρες. Εάν τα στοιχεία της φόρμας είναι έγκυρα τότε η παραγγελία ολοκληρώνεται και ένα αντίγραφο της αποθηκεύεται στην βάση δεδομένων και ένα αποστέλλεται στον πιστοποιημένο χρήστη μαζί με τις πληροφορίες παραλαβής, ημερομηνία κ.α.



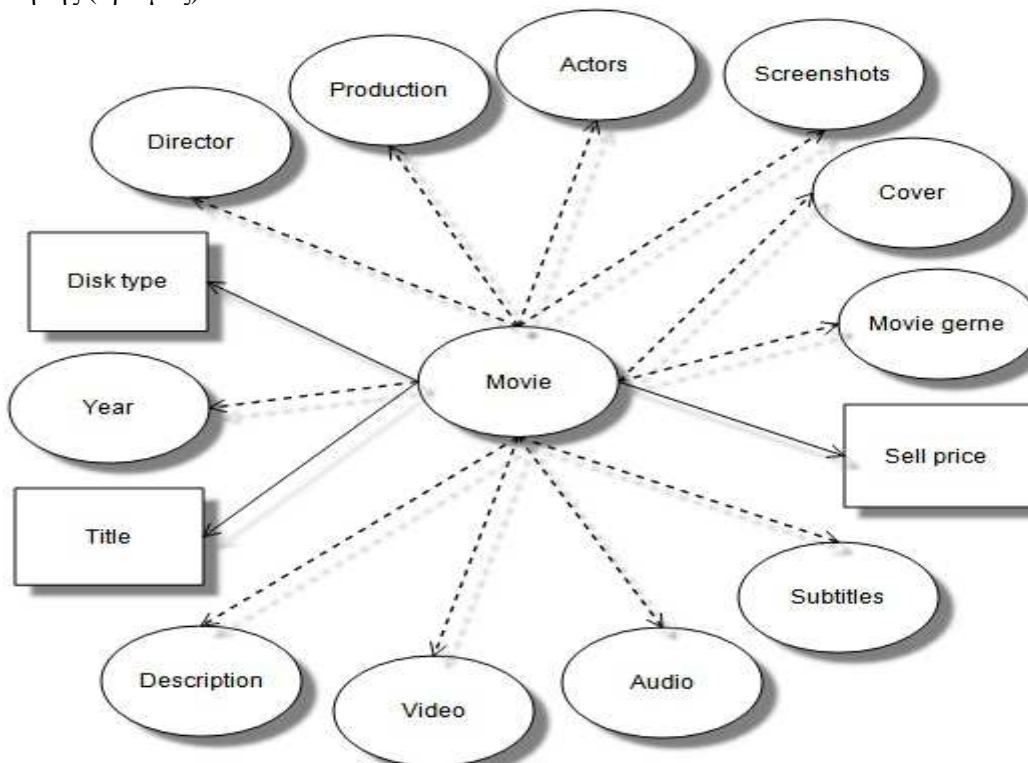
**Σχήμα 21: Διαδικασία παραγγελίας πιστοποιημένου χρήστη**

Μια ακόμη σημαντική λειτουργία είναι η κατηγοριοποίηση των προϊόντων. Κάθε προϊόν απαρτίζεται από κάποια πεδία. Τα προϊόντα του ηλεκτρονικού καταστήματος της παρούσας πτυχιακής είναι ταινίες. Κάθε ταινία αποτελείται από κάποια κοινά χαρακτηριστικά με όλες τις άλλες ταινίες. Τα κοινά αυτά χαρακτηριστικά εκμεταλλεύονται κατάλληλα για την δημιουργία της δυνατότητας ταξινόμησης των ταινιών βάση των χαρακτηριστικών αυτών. Τα κοινά αυτά χαρακτηριστικά είναι οι ηθοποιοί της ταινίας ο σκηνοθέτης το έτος παραγωγής η εταιρία παραγωγής το και είδος της. Κάθε τέτοιο χαρακτηριστικό θα μετατρέπεται αυτόματα σε λέξη κλειδί δίνοντας την δυνατότητα σε οποιονδήποτε χρήστη να ταξινομεί να αναζητεί αλλά και να κατηγοριοποιεί το περιεχόμενο κατά βούληση.



**Σχήμα 22: Κατηγοριοποίηση προϊόντων ανά χαρακτηριστικό**

Ο σχεδιασμός του ηλεκτρονικού καταστήματος απαιτεί την ύπαρξη κάποιου βασικού τύπου περιεχομένου. Αυτός δεν είναι άλλος από τον τύπο ταινία (movie). Ο τύπος αυτός θα πρέπει να περιλαμβάνει τα εξής πεδία: τίτλο (κείμενο-υποχρεωτικό), περιγραφή (κείμενο), διάρκεια (αριθμός), χαρακτηριστικά βίντεο (αλφαριθμητικό), χαρακτηριστικά ήχου (αλφαριθμητικό), υπότιτλους (κείμενο), τύπο δίσκου (κείμενο-υποχρεωτικό) είδος ταινίας (κείμενο λέξη-κλειδί), ηθοποιούς (κείμενο λέξη-κλειδί), σκηνοθέτη (κείμενο λέξη-κλειδί), στούντιο παραγωγής (κείμενο λέξη-κλειδί), έτος παραγωγής (αριθμός λέξη-κλειδί), εξώφυλλο (φωτογραφία), screenshots (φωτογραφία) και τιμή πώλησης (αριθμός).



**Σχήμα 23: Πεδία βασικού τύπου περιεχομένου movie**

### 4.3 Υλοποίηση

Κατά την φάση της υλοποίησης του συστήματος γίνεται η υλοποίηση των βημάτων που περιγράφηκαν στη φάση της σχεδίασης. Καθοριστικό ρόλο παίζει ο διαχωρισμός των χρηστών. Πριν καταχωρηθούν δικαιώματα σε κάθε χρήστη θα πρέπει πρώτα κάθε χρήστης να πάρει κάποιο ρόλο. Για την δημιουργία των ρόλων κάθε χρήστη ακολουθούνται τα εξής βήματα. Μετά από την εισαγωγή στο σύστημα ως user/1 από το μενού διαχείρισης επιλέγουμε διαχείριση χρηστών (user management) και από κει ρόλους. Υπάρχουν ήδη οι ρόλοι anonymous user και authenticated user όμως χρειαζόμαστε και ένα ρόλο διαχειριστή και ένα ρόλο webmaster. Είναι τακτική του Drupal να δημιουργείται ένας ρόλος διαχειριστή με τα ίδια ακριβώς δικαιώματα που έχει ο user/1 σε περίπτωση που κάτι πάει λάθος με των διαχειριστή να μπορεί να επέμβει ο user/1.

Permission	anonymous user	authenticated user	administrator	webmaster
<b>node module</b>				
access content	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
administer content types	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
administer nodes	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
create page content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
create search content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
create simplenews content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
create story content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
create webform content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete any page content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete any search content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete any simplenews content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete any story content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete any webform content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
delete own page content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete own search content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete own simplenews content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete own story content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete own webform content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
delete revisions	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit any page content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit any search content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit any simplenews content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit any story content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit any webform content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit own page content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit own search content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit own simplenews content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Εικόνα 5: Λίστα με τα δικαιώματα για κάθε χρήστη**

Κατασκευάζουμε τους ρόλους του διαχειριστή και του webmaster και το επόμενο βήμα είναι να τους καταχωρίσουμε δικαιώματα σε αυτούς αλλά και τους άλλους ρόλους. Από το user management επιλέγουμε δικαιώματα (permissions) και από την λίστα που εμφανίζεται (βλ. εικόνα 5 ) επιλέγουμε τι δικαιώματα θα έχει ο κάθε χρήστης. Η λίστα αυτή είναι η λίστα με τις επιλογές που παρέχει κάθε μονάδα, έτσι ελέγχουμε ποιος χρήστης θα έχει τι δικαιώματα σε κάθε μονάδα και κατά επέκταση σε ολόκληρο το σύστημα.

Το επόμενο που πρέπει να κατασκευαστεί είναι ο βασικός τύπος περιεχομένου *movie*. Το *Ubercart* προσφέρει την δυνατότητα κατασκευής κλάσεων. Για την δημιουργία της κλάσης (βλ. εικόνα 6) που χρειάζεται το σύστημα επιλέγουμε *Store administrator-> products->Manage classes* και δημιουργούμε την κλάση *movie*. Η κλάση που δημιουργείται έχει ως πεδία τον τίτλο την περιγραφή και την τιμή του προϊόντος, όμως αυτά δεν αρκούν.

**Εικόνα 6: Δημιουργία κλάσης *movie***

Για την δημιουργία νέων πεδίων χρησιμοποιείτε η μονάδα [CCK](#) (Content Construction Kit) του *Drupal*. Η μονάδα *CCK* επιτρέπει την προσθήκη προσαρμοσμένων πεδίων για προσαρμοσμένους τύπους περιεχομένου με τη χρήση ενός *web interface*, συνήθως του *web browser*. Επιτρέπει την δημιουργία πεδίων για την χρήση κειμένου (*Text field*, *Text area*, *Select list*, *Radio buttons*, *Check boxes*) την προσάρτηση αρχείων (*File upload*, *image*) ημερομηνίας, ακεραίων, πραγματικών κ.α. Τα επιπρόσθετα πεδία (βλ. εικόνα 7) είναι το *Duration* (αριθμητικό), *Video* (κείμενο), *Audio* (κείμενο), *Subtitles* (κείμενο), *Image* (*image*), *Screenshots* (*image*), *Slide* (*image*) και *Movie Trairel* (*URL*). Για την προσθήκη νέων πεδίων δικαίωμα έχει μόνο ο διαχειριστής.

Label	Name	Type	Operations
+ Name	Node module form.		
+ Description	Node module form.		
+ Duration	field_duration	Text	Configure Remove
+ Video	field_video	Text	Configure Remove
+ Audio	field_audio	Text	Configure Remove
+ Subtitles	field_subtitles	Text	Configure Remove
+ Taxonomy	Taxonomy module form.		
+ Image	field_image_cache	File	Configure Remove
+ Screenshots	field_screenshots	File	Configure Remove
+ Slide	field_slide	File	Configure Remove
+ Movie Trairel	field_movie_trairel	Embedded Video	Configure Remove
+ Shipping	Shipping settings form.		
+ Product information	Product module form.		
+ Menu settings	Menu module form.		
+ Description	Node description. (View tab)		
+ Revision information	Node module form.		
+ Authoring information	Node module form.		
+ Publishing options	Node module form.		

**Εικόνα 7: Πεδία του βασικού τύπου περιεχομένου *movie***



Το επόμενο βήμα είναι η δημιουργία λεξικών για την κατηγοριοποίηση των προϊόντων. Κάθε λεξικό θα περιέχει τις λέξεις κλειδιά οι οποίες θα αντιστοιχούν σε κάθε τύπο περιεχομένου. Το Drupal προσφέρει την μονάδα [Taxonomy](#) για την δημιουργία αυτών το λεξικών. Η μονάδα ταξινόμησης επιτρέπει την κατηγοριοποίηση του περιεχομένου χρησιμοποιώντας λέξεις-κλειδιά αλλά και καθορισμένους όρους από τον διαχειριστή του συστήματος. Πρόκειται για ένα ευέλικτο εργαλείο για την ταξινόμηση του περιεχομένου με πολλά προηγμένα χαρακτηριστικά. Κάθε λεξικό από την στιγμή που συνδέεται με ένα τύπο περιεχομένου δημιουργούνται νέα πεδία (κάθε πεδίο αντιστοιχεί σε ένα λεξικό) στον τύπο αυτό όπου τα περιεχόμενα των πεδίων αυτών τοποθετούνται αυτόματα στο αντίστοιχο λεξικό ενώ αυτόματα μετατρέπονται και σε λέξεις κλειδιά όπου με την σειρά τους απλοποιούν την κατηγοριοποίηση του περιεχομένου. Κάθε λεξικό μπορεί να έχει περιορισμένες λέξεις κλειδιά ορισμένες από τον διαχειριστή ή ελεύθερη επισήμανση (Free tagging).

Name	Type	Operations
+ Disk type	New movie	edit vocabulary list terms add terms
+ Movie genre	New movie	edit vocabulary list terms add terms
+ Actors	New movie	edit vocabulary list terms add terms
+ Director	New movie	edit vocabulary list terms add terms
+ Production	New movie	edit vocabulary list terms add terms
+ Year	New movie	edit vocabulary list terms add terms
+ Catalog	Product	edit vocabulary list terms add terms
+ Newsletter	Newsletter issue	edit vocabulary list terms add terms

Save

**Εικόνα 8: Λεξικά που συνδέονται με τον τύπο περιεχομένου movie**

Τα λεξικά που δημιουργήθηκαν στην παρούσα πτυχιακή εργασία (βλ. εικόνα 8) είναι και των δύο ειδών. Για την επιλογή μεταξύ του εάν ένας δίσκος ανήκει σε DVD ή Blu-Ray δημιουργήθηκε το λεξικό Disk type και περιέχει μόνο τις ορισμένες από τον διαχειριστή λέξεις κλειδιά DVD και Blu-Ray και είναι υποχρεωτικό να επιλεγεί ένα από τα δύο. Αν είχαμε περισσότερα είδη δίσκων θα είχαμε περισσότερες λέξεις κλειδιά. Τα υπόλοιπα λεξικά είναι το Movie genre (Free tagging) το Actors (Free tagging) το Director (Free tagging) το Production (Free tagging) και το Year (Free tagging), τα υπόλοιπα δεν συνδέονται με το content type movie.

Τώρα που έχουν αποδοθεί όλα τα απαραίτητα πεδία στον βασικό τύπο περιεχομένου movie είναι δυνατή και η κατασκευή του πρώτου προϊόντος (βλ. εικόνα 9). Από το μενού διαχείρισης επιλέγουμε Content management->Create content->movie. Στη σελίδα που εμφανίζει το σύστημα το πρώτο πεδίο που είναι υποχρεωτικό να συμπληρωθεί είναι ο τίτλος του νέου προϊόντος. Το επόμενο μη υποχρεωτικό είναι η περιγραφή-description. Τα επόμενα πεδία είναι τα πεδία που προστέθηκαν με την χρήση της μονάδας CCK τα οποία και είναι μη υποχρεωτικά.

Στο πεδίο της περιγραφής έχει εγκατασταθεί η μονάδα WYSIWYG (What You See Is What You Get) η οποία επιτρέπει να χρησιμοποιηθούν client-side συντάκτες για επεξεργασία του περιεχομένου. Απλοποιεί την εγκατάσταση και την ολοκλήρωση ενός επεξεργαστεί κειμένου οποιαδήποτε επιλογής. Η μονάδα αυτή αντικαθιστά όλες τις άλλες μονάδες επεξεργασίας κειμένου. Η WYSIWYG μονάδα είναι σε θέση να υποστηρίξει οποιοδήποτε είδος client-side επεξεργαστή κειμένου. Μπορεί να είναι ένας HTML-editor (γνωστός και ως WYSIWYG), ένας pseudo-editor (κουμπιά για την εισαγωγή markup σε ένα text area), ή ακόμα και flash-based εφαρμογές. Η βιβλιοθήκη για τους συντάκτες πρέπει να εγκατασταθεί ξεχωριστά. Διάφοροι συντάκτες υποστηρίζονται είναι οι jEDIT Notepad WordPad Metapad κ.α.

**Create New movie**

**Name:** \*

**Description:**  Show summary in full view

**Disable rich-text**  
Enter the product description used for product teasers and pages.

**Duration:**  
Εισάγετε ΕΔΩ την διάρκεια της ταινίας.

**Video:**  
Εισάγετε ΕΔΩ τα χαρακτηριστικά του video.

**Audio:**  
Εισάγετε ΕΔΩ τα χαρακτηριστικά του ήχου.

**Subtitles:**  
Εισάγετε ΕΔΩ τους διαθέσιμους υπότιτλους.

**Εικόνα 9: Πεδία τίτλος, περιγραφή και επιπρόσθετα πεδία από το CCK**

Τα επόμενα πεδία του βασικού τύπου περιεχομένου monie είναι τα πεδία των λεξικών που συνδέονται με τον συγκεκριμένο τύπο περιγεομένου (βλ. εικόνα 10). Το πρώτο πεδίο είναι υποχρεωτικό και έχει μόνο δύο επιτρεπτές τιμές ορισμένες από τον διαχειριστή. Τα επόμενα πεδία είναι ελεύθερης σήμανσης και μπορούν να πάρουν οποιαδήποτε τιμή. Το μεγάλο πλεονέκτημα των πεδίων αυτών είναι ότι είναι auto complete, δηλαδή αν κάποια τιμή έχει εισαχθεί από προηγούμενη καταχώρηση συμπληρώνεται αυτόματα το πεδίο χωρίς να ξανά εισαχθεί η τιμή.

**Vocabularies**

**Disk type:** \*  
- Please choose -  
Επιλέξτε ένα είδος δίσκου.

**Movie genre:**  
Εισάγετε ένα είδος ταινίας ή και περισσότερα χωρίζοντας τα με ένα κόμμα.

**Actors:**  
Εισάγετε τα ονόματα των ηθοποιών χωρίζοντας τα με ένα κόμμα.

**Director:**  
Εισάγετε το όνομα-τα του σκηνοθέτη-τών χωρίζοντας τα με ένα κόμμα.

**Production:**  
Εισάγετε την εταιρία παραγωγής της ταινίας.

**Year:**  
Εισάγετε το έτος παραγωγής της ταινίας.

**Εικόνα 10: Πεδία λεξικών που συνδέονται με τον τύπο monie**

Τα επόμενα πεδία για την δημιουργία του τύπου περιεχομένου *movie* έχουν και αυτά δημιουργηθεί με την μονάδα CCK. Είναι τα πεδία *image*, *screenshots*, *slide* και *movie trailer* (βλ. εικόνα 11).. Το πρώτο πεδίο επιτρέπει την εισαγωγή φωτογραφίας (συγκεκριμένου μεγέθους και για συγκεκριμένους τύπους αρχείων) για το εξώφυλλο του προϊόντος αλλά δεν είναι υποχρεωτικό, αν δεν εισαχθεί κάποια φωτογραφία δηλαδή θα χρησιμοποιηθεί η εξ ορισμού. Το επόμενο πεδίο επιτρέπει την εισαγωγή επιπλέον φωτογραφιών (συγκεκριμένου μεγέθους και για συγκεκριμένους τύπους αρχείων) που θα χρησιμοποιηθούν για τον εμπλουτισμό της περιγραφής του προϊόντος και οι οποίες θα εμφανίζονται μόνο όταν ανοιχτεί πλήρως ένας κόμβος, όχι δηλαδή κατά την προεπισκόπηση. Ούτε αυτό το πεδίο είναι υποχρεωτικό αλλά δεν υπάρχουν εξ ορισμού φωτογραφίες για αυτό το πεδίο.

**Image:**

Add another item

**Screenshots:**

Εισάγεται τα Screenshots της τανείας ΕΔΩ.

Add another item

**Slide:**

Ανεβάστε μια εικόνα για να προβάλεται στο slideshow του frontpage.

**Movie Trairel:**

**Εικόνα 11: Επιπλέον πεδία του τύπου *movie***

Το επόμενο πεδίο επιτρέπει πάλι την εισαγωγή φωτογραφιών (συγκεκριμένου μεγέθους και για συγκεκριμένους τύπους αρχείων) αλλά αυτή την φορά οι φωτογραφίες αυτές θα προβάλλονται στην αρχική σελίδα σε ένα *slide show* με τα πιο πρόσφατα προϊόντα. Είναι ένα πεδίο δηλαδή για την προώθηση νέων προϊόντων άμεσα στην πρώτη σελίδα επόμενος είναι στην κρίση του χρήστη πότε θα το χρησιμοποιήσει. Ούτε αυτό το πεδίο είναι υποχρεωτικό αλλά δεν υπάρχουν εξ ορισμού φωτογραφίες για αυτό το πεδίο. Το *slide show* κατασκευάστηκε μέσω της μονάδας *views* την οποία θα μελετήσουμε αργότερα στο κεφάλαιο αυτό. Τέλος το πεδίο *movie trailer* επιτρέπει την εισαγωγή διευθύνσεων URL από πού και “τραβάει” τα αντίστοιχα *trailer* για τα αντίστοιχα προϊόντα. Στην παρούσα εργασία έχει επιλεγθεί ως “πηγή” το YouTube. Ούτε αυτό το πεδίο είναι υποχρεωτικό αλλά δεν υπάρχουν εξ ορισμού φωτογραφίες για αυτό το πεδίο. Τα τελευταία πεδία που απομένουν είναι αυτά του *Ubercart* (βλ. εικόνα 12). Το *Ubercart* παρέχει τα πεδία *SKU* για τον κωδικό του προϊόντος (αριθμός- υποχρεωτικό) *List price* για την τιμή που θα έχουν τα προϊόντα στη λίστα με τα προϊόντα (αριθμός- μη υποχρεωτικό) το πεδίο *Cost* για το κόστος του προϊόντος (αριθμός- μη υποχρεωτικό) και το πεδίο *Sell price* για την τιμή πώλησης του προϊόντος (αριθμός- υποχρεωτικό). Επίσης το τελευταίο πεδίο επιτρέπει τον εξ ορισμού καθορισμό του αριθμού των προϊόντων που θα προσθέτονται στο καλάθι αγορών.

Μετά από την ορθή συμπλήρωση όλων των απαραίτητων πεδίων μπορούμε να σώσουμε το την φόρμα δημιουργίας του τύπου *movie* και ανάλογα τις λέξεις κλειδιά που δόθηκαν το περιεχόμενο κατατάσσεται αυτόματα στην κατηγορία που ανήκει. Έτσι δημιουργείται και ένας νέος κόμβος. Εξ ορισμού κάθε νέος κόμβος δημοσιεύεται και στην πρώτη σελίδα αλλά αυτό δεν είναι υποχρεωτικό.

**Product information**

**SKU:** \*

Product SKU/model.

**List price:** € 0 The listed MSRP.

**Cost:** € 0 Your store's cost.

**Sell price:** \* € 0 Customer purchase price.

**Default quantity to add to cart:** 1

Leave blank or zero to disable the quantity field next to the add to cart button, if it is enabled in **general**. If it is disabled, this field is ignored.

**Εικόνα 12: Παρεχόμενα πεδία από την κλάση του Ubercart**

Έτσι δημιουργείται ένας κόμβος. Ας δούμε τώρα όμως πως αυτός και κάθε κόμβος κατηγοριοποιείται αυτόματα με βάση τις λέξεις κλειδιά άλλα και πως μορφοποιείται το περιεχόμενο αυτού και της κάθε σελίδας που ανήκει. Για ακόμη μια φορά το Drupal δίνει την λύση και η λύση αυτή δεν είναι άλλη από την μονάδα Views. Η μονάδα views είναι ένα ισχυρός δημιουργός ερωτημάτων προς τη βάση δεδομένων για το Drupal. Αυτό το εργαλείο είναι ουσιαστικά ένας δημιουργός ερωτημάτων δεδομένου ότι μπορεί να χτίσει το κατάλληλο ερώτημα να το εκτελέσει, και να το εμφανίσει με τα κατάλληλα αποτελέσματα. Είναι μια μονάδα που επιτρέπει τη μεταφορά περιεχομένου από τη βάση δεδομένων και το παρουσιάζει στο χρήστη με τρόπους προσαρμοσμένους στις ανάγκες του: λίστες, μηνύματα, γκαλερί, εκθέσεις, δημοσιεύσεις. Μερικές από τις ανάγκες που εξυπηρετεί η μονάδα αυτή είναι η δημιουργία της πρώτης σελίδας και η ανάλογη ταξινόμηση της η ταξινόμηση των λέξεων κλειδιών κατά οποιοδήποτε τρόπο κ.α.

**The views 2 user interface**

Items with a gear icon have additional settings.

Add items to the list with the plus buttons.

Rearrange the order of the items in the list.

Changes to this view have not been saved yet.

All displays for this view.

This indicates which display is being edited.

The active item is highlighted.

Changed items are highlighted.

Lighter color and italics indicate that this item is using the default display's values.

The question mark icons provide additional help.

The update button stores changes to the current item. This either opens the 'next' item if there is one, or removes the form and updates the live preview.

This button will cancel all changes to the view. Save is disabled while a settings form is open to prevent clicking the wrong update button.

Click an item to display its settings in the lower area.

Click this button to change whether this item uses defaults or its own, overridden values.

Discard changes to just this item.

Block 1: Items per page

Status: using overridden values.

5

The number of items to display per page. Enter 0 for no limit.

Offset:

0

The number of items to skip. For example, if this field is 3, the first 3 items will be skipped and not displayed. Offset can not be used if items to display is 0; instead use a very large number there.

Update Cancel

Save Cancel Delete

**Εικόνα 13: Το User Interface των Views**

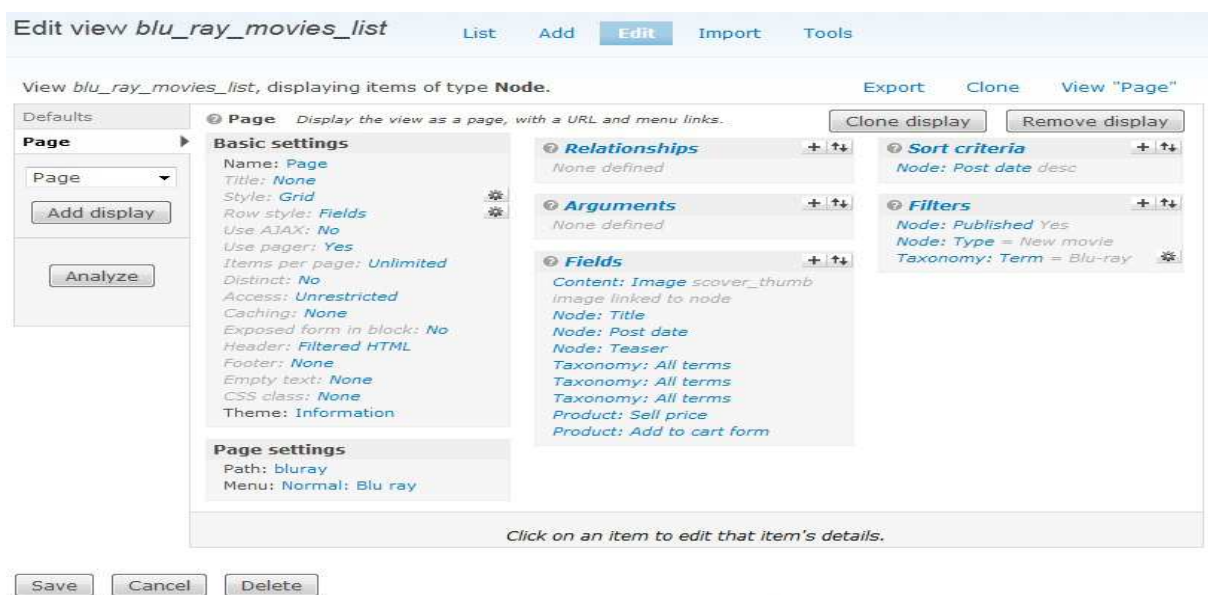
### 4.3.1 Υλοποίηση με Views

Η μονάδα [Views](#) μας επιτρέπει με λίγα λόγια την δημιουργία ταξινομημένου περιεχομένου, οτιδήποτε και αν είναι αυτό είτε φωτογραφίες είτε πίνακες δεδομένων είτε κείμενο είτε ημερομηνίες αλλά και τον συνδυασμό σύνθετων ερωτήσεων προς την βάση δεδομένων. Στην παρούσα εργασία κατασκευάστηκαν συνολικά έντεκα Views (βλ. εικόνα 14).

ⓘ Normal Node view: <b>blu_ray_movies_list</b> (custom) Path: <a href="#">blu-ray</a> Page	The Blu-ray movies list.	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>dvd_movies_list</b> (custom) Path: <a href="#">dvd</a> Page	The DVD movies list.	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>Latest_movies</b> (custom) Path: <a href="#">latest-movies</a> Page	A list of the latest movies	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>movie_actors_list</b> (custom) Title: Actors list Path: <a href="#">actors</a> Block, Page	A list of movies actors	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>movie_directors_list</b> (custom) Title: Directors list Path: <a href="#">director</a> Block, Page	A list of movies directos	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>movie_year_list</b> (custom) Title: Years list Path: <a href="#">year</a> Block, Page	A list of movies year	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>Movies_categorys</b> (custom) Title: Movies Categorys Path: <a href="#">categorys</a> Block, Page	A list of movies categorys	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>search_database</b> (custom)	The database search view	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>studios_list</b> (custom) Title: Studios list Path: <a href="#">studios</a> Block, Page	A list of movies studios	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>The_front_page</b> (custom) Path: <a href="#">home</a> Page	The front page	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
ⓘ Normal Node view: <b>Top_rated_movies1</b> (custom) Title: Top rated movies Path: <a href="#">top-rated</a> Block, Page	A list of the top rated movies	<a href="#">Edit</a>   <a href="#">Export</a>   <a href="#">Clone</a>   <a href="#">Delete</a>

Εικόνα 14: Οι custom Views της εργασίας

Στο Drupal τα πάντα συνδέονται και αλληλεπιδρούν μεταξύ τους, έτσι συμβαίνει και με τις views. Σε κάθε view μπορούμε να εισάγουμε όλα εκείνα τα στοιχεία που δημιουργήθηκαν από τα προηγούμενα βήματα, από τα πεδία και τις τιμές τους μέχρι και του ίδιους τους κόμβους και όχι μόνο αυτό αλλά μπορούμε και να τα ταξινομήσουμε να τα ομαδοποιήσουμε να τα φιλτράρουμε και να τα μορφοποιήσουμε όλα αυτά ανάλογα με το πως θέλουμε να εμφανίζονται. Έτσι γίνεται και σε αυτή την εργασία. Κάθε μια view εξυπηρετεί ένα σκοπό. Η πρώτη view είναι η `blu_ray_movies_list` (βλ. εικόνα 15) αυτό είναι και το όνομα της view. Σκοπός αυτής της view είναι να περιλαμβάνει μόνο όσα προϊόντα ανήκουν στην κατηγορία Blu-ray ή αλλιώς που έχουν Taxonomy term (όρος ταξινόμησης) ίσο με Blu-ray. Αυτό επιτυγχάνεται επιλέγοντας από τα φίλτρα (Filters) την λέξη κλειδί Blu-ray (όρος του λεξικού Disk type), δηλαδή για να εμφανιστεί κάποιος κόμβος σε αυτή την view (φιλτράρονται όλοι οι κόμβοι) θα πρέπει να περιέχει στο πεδίο Disk type, κατά την δημιουργία του, την λέξη κλειδί Blu-ray. Με άλλα λόγια στην view αυτή θα εμφανίζονται μόνο οι κόμβοι με Disk type: Blu-ray. Έτσι δημιουργήθηκε μια ομάδα κόμβων.



Εικόνα 15: Η view Blu-ray

Για να εμφανιστεί επίσης κάποιος κόμβος στην συγκεκριμένη view θα πρέπει να είναι δημοσιευμένος και ο τύπος του να είναι New movie διαφορετικά δεν θα εμφανιστεί. Το επόμενο που έχει οριστεί στην view αυτή είναι η σειρά με την οποία θα εμφανίζονται οι κόμβοι (αυτό ορίζεται από το Sort criteria) και η σειρά αυτή είναι κατά φθίνουσα ημερομηνία δημοσίευσης, δηλαδή ότι δημιουργείται τελευταίο θα εμφανίζεται πρώτο. Από το πεδίο Fields επιλέγονται ποια πεδία του τύπου περιεχομένου New movie θα εμφανίζονται στην view αυτή. Κάθε πεδίο είναι πλήρως παραμετροποιήσιμο ανάλογα του τύπου στον οποίο ανήκει το κάθε στοιχείο, παράδειγμα αν είναι ημερομηνία μπορεί να επιλεγεί το format της ημερομηνίας που θα εμφανίζεται, επίσης κάθε πεδίο μπορεί να εξαιρεθεί από την view και να εμφανίζεται χειροκίνητα, μέσω κώδικα δηλαδή, από ένα άλλο πεδίο. Το πρώτο πεδίο είναι το Image τα επόμενα είναι ο τίτλος, η ημερομηνία δημοσίευσης, η περιγραφή του κόμβου, οι αντίστοιχες λέξεις κλειδιά, η τιμή πώλησης και τέλος το κουμπί add to cart. Η σειρά με την οποία θα εμφανίζονται είναι αντίστοιχη της σειράς εισαγωγής των πεδίων και φυσικά υπάρχει η δυνατότητα αλλαγής της σειράς αυτής. Εδώ αυτή η σειρά είναι η καταλληλότερη. Έτσι κάθε κόμβος που ανήκει στον τύπο περιεχομένου New movie και έχει συμπληρωμένα τα κατάλληλα πεδία (Disk type = Blu-ray ) θα εμφανίζεται στην κατηγορία Blu-ray όλα τα υπόλοιπα πεδία. Στο πεδίο Page settings μια πολύ σημαντική επιλογή είναι το path.

Εκεί ορίζεται το μονοπάτι που θα οδηγεί στην view αυτή. Για να προστεθεί το μονοπάτι αυτό στο βασικό μενού θα πρέπει από το μενού διαχείρισης να επιλέγει Site building->Menus->Primary Links και εκεί να δημιουργηθεί ένα στοιχείο μενού (βλ. εικόνα 16) με το όνομα και το path που ορίστηκε σε αυτή την view. Αυτό το βήμα είναι κοινό για κάθε view.

Το επόμενο και τελικό βήμα είναι η μορφοποίηση των πεδίων στην view, πώς δηλαδή θα εμφανίζονται τα πεδία που έχουν προστεθεί στη view, με ποια σειρά με τι χρώμα πώς θα στοιχίζονται κ.α. Για την επιτυχία του επιθυμητού αποτελέσματος ο δρόμος της μορφοποίησης περνάει μέσα από το CSS. Το Drupal βοηθάει πολύ σε αυτό το μέρος αλλά απαιτείται και η γνώση CSS από το χρήστη. Για την μορφοποίηση των πεδίων της view ακολουθήθηκε μια συγκεκριμένη διαδικασία. Πρώτα εκμεταλλεύτηκε το γεγονός εξαίρεσης όλων των πεδίων, εκτός από ένα-διαφορετικά δεν θα εμφανίζονταν τίποτα, από την view και αυτό γιατί δίνει την δυνατότητα στο χρήστη να διαχειριστεί τα πεδία μεμονωμένα ορίζοντας δικές του κλάσεις CSS και δικό του HTML κώδικα. Με αυτό τον τρόπο παρακάμπτονται οι εξ ορισμού κλάσεις του Drupal ορίζοντας δικές του κλάσεις ο χρήστης ένα επίπεδο πιο κάτω από τις αρχικές κλάσεις. Στο Drupal κάθε πεδίο έχει ένα πρότυπο, ένα pattern, το οποίο μπορεί να χρησιμοποιήσει ο χρήστης έτσι ώστε να διαχειριστεί τα πεδία ανάλογα με τις ανάγκες του. Τώρα στο τελευταίο πεδίο της view το Add to cart έχει προστεθεί ο κώδικας της εικόνας 17.

**Εικόνα 16: Δημιουργία στοιχείου μενού**

Έχει επιλεγεί να ξαναγραφτεί η έξοδος του πεδίου και το πώς θα ξαναγραφτεί ορίζεται μέσα στο πεδίο text. Ορίζονται αρχικά οι κλάσεις myimg για το πεδίο που αντιστοιχεί στο εξώφυλλο του προϊόντος, mytitle για το πεδίο που αντιστοιχεί στον τίτλο, myprc για το πεδίο που αντιστοιχεί στην τιμή και mybtn για το πεδίο που αντιστοιχεί στο κουμπί add to cart. Οι κλάση mystyle περιλαμβάνει και ομαδοποιεί σε μια κλάση τα πεδία του τίτλου, της ημερομηνίας δημοσίευσης και των λέξεων κλειδιών. Όπως φαίνεται κάθε λέξη κλειδί έχει ένα αναγνωστικό, ένα tid, το οποίο είναι εξαιρετικά ωφέλιμο όπως θα αποδειχθεί αργότερα. Κάθε πεδίο έχει το δικό του pattern, εάν υπήρχαν περισσότερα πεδία θα υπήρχαν και περισσότερα patterns. Ο κώδικας CSS που απαιτείται για να λειτουργήσουν οι κλάσεις που ορίστηκαν θα πρέπει να γραφεί στο αντίστοιχο CSS αρχείο του θέματος του ιστότοπου (βλ. εικόνα 18). Για να καταλάβει το Drupal σε ποια view αναφέρεται ο αντίστοιχος κώδικας CSS θα πρέπει να οριστεί πρώτα μια κλάση με το όνομα της view και αμέσως μετά η κλάση που έχει οριστεί μέσα σε εκείνη την view. Το τελικό αποτέλεσμα φαίνεται στην εικόνα 19.

Rewrite the output of this field ▼

If checked, you can alter the output of this field by specifying a string of text with replacement tokens that can use any existing field output.

**Text:**

```
<div class="myimg">[field_image_cache_fid]</div>
<div class="mystyle">
<div class="mytitle">[title]</div>
[created]<br>
[tid_2] | [tid]<br>
[tid_1]<br>
</div>
<div class="myprc">[sell_price]</div>
<div class="mybtn">[addtocartlink]</div>
```

The text to display for this field. You may include HTML. You may enter data from this view as per the "Replacement patterns" below.

Output this field as a link ▶

If checked, this field will be made into a link. The destination must be given below.

Replacement patterns

The following tokens are available for this field. Note that due to rendering order, you cannot use fields that come after this field; if you need a field not listed here, rearrange your fields. If you would like to have the characters %5B and %5D please use the html entity codes '%5B' or '%5D' or they will get replaced with empty space.

Fields

- [field\_image\_cache\_fid] == Content: Image (field\_image\_cache)
- [title] == Node: Title
- [created] == Node: Post date
- [teaser] == Node: Teaser
- [tid] == Taxonomy: All terms
- [tid\_2] == Taxonomy: All terms
- [tid\_1] == Taxonomy: All terms
- [sell\_price] == Product: Sell price
- [addtocartlink] == Product: Add to cart form

Εικόνα 17: Ορισμός κλάσεων για τα πεδία της view Blu-ray

```
1361   div.view-blu-ray-movies-list div.myview
1362   {
1363     position: relative;
1364     height: 138px;
1365     left: 0px;
1366   }
1367   div.view-blu-ray-movies-list div.mytitle
1368   {
1369     font-size: 15px;
1370     width: 314px;
1371   }
1372   div.view-blu-ray-movies-list div.mystyle
1373   {
1374     position: relative;
1375     left: 86px;
1376     bottom: 92px;
1377     height: 80px;
1378     width: 314px;
1379     font-size: 11px;
1380   }
1381   div.view-blu-ray-movies-list div.myimg
1382   {
1383     position: relative;
1384     left: 13px;
1385     bottom: -7px;
1386   }
1387   div.view-blu-ray-movies-list div.myprc
1388   {
1389     position: relative;
1390     left: 26px;
1391     bottom: 78px;
1392   }
1393   div.view-blu-ray-movies-list div.mybtn
1394   {
1395     position: relative;
1396     left: 6px;
1397     bottom: 79px;
1398   }
```


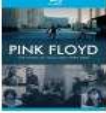





Εικόνα 18: Κώδικας CSS για τα πεδία της view Blu-ray



Τα ίδια βήματα έχουν ακολουθηθεί για την κατασκευή και μορφοποίηση της view `dvd_movies_list`, με την διαφορά ότι στο Taxonomy term έχει οριστεί ο τύπος DVD της view `Latest_movies`, με την διαφορά ότι στα φίλτρα έχει οριστεί η ημερομηνία δημοσίευσης να είναι μικρότερη των δεκαπέντε ημερών (μια από τις πολλές δυνατότητες που παρέχει το Drupal), δηλαδή σε αυτή την view θα εμφανίζονται προϊόντα που έχουν δημοσιευθεί μέχρι και δεκαπέντε μέρες της view `movie_actors_list`, της view `movie_directors_list` και τέλος της view `movie_year_list`. Αντίστοιχα για κάθε view έχει γραφτεί ο ανάλογος CSS και HTML κώδικας. Για τις view `Movies_categories`, `studios_list` και `The_front_page` έχουν γίνει κάποιες παραπάνω ενέργειες.

Οι views εκτός από την κατασκευή ταξινομημένων, μορφοποιημένων και γενικά παραμετροποιήσιμων σελίδων δίνουν και την δυνατότητα κατασκευής μπλόκς (Block). Ένα block είναι ένα τμήμα μιας σελίδας οπουδήποτε μέσα σε αυτή το οποίο μπορεί να περιέχει διάφορες πληροφορίες, από φωτογραφίες, κείμενο, στατιστικά μέχρι μενού και feeds. Στην παρούσα εργασία, όπως έχει ήδη αναφερθεί, είναι σημαντική η άμεση κατηγοριοποίηση των προϊόντων. Χρειάζεται λοιπόν ένα σημείο μέσα στον ιστότοπο που να μπορεί άμεσα ο χρήστης να κατηγοριοποιήσει το περιεχόμενο, αυτό επιτυγχάνεται πλήρως μέσω των views και των blocks. Οι κατηγορίες θα είναι δύο, μια με το είδος της ταινίας και η άλλη με το στούντιο παραγωγής. Για την πρώτη κατηγορία κατασκευαστική η view `Movies_categories` και για την άλλη η view `studios_list`.

### Blu-ray Releases

 <p><b>Bob Marley</b> October 23 2012 2012   Magnolia Pictures Documentary   Music</p> <p>39.99 € Add to cart</p>	 <p><b>The Story of Wish You Were Here</b> October 23 2012 2011   Eagle Rock Entertainment Music</p> <p>36.67 € Add to cart</p>
 <p><b>The Doors: Live At The Bowl '68</b> October 23 2012 2012   Eagle Rock Entertainment Music</p> <p>28.98 € Add to cart</p>	 <p><b>Sherlock Holmes: A Game of Shadows</b> March 30 2012 2011   Warner Bros. Action   Adventure   Mystery</p> <p>35.98 € Add to cart</p>
 <p><b>The Girl with the Dragon Tattoo</b> March 13 2012 2011   Columbia Pictures Crime   Drama   Mystery</p> <p>35.90 € Add to cart</p>	 <p><b>The Muppets</b> March 13 2012 2011   Walt Disney Pictures Comedy   Family   Musical</p> <p>41.90 € Add to cart</p>
 <p><b>Hugo</b> March 10 2012 2011   Paramount Pictures Adventure   Drama   Family</p> <p>37.99 € Add to cart</p>	 <p><b>The Adventures of Tintin</b> March 10 2012 2011   Columbia Pictures Adventure   Animation   Family</p> <p>35.87 € Add to cart</p>

Εικόνα 19: Τελικό αποτέλεσμα της view Blu-ray

Η view `Movies_categories` (βλ. Εικόνα 20) έχει ένα display παραπάνω από τις προηγούμενες views, το display Block. Τα displays “λένε” σε μια view που πρέπει να πάει η έξοδος. Με την προσθήκη ενός display σε μια view, είναι δυνατό να έχετε την εμφάνιση της view σας ως μια σελίδα, ή ως ένα μπλοκ, ή ακόμα και ως συνημμένο σε διαφορετικό display μέσα στην view. Κάθε display μπορεί να έχει τις δικές του ρυθμίσεις, αλλά όταν δημιουργείται, ένα display θα λάβει όλο το σύνολο των βασικών ρυθμίσεων του από το προεπιλεγμένο display που όλες οι views πρέπει να έχουν. Η επιλογή add display επιτρέπει στο χρήστη να εισάγει πολλαπλά επίπεδα χρηστικότητας, τα οποία και επικοινωνούν μεταξύ τους, σε μια view (όπως attachments feeds blocks κ.α.) και να ορίσει αυτός πια θα εμφανίζονται και πια όχι (βλ. και <http://drupal.org/documentation/modules/views>). Επίσης στα φίλτρα έχει επιλέγει το λεξικό `Movie genre`. Αυτό ορίζει ότι στην view αυτή θα εμφανίζονται μόνο όροι από το λεξικό `Movie genre`. Στο display Page έχουν οριστεί και τα επιθυμητά πεδία που θα εμφανίζονται στην σελίδα της view.

Ένα επιπλέον χαρακτηριστικό της view αυτής είναι το πεδίο Argument. Στο πεδίο αυτό εισάγονται ορίσματα τα οποία κατά κύριο λόγο προέρχονται από την URL. Κάθε τύπος display μπορεί να έχει τη δική του πηγή για τα επιχειρήματα. Τα block displays δεν έχουν καμία πηγή ορισμάτων. Δεν μπορούν να τραβήξουν τα ορίσματα από το URL, και συχνά απαιτούν τη χρήση προεπιλεγμένων ορισμάτων από τον PHP κώδικα, προκειμένου να πάρει τα ορίσματα. Σε γενικές γραμμές, τα ορίσματα χρησιμοποιούνται για να φιλτράρετε μια view, και με αυτή την έννοια έχουν μια πολύ στενή σχέση με τα φίλτρα, αλλά αυτό δεν είναι απαραίτητα αλήθεια για κάθε όρισμα. Τα ορίσματα μπορεί να χρησιμοποιηθούν για οποιοδήποτε σκοπό, πραγματικά. Η έκταση του τι θα κάνει ένα όρισμα εξαρτάτε από τον δημιουργό του ορισματος, αλλά τα ορίσματα που έρχονται με τις views είναι σχεδόν εξ ολοκλήρου φίλτρα. Μια τυπική χρήση των ορισμάτων θα μπορούσε να είναι η μείωση μιας view σε ένα μοναδικό κόμβο, ένα μεμονωμένο χρήστη, ή κόμβους με ένα συγκεκριμένο όρο ταξινόμησης. Αυτόν ακριβώς τον σκοπό εξυπηρετεί και το όρισμα στην view `Movies_categories`, το φιλτράρισμα των κόμβων στη view με βάση το είδος ταινίας (Movie genre) που ανήκουν

The screenshot displays the 'Edit view Movies\_categories' interface. At the top, there are tabs for 'List', 'Add', 'Edit' (selected), 'Import', and 'Tools'. Below the tabs, it indicates 'View Movies\_categories, displaying items of type Node.' and provides options for 'Export', 'Clone', and 'View "Page"'. The main configuration area is divided into several sections:

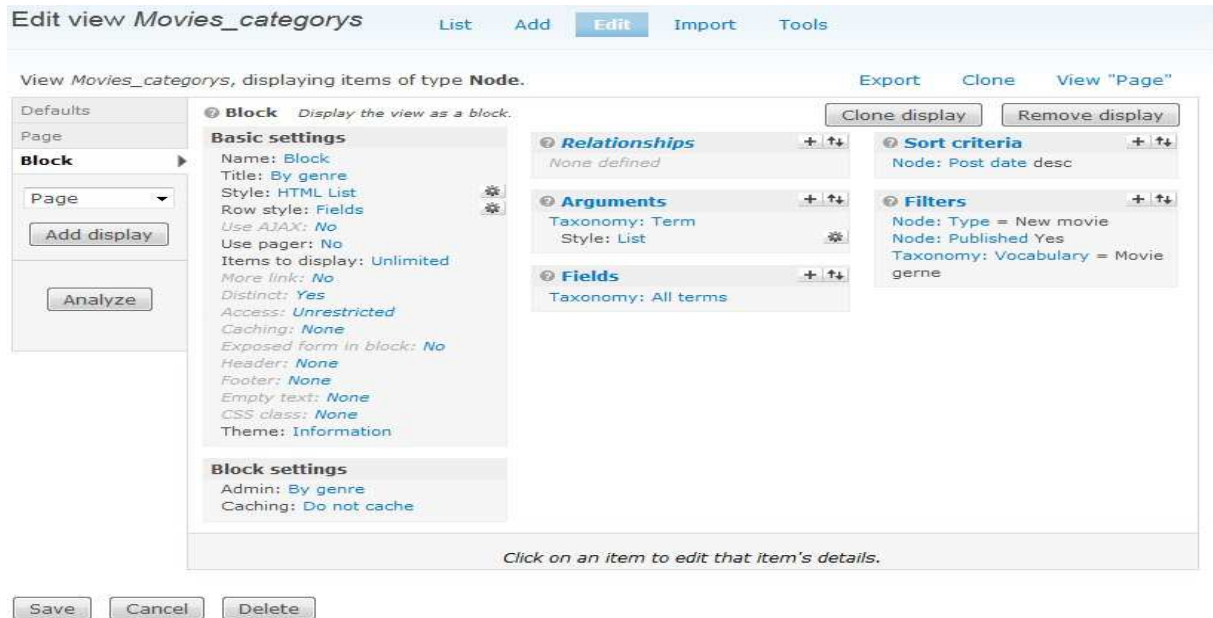
- Defaults:** Shows the current display type as 'Page'.
- Basic settings:** Includes fields for Name (Page), Title (None), Style (Unformatted), Row style (Fields), Use AJAX (Yes), Use pager (Yes), Items per page (15), Distinct (Yes), Access (Unrestricted), Caching (None), Exposed form in block (No), Header (None), Footer (None), Empty text (None), CSS class (None), and Theme (Information).
- Page settings:** Shows Path (categories) and Menu (No menu).
- Relationships:** Currently set to 'None defined'.
- Arguments:** Set to 'Taxonomy: Term'.
- Fields:** Includes 'Content: Image scover\_thumb' (image linked to node), 'Node: Title', 'Taxonomy: All terms', 'Content: Duration Default', 'Node: Post date', 'Node: Teaser', 'Taxonomy: All terms', 'Product: Sell price', and 'Product: Add to cart form'.
- Sort criteria:** Set to 'Node: Post date desc'.
- Filters:** Includes 'Node: Type = New movie', 'Node: Published Yes', and 'Taxonomy: Vocabulary = Movie genre'.

At the bottom, there are 'Save', 'Cancel', and 'Delete' buttons. A note at the bottom of the configuration area says 'Click on an item to edit that item's details.'

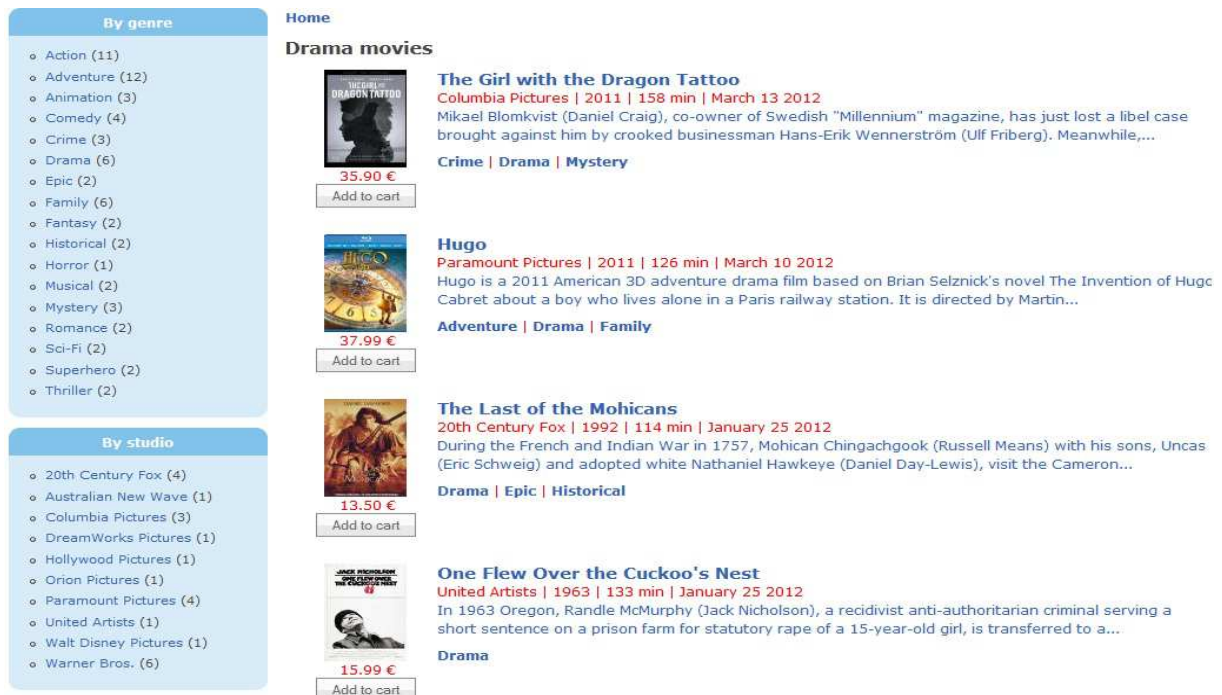
Εικόνα 20: Η view `Movies_categories`

Για να είναι άμεσα εφικτή η ταξινόμηση των προϊόντων με βάση το είδος τους θα χρησιμοποιηθεί ένα block. Το block αυτό θα εμφανίζεται πάντα στα δεξιά οποιαδήποτε σελίδας. Έτσι από τα displays έχει εισαχθεί ένα block με τις εξής ρυθμίσεις (βλ. εικόνα 21). Στο πεδίο Fields έχει επιλεγεί να εμφανίζεται μόνο το Taxonomy term της κατηγορίας που ανήκει κάθε προϊόν, δηλαδή στο block θα εμφανίζονται μόνο οι πιθανές κατηγορίες που μπορεί να ανήκει ένα προϊόν. Μια ταινία μπορεί να ανήκει σε μια ή και περισσότερες κατηγορίες. Στο πεδίο Argument έχει επιλεγεί πάλι το Taxonomy term που καθορίζει την κατηγορία που ανήκει ένα προϊόν, στην περίπτωση αυτή το Movie genre. Για να εμφανιστούν τα προϊόντα κάποιας κατηγορίας στην view θα πρέπει οι κατηγορία να υπάρχει αλλιώς δεν θα εμφανιστεί τίποτα. Αυτή είναι και η δουλειά του ορισματος που έχει οριστεί. Η διαδρομή για της κατηγορίες είναι η εξής: `http://localhost/videoclub/?q=categories` αλλά εδώ απουσιάζει κάποιο όρισμα και έτσι η view δεν θα εμφανίσει τίποτα. Για να εμφανίσει κάτι θα πρέπει να είναι παρόν κάποιο όρισμα από αυτά που έχουν οριστεί στο πεδίο Arguments και τα οποία με τη σειρά τους έχουν οριστεί, στην περίπτωση αυτή, σε κάποιο λεξικό και συγκεκριμένα στο `Movies genre`, έτσι για να εμφανιστούν οι ταινίες που ανήκουν στην κατηγορία, για παράδειγμα `Drama`, θα πρέπει να περάσει ως όρισμα ένας έγκυρος όρος κάποιας κατηγορίας που έχει οριστεί δηλαδή `http://localhost/videoclub/?q=categories/Drama`. Όλοι οι όροι και το τελικό αποτέλεσμα του της view και του block φαίνονται στην εικόνα 22. Δίπλα από κάθε όρο υπάρχει ο αριθμός των προϊόντων που ανήκουν στον όρο αυτό.

Όταν επιλεγεί κάποιος όρος από το block τότε η σελίδα που ανοίγεται με όλα τα προϊόντα που ανήκουν στην κατηγορία αυτή είναι η σελίδα Page που ανήκει στην ίδια view που ανήκει και το block. Η Page είναι υπεύθυνη για την εμφάνιση των προϊόντων στην Page και το block για την εμφάνιση των όρων στο block. Αυτός είναι και ο τρόπος που συνδέονται τα displays μεταξύ τους. Με τον ίδιο ακριβώς τρόπο κατασκευάστηκε και η view studios\_list. Επίσης η μορφοποίηση των views γραφικέ με τον ίδιο ακριβώς τρόπο με τις προηγούμενες views. Για κάθε view εννοείτε ότι έχει γραφεί ξεχωριστός κώδικας CSS.

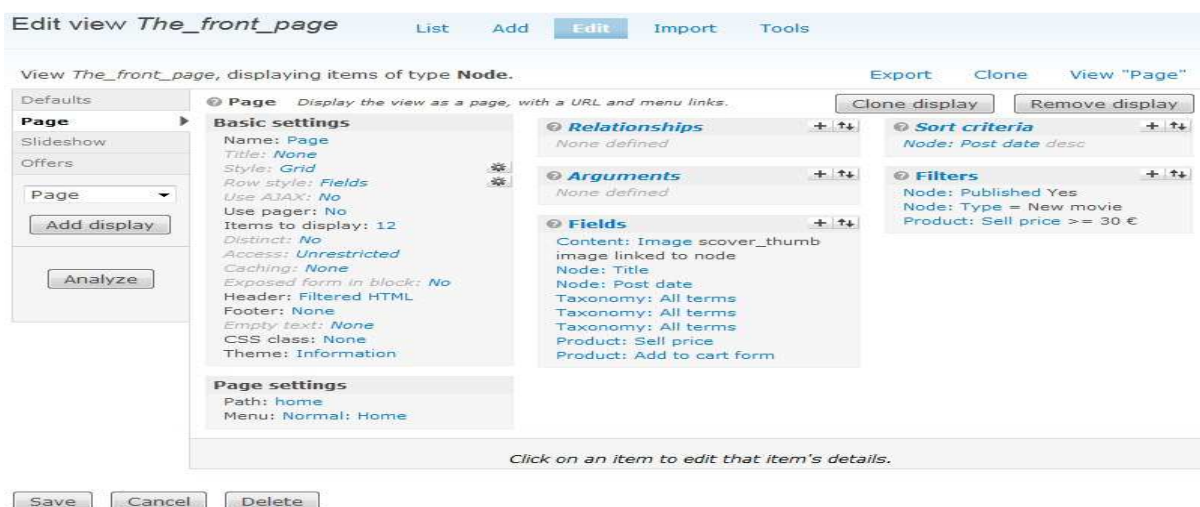


Εικόνα 21: Το block της view Movies\_categorys



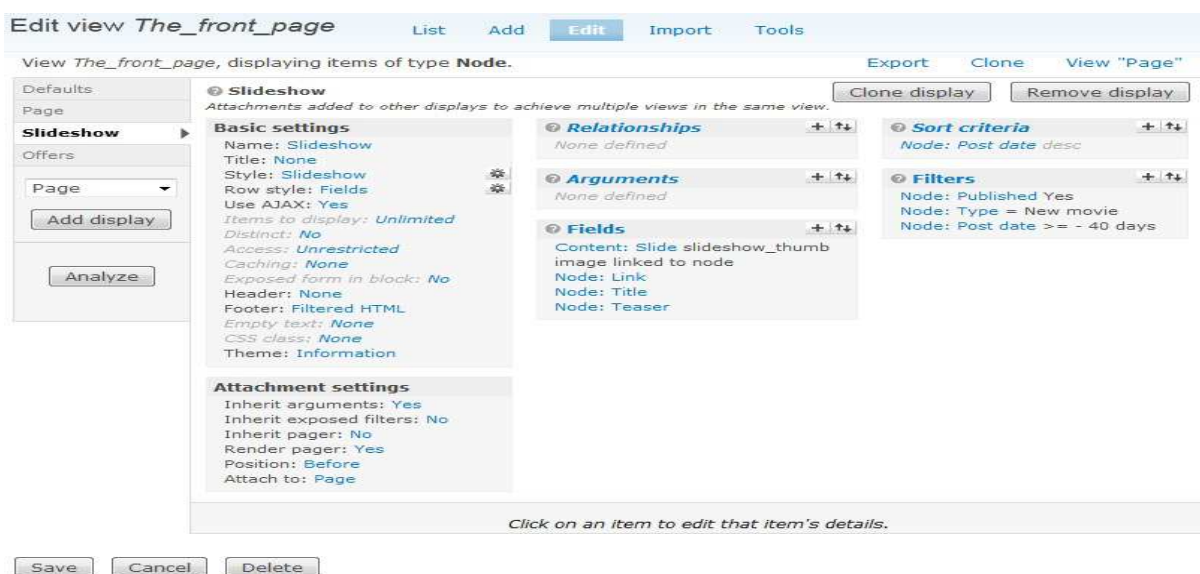
Εικόνα 22: Τελικό αποτέλεσμα της view Movies\_categorys

Η επόμενη view που κατασκευάστηκε, ίσως και η πιο σημαντική, είναι η view `The_front_page` (βλ. εικόνα 23). Σε αυτή τη view έχουν δημιουργηθεί τρία διαφορετικά displays. Το πρώτο είναι το Page. Στο Page έχουν οριστεί όλα τα πεδία που πρέπει να εμφανίζονται στην πρώτη σελίδα, όπως ο τίτλος η τιμή πώλησης οι λέξεις κλειδιά κ.α. Το Drupal επιτρέπει το ορισμό οποιασδήποτε σελίδας ως αρχικής. Αυτό επιτυγχάνεται ως εξής: Site configuration->Site information στο πεδίο Default front page ορίζεται το μονοπάτι της σελίδας που θα εμφανίζεται ως πρώτη σελίδα. Στην εργασία αυτή το μονοπάτι που έχουμε δώσει είναι το μονοπάτι που έχει οριστεί στην view `The_front_page` στο πεδίο Page settings->path:home. Στο πεδίο Filters έχει οριστεί η τιμή πώλησης να είναι πάνω από τριάντα ευρώ, δηλαδή στην Page της view αυτής θα εμφανίζονται μόνο προϊόντα που στοιχίζουν από τριάντα ευρώ και πάνω. Φυσικά κάθε προϊόν θα πρέπει να είναι δημοσιευμένο και να ανήκει στον τύπο New movie.



Εικόνα 23: Η view `The_front_page`

Το επόμενο display είναι το Slideshow (βλ. εικόνα 24). Το display αυτό είναι ένα attachment, δηλαδή είναι ένα display που θα εμφανίζεται μαζί με το Page display μέσα στην ίδια view το ένα κάτω από το άλλο.



Εικόνα 24: Το Slideshow της view `The_front_page`

Η χρησιμότητα του έγκειται στο γεγονός ότι εμφανίζει τα τελευταία προϊόντα που ανέβηκαν στην ιστοσελίδα και είναι άμεσα διαθέσιμα από την πρώτη σελίδα. Για την κατασκευή του display slideshow απαιτείται η εγκατάσταση της μονάδας views\_slideshow. Η μονάδα views\_slideshow είναι γραμμένη σε jQuery και μπορεί να παραμετροποιηθεί με πολλούς τρόπους. Το πιο βασικό είναι η μορφοποίηση (βλ. εικόνα 25) που έχει γίνει στο display αυτό. Για την υλοποίηση της μορφοποίησης του ιστότοπου χρησιμοποιήθηκε κατά κόρον το add-on του Firefox firebug. Με τον firebug μπορεί να γίνει επεξεργασία, αποσφαλμάτωση και παρακολούθηση του CSS της HTML και της JavaScript “ζωντανά” σε κάθε ιστοσελίδα. Η αλλαγές που γίνονται από την επεξεργασία είναι “on the fly” δηλαδή δεν αποθηκεύονται σε κάποιο αρχείο ούτε είναι μόνιμες.

```

/*-----Slideshow Theming-----*/
#views_slideshow_singleframe_pager_The_front_page-attachment_1
{
background-color: #3696D1;
border: 3px solid #000;
border-top: 3px solid #fff;
width: 800px;
border-bottom-right-radius: 25px;
border-bottom-left-radius: 25px;
text-align: center;
position: relative;
bottom: -2px;
left: 0px;
}
div.view-The-front-page div.attachment-before div.transparency
{
border-top: 3px solid #fff;
border-left: 3px solid #000;
border-right: 3px solid #000;
position: relative;
bottom: 0px;
left: 0px;
width: 800px;
height: 95px;
background-color: #6BC4F7;
bottom: -27px;
left: 0px
}
div.view-The-front-page div.attachment-before div.mycontent
{
font-family: Helvetica, Arial;
font-size: 12px;
color: #000;
position: relative;
left: 15px;
bottom: 78px;
width: 790px;
}
.imagecache .imagecache-slideshow_thumb
{
position: relative;
top: 10px;
border: 3px solid #000;
left: 0px;
top: 0px;
}
div.field-content
{
width: 0px;
}
div.views-field-teaser
{
height: 123px;
}
.imagecache .imagecache-cover_thumb
{
border: 3px solid #000;
}
.views_slideshow_singleframe_pager div a img
{
height: 50px;
width: 90px
}

```

Εικόνα 25: Ο CSS κώδικας για το Slideshow display

Το τελευταίο display είναι το Offers. Είναι ακριβώς το ίδιο ως προς τις ρυθμίσεις με το display Page μόνο που στα Filters έχει οριστεί να εμφανίζονται μόνο προϊόντα με τιμή κάτω τον τριάντα ευρώ. Έτσι καταφέρνουμε ένα διαχωρισμό των προϊόντων με βάση τη τιμή τους. Το τελικό αποτέλεσμα της view αυτής φαίνεται στην εικόνα 27. Ο κώδικας CSS για το display Page και Offers φαίνεται στην εικόνα 26.

```

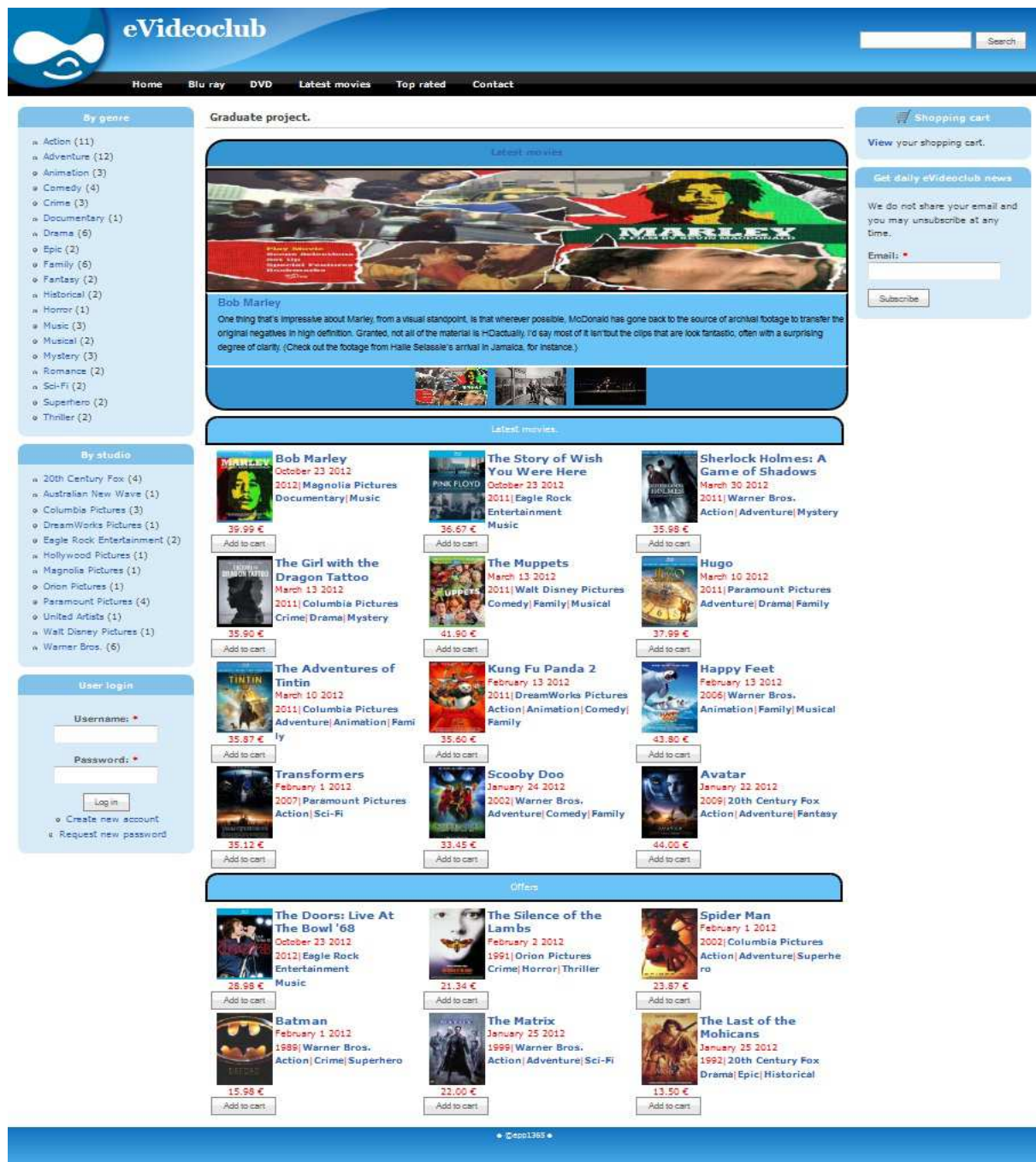
973 /*-----view-The-front-page Theming-----*/
974 div.view-The-front-page div.view-header /*view tou page*/
975 {
976   border-top-right-radius: 25px;
977   border-top-left-radius: 25px;
978   background-color: #3696D1;
979   border : 3px solid #000;
980   border-bottom : 3px solid #fff;
981   color: #fff;
982   text-align: center;
983   width: 800px;
984 }
985 div.view-The-front-page div.attachment-after div.view-header /*view twm offers*/
986 {
987   border-radius: 15px 15px 0px 0px;
988   background-color: #6BC4F7;
989   border: 3px solid #000;
990   color: #fff;
991   text-align: center;
992   width: 794px;
993 }
994 div.view-The-front-page div.attachment-before div.view-footer /*view tou slideshow*/
995 {
996   border-radius: 15px 15px 0px 0px;
997   background-color: #6BC4F7;
998   border: 3px solid #000;
999   color: #fff;
1000   text-align: center;
1001 }
1002 div.view-The-front-page div.view-footer/*view ths page*/
1003 {
1004   border-radius: 0px 0px 15px 15px;
1005   background-color: #6BC4F7;
1006   border: 3px solid #000;
1007   color: #fff;
1008   text-align: center;
1009 }
1010 div.view-The-front-page div.myview
1011 {
1012   position: relative;
1013   height:138px;
1014   left: 0px
1015 }
1016 div.view-The-front-page div.mytitle
1017 {
1018   font-size: 15px;
1019   width: 180px;
1020 }
1021 div.view-The-front-page div.mystyle
1022 {
1023   position: relative;
1024   left: 86px;
1025   bottom: 92px;
1026   height: 80px;
1027   width: 180px;
1028   font-size: 11px;
1029 }
1030 div.view-The-front-page div.myimg
1031 {
1032   position: relative;
1033   left: 13px;
1034   bottom: -7px;
1035 }
1036 div.view-The-front-page div.myprc
1037 {
1038   position: relative;
1039   left: 27px;
1040   bottom: 78px;
1041 }
1042 div.view-The-front-page div.mybtn
1043 {
1044   position: relative;
1045   left: 6px;
1046   bottom: 79px;
1047 }

```

**Εικόνα 26: Ο CSS κώδικας για τα display Page και Offers**

Από πλευράς χρήστη η view αυτή προσφέρει άμεση προσπέλαση του περιεχομένου κάνοντας απλά κλικ στον τίτλο ή την φωτογραφία κάποιου προϊόντος και άμεσα μεταφέρεται στον κόμβο που επέλεξε. Επίσης προσφέρει άμεση κατηγοριοποίηση είτε από τα blocks By genre (κατά είδος) ή By studio (κατά στούντιο) στα αριστερά της σελίδας είτε από τις λέξεις κλειδιά με βάση το έτος παραγωγής του προϊόντος δίπλα από κάθε κόμβο. Επίσης τα προϊόντα που βρίσκονται κάτω από το slideshow είναι προϊόντα που έχουν δημοσιευτεί τον τελευταίο μήνα και η τιμή τους είναι μεγαλύτερη ή ίση των τριάντα ευρώ ενώ στην από κάτω κατηγορία βρίσκονται τα προϊόντα με τιμή μικρότερη ή ίση με τριάντα ευρώ.

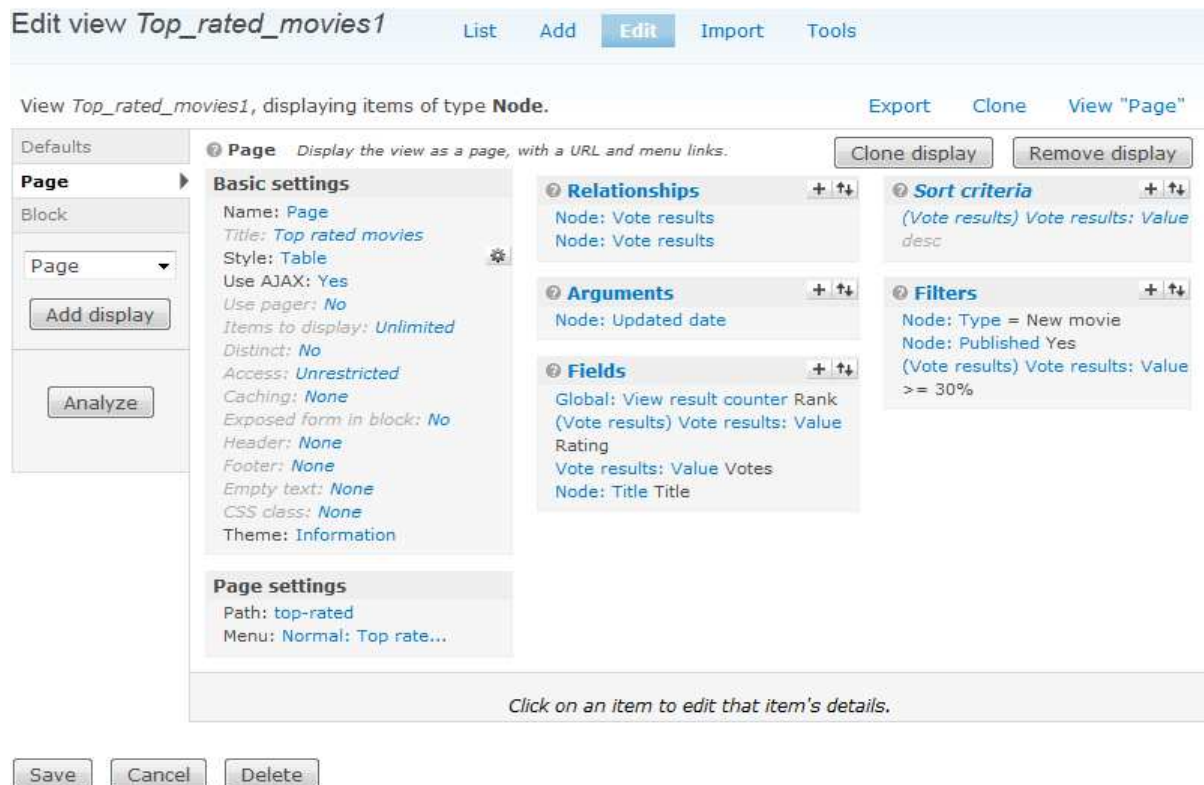
Τέλος από το κύριο μενού ο χρήστης μπορεί να πλοηγηθεί στη σελίδα με τα προϊόντα που έχουν βαθμολογηθεί με πέντε και πάνω αλλά και να κατηγοριοποιήσει το περιεχόμενο σε δύο μεγάλες κατηγορίες. Η τελευταία επιλογή είναι της επικοινωνίας με το ηλεκτρονικό κατάστημα όπου ο χρήστης μπορεί να υποβάλει αιτήσεις-σχόλια-παρατηρήσεις για διαφορά θέματα.



Εικόνα 27: Τελικό αποτέλεσμα της view The\_front\_page

Η τελευταία view που δημιουργήθηκε είναι η Top\_rated\_movies (βλ. εικόνα 28). Η view αυτή εξυπηρετεί τον σκοπό της παρουσίασης των αποτελεσμάτων της βαθμολόγησης των προϊόντων. Ωστόσο για την βαθμολόγηση των προϊόντων χρησιμοποιείται η μονάδα [Fivestar](#). Η μονάδα αυτή προσθέτει ένα γραφικό στοιχείο (widget) σε κάθε κόμβο με την μορφή αστεριών και επιτρέπει σε κάθε πιστοποιημένο χρήστη να βαθμολογεί τα προϊόντα. Αυτό το επιτυγχάνει με τη χρήση της jQuery. Για να λειτουργήσει η μονάδα αυτή απαιτεί και την ύπαρξη της μονάδας [VotingAPI](#).

Το VotingAPI είναι ένα ευέλικτο και εύκολο στη χρήση framework για την αξιολόγηση, την ψηφοφορία του περιεχομένου ενός ιστότοπου σε Drupal. Επιτρέπει στους προγραμματιστές να επικεντρωθούν στις ιδέες τους χωρίς να ανησυχούν για την εργασία της αποθήκευσης των ψήφων, τον υπολογισμό των αποτελεσμάτων, και ούτω καθεξής. Το VotingAPI συνεργάζεται με την μονάδα Views, η οποία επιτρέπει να διαμοιραστεί το περιεχόμενο ενός ιστότοπου με βάση τη συναίνεση του χρήστη. Παρέχει επίσης συναρτήσεις για την μορφοποίηση των δεδομένων της ψηφοφορίας για προβολή στους χρήστες.



**Εικόνα 28: Η view Top\_rated\_movies1**

Στο πεδίο Filters του display Page έχει επιλεγθεί να εμφανίζονται από την view όλοι οι κόμβοι που ανήκουν στον τύπο movie είναι δημοσιευμένοι και έχουν ως αποτέλεσμα από την ψηφοφορία-βαθμολόγηση πάνω από 30% με 100% το 7. Στο πεδίο Fields έχουν επιλεγεί τα πεδία για την απαρίθμηση των αποτελεσμάτων από το μεγαλύτερο στο μικρότερο, ένας μετρητής δηλαδή, το πεδίο με τις βαθμολογίες των κόμβων, το πεδίο με το σύνολο των ψήφων ανά κόμβο και τέλος το πεδίο με τον τίτλο του κάθε κόμβου. Τώρα στο πεδίο με τα Relationships έχουν επιλεγεί τα πεδία Vote results δυο φορές, μια για τον αριθμό των ψήφων και μια για τα αποτελέσματα των ψήφων. Τα Relationships λειτουργούν ως εξής: επεκτείνουν τα ερωτήματα να περιλαμβάνοντας αντικείμενα, εκτός από το ερώτημα βάσης. Όταν υπάρχουν οι σχέσεις, όλα τα πεδία (συμπεριλαμβανομένων των σχέσεων) θα αποκτήσουν ένα νέο στοιχείο φόρμας που επιτρέπει την επιλογή ποιας σχέσης θα χρησιμοποιήσουν. Για παράδειγμα με την δημιουργία του vote results για τον αριθμό των ψήφων στη φόρμα συμπλήρωσης του πεδίου vote results: value votes δημιουργήθηκε ένα νέο πεδίο που επιλέγοντάς το μας επιτρέπει να πάρουμε τα αποτελέσματα από το πεδίο votes results του πεδίου Relationships. Με άλλα λόγια οι σχέσεις μας επιτρέπουν να δημιουργούμε πεδία στο εσωτερικό των υπάρχοντων πεδίων και τη μεταξύ τους επικοινωνία. Δημιουργούν δηλαδή σύνθετα ερωτήματα προς τη βάση δεδομένων με τον συνδυασμό δεδομένων από δύο διαφορετικά πεδία. Το τελικό αποτέλεσμα από την view αυτή εμφανίζεται στην εικόνα 29.



## Top rated movies

Rank	Rating	Votes	Title
1	6.22	5	Bob Marley
2	6.02	4	The Last of the Mohicans
3	5.29	4	Forrest Gump
4	5.23	3	Braveheart
5	5.04	3	Batman
6	5.04	3	The Story of Wish You Were Here
7	4.94	4	Kung Fu Panda 2
8	4.48	4	The Matrix
9	4.39	3	Edward Scissorhands
10	4.39	3	Happy Feet
11	4.36	3	The Doors: Live At The Bowl '68
12	4.34	4	Hugo
13	4.32	4	Sherlock Holmes: A Game of Shadows
14	4.22	3	One Flew Over the Cuckoo's Nest
15	4.04	3	Transformers
16	4.04	3	Scooby Doo
17	4.04	3	The Adventures of Tintin
18	3.78	4	The Girl with the Dragon Tattoo
19	3.43	3	Mad Max
20	3.36	3	Spider Man
21	3.29	3	The Rock
22	3.08	3	The Muppets
23	3.01	3	Indiana Jones and the Last Crusade
24	2.75	3	A Space Odyssey
25	2.71	3	The Silence of the Lambs
26	2.5	3	Avatar

Εικόνα 29: Τελικό αποτέλεσμα της view Top\_rated\_movies

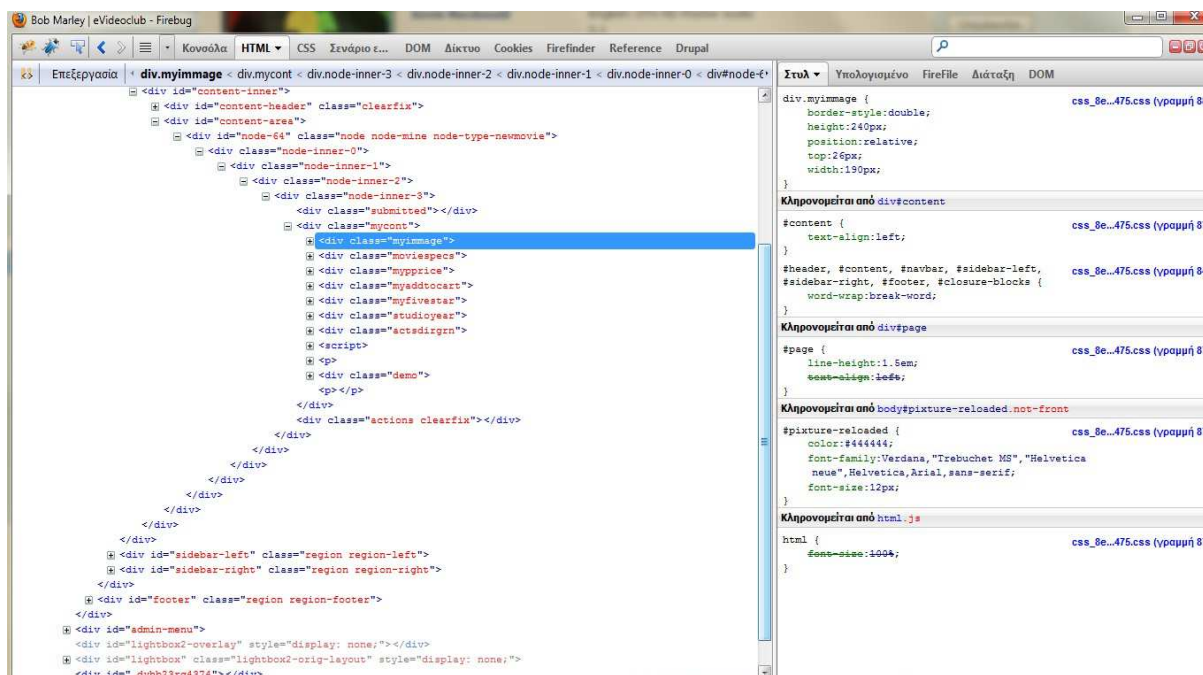
#### 4.3.2 Δοο λόγια για το CSS και τον Firebug

Η **CSS** (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) ή ( αλληλουχία φύλλων στυλ ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Ο **Firebug** είναι ένα δωρεάν και ανοιχτού κώδικα εργαλείο ανάπτυξης ιστοσελίδων που διευκολύνει την αποσφαλμάτωση, την επεξεργασία και την παρακολούθηση των CSS, HTML, DOM, XHR, και JavaScript, οποιουδήποτε δικτυακού τόπου. Προσφέρει επίσης και άλλα εργαλεία ανάπτυξης ιστοσελίδων. Ο Firebug καθιστά απλό το να βρει κανείς στοιχεία HTML που είναι βαθιά “θαμμένα” στην σελίδα. Μόλις βρεθεί αυτό που ψάχνετε, ο Firebug δίνει έναν πλούτο πληροφοριών, και επιτρέπει την επεξεργασία του HTML κώδικα “ζωντανά”-live. Στο tab HTML εμφανίζεται τη συμβαίνει στο HTML αρχείο “αυτή την στιγμή”. Επιπλέον, οι καρτέλες στη δεξιά πλευρά επιτρέπουν να γίνουν ορατές οι ιδιότητες ενός συγκεκριμένου στοιχείου, συμπεριλαμβανομένων των κανόνων CSS που το μορφοποιούν, τα εικονοστοιχεία που καθορίζουν τη θέση και το μέγεθός του, καθώς και τις ιδιότητες DOM από όπου και μπορεί κάποιος να έχει πρόσβαση από το JavaScript. Ο Firebug υπογραμμίζει τις αλλαγές στην HTML με κίτρινο χρώμα αμέσως όταν αυτά συμβαίνουν. Ο Firebug δίνει ένα πραγματικά διασκεδαστικό τρόπο για να γίνονται “πειράματα” με τις HTML αλλαγές και την άμεση ισχύ τους. Είναι δυνατό να δημιουργηθούν, να διαγραφούν ή να επεξεργαστούν ιδιότητες HTML και κείμενο απλά κάνοντας κλικ επάνω τους και tabbing από το ένα στο άλλο.

Οι αλλαγές θα εφαρμοστούν αμέσως καθώς πληκτρολογούνται. Εάν κάποιος θέλει να επιθεωρήσει άμεσα κάτι στην σελίδα και να δει τον κώδικα που βρίσκεται από πίσω, HTML και CSS, ο Firebug παρέχει την δυνατότητα αυτή μέσω του κουμπιού inspect.

Επίσης ο Firebug μέσω της καρτέλας CSS παρουσιάζει όλα όσα πρέπει να ξέρει κάποιος για το στυλ στις ιστοσελίδες του, και αν δεν του αρέσει αυτό που του παρουσιάζει, μπορεί να το αλλάξει και να δει τις αλλαγές σε ισχύ αμέσως. Ο Firebug δείχνει την αλληλουχία των κανόνων μαζί με το στυλ για κάθε στοιχείο. Οι κανόνες είναι ταξινομημένοι σε σειρά προτεραιότητας, και οι ιδιότητες που έχουν παρακαμφθεί (override) έχουν διαγραμμιστεί (stricken out). Κάθε κανόνας έχει μια σύνδεση προς τα πίσω στο αρχείο απ' όπου ήρθε μέσω της οποίας μπορεί κάποιος να κάνει κλικ για να μεταβεί στην αντίστοιχη γραμμή του αρχείου. Καθώς μετακινείται το ποντίκι πάνω από τα χρώματα και τις διευθύνσεις URL μιας εικόνας μέσα από την καρτέλα CSS, ένα πρακτικό μικρό tooltip θα εμφανιστεί με την προεπισκόπηση του χρώματος ή της εικόνας.



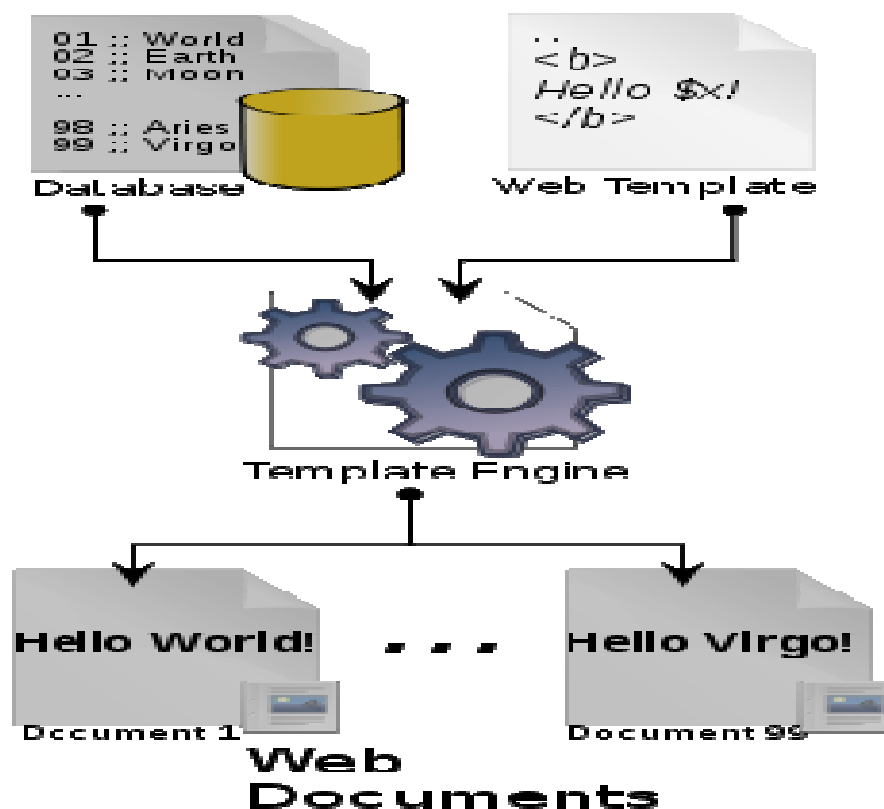
Εικόνα 30: To add on Firebug του Firefox

Η εικόνα tooltip επίσης δείχνει τις διαστάσεις του αρχείου εικόνας, η οποία μπορεί να είναι μια τεράστια εξοικονόμηση χρόνου, ειδικά όταν θα πρέπει να γράφει CSS για να κάνει ένα στοιχείο που ταιριάζει με το μέγεθος μιας εικόνας με ακρίβεια. Μπορεί κάποιος να κάνει κλικ σε οποιαδήποτε ιδιότητα CSS και ένας μικρός επεξεργαστής κειμένου θα εμφανιστεί. Καθώς πληκτρολογείται κάτι, οι αλλαγές εφαρμόζονται αμέσως. Ο Firebug συμπληρώνει αυτόματα τις τιμές των ιδιοτήτων καθώς πληκτρολογείτε μέσω μιας λίστα των πιθανών τιμών για μια ιδιότητα που επεξεργάζεστε. Μπορείτε να χρησιμοποιήσετε το πλήκτρο Esc για να ακυρώσετε τις αλλαγές που έχετε κάνει, ή να χρησιμοποιήσετε το πλήκτρο Tab για να μετακινηθείτε από τη μία ιδιότητα στην άλλη. Με το CSS, κάθε στοιχείο αποτελείται από ένθετα κουτιά γνωστά ως περιθώρια (margin), σύνορα (border), βάτες (padding), και το περιεχόμενο (content). Καθώς μετακινείτε το ποντίκι σας πάνω από ένα στοιχείο HTML σε οποιαδήποτε από τις καρτέλες του Firebug, θα δείτε τα κουτιά του στοιχείου σκιασμένα σε διάφορα χρώματα ακριβώς μέσα στη σελίδα. Δεν υπάρχει πιο γρήγορος τρόπος για να απεικονιστεί η διαφορά μεταξύ των margin και padding. Με λίγα λόγια ο Firebug είναι ένα σημαντικό και απαραίτητο εργαλείο τόσο στην ανάπτυξη και την αποσφαλμάτωση όσο και στην μορφοποίηση μιας ιστοσελίδας.

### 4.3.3 Υλοποίηση της πλήρους μορφής του προϊόντος με τη χρήση των template αρχείων

Το επόμενο στάδιο της υλοποίησης περιλαμβάνει την μορφοποίηση και την παρουσίαση του τελικού προϊόντος στην δική του σελίδα στον δικό του κόμβο, με όλες τις πληροφορίες που το αφορούν καθώς και την δυνατότητα αγοράς και κράτησης ενός προϊόντος. Για την κράτηση ενός προϊόντος έχει κατασκευαστεί μια μονάδα τα βήματα κατασκευής της οποίας θα αναλυθούν αργότερα σε αυτό το κεφάλαιο. Πριν ξεκινήσουμε θα πρέπει να γνωρίζουμε πως μπορούμε να επέμβουμε σε ένα κόμβο του Drupal, τόσο προγραμματιστικά όσο και σχεδιαστικά.

Όπως έχει ήδη αναφερθεί το Drupal χρησιμοποιεί κάποια αρχεία που ονομάζονται templates και τη μηχανή PHPTemplate ένα web template system δηλαδή. Ένα τυπικό Web template system (βλ. εικόνα 31) περιγράφει το λογισμικό και τις μεθοδολογίες που χρησιμοποιούνται για την παραγωγή ιστοσελίδων την ανάπτυξη τους σε ιστότοπους και την παράδοση τους μέσω του Διαδικτύου. Τα εν λόγω συστήματα επεξεργάζονται web templates ,χρησιμοποιώντας μια template engine.



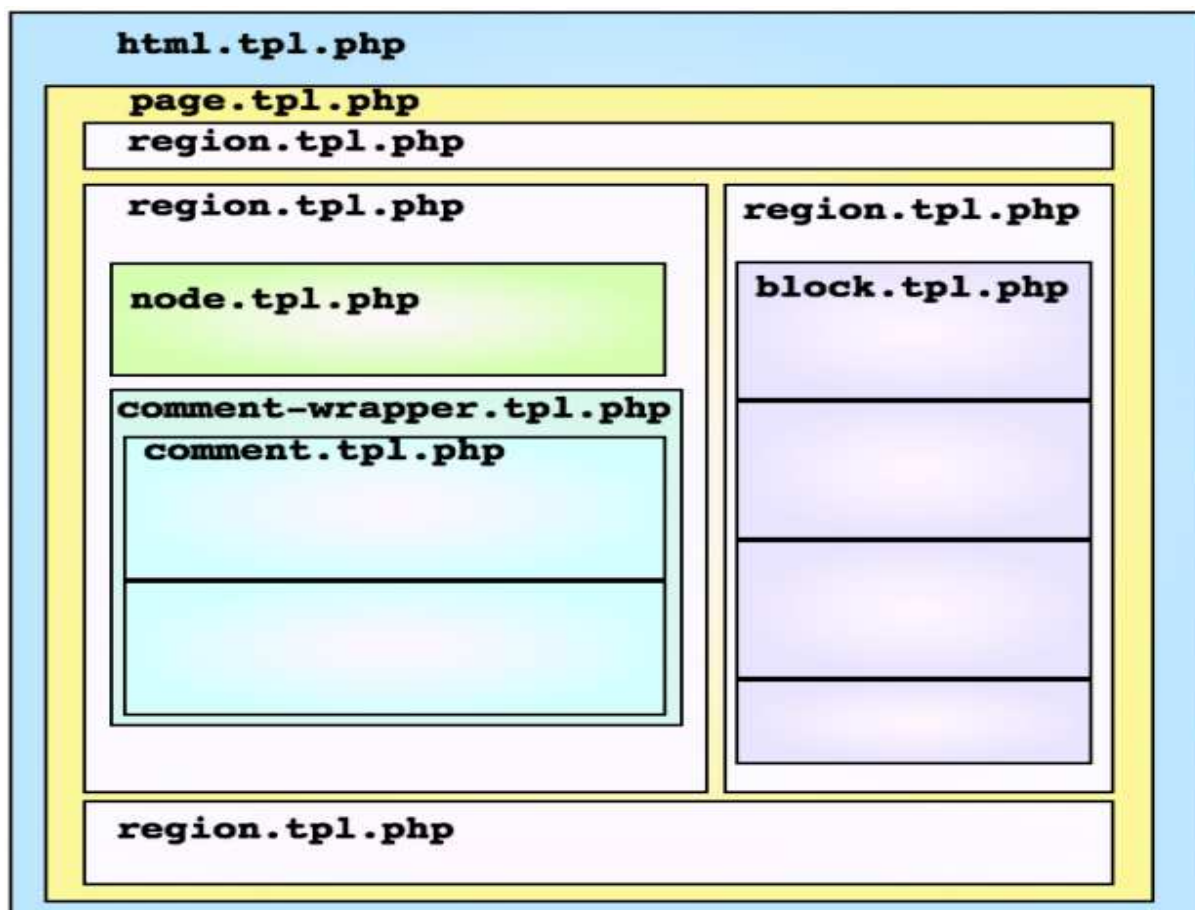
Εικόνα 31: Ένα τυπικό Web template system

Είναι εν ολίγης ένα εργαλείο δημοσίευσης ιστοσελίδων που εμφανίζεται στα συστήματα διαχείρισης περιεχομένου (Content management systems), στα πλαίσια λογισμικού (software frameworks), τους HTML συντάκτες, αλλά και σε πολλά άλλα περιβάλλοντα.

Μια μηχανή template, από την άλλη, είναι ένα λογισμικό σχεδιασμένο για την επεξεργασία web template αρχείων και πληροφοριών περιεχομένου έτσι ώστε να παράγουν web έγγραφα. Το λογισμικό αυτό τρέχει στα πλαίσια ενός web system. Μια τέτοια μηχανή μπορεί να χρησιμοποιηθεί και ως προεπεξεργαστής. Γλώσσες προγραμματισμού όπως η Perl η C και Java υποστηρίζουν επεξεργασία template αρχείων είτε εγγενώς είτε μέσω κάποιας βιβλιοθήκης. Γλώσσες προγραμματισμού όπως η JavaServer Page (JSP), η PHP και η Active Server Pages (ASP με VBscripts) αποτελούν παραδείγματα web template engines από μόνες τους.

Τέλος τα templates ή αλλιώς τα web templates είναι αρχεία που χρησιμοποιούνται για τον διαχωρισμό του περιεχομένου από την παρουσίαση στο web σχεδιασμό αλλά και για την μαζική παραγωγή web εγγράφων. Αποτελούν το βασικό συστατικό των web template systems. Τα αρχεία αυτά επεξεργάζονται από τις μηχανές template.

Στο Drupal τώρα, τα template αρχεία βρίσκονται στο κατάλογο του ενεργοποιημένου από τον διαχειριστή θέματος, τα οποία επεξεργάζονται από την μηχανή PHPTemplate. Τα αρχεία αυτά περιέχουν κώδικα HTML με μικρά κομμάτια (snippets) PHP κώδικα τα οποία μπορούν φυσικά να τροποποιηθούν ανάλογα με τις εκάστοτε απαιτήσεις του διαχειριστή ή του σχεδιαστή. Επίσης κάθε θέμα προσφέρει διαφορετικό αριθμό αρχείων, αν και είναι σχετικά εύκολο να δημιουργηθούν όσα απαιτούνται, αλλά κάποια από αυτά υπάρχουν σε κάθε θέμα όπως τα αρχεία `page.tpl.php` και `comment.tpl.php`. Κάθε αρχείο αντιστοιχεί σε ένα συγκεκριμένο τμήμα (βλ. σχήμα 24) του Drupal ιστότοπου έτσι ώστε να είναι ευκολότερη η επεξεργασία αντί τα πάντα να βρίσκονται σε ένα αχανές αρχείο. Αν για παράδειγμα θέλουμε να επεξεργαστούμε κάποιο block μέσα στον ιστότοπο θα επεξεργαστούμε το αρχείο `block.tpl.php`



**Εικόνα 32: Το region.tpl.php template του Drupal**

Στην παρούσα εργασία υπήρχε ανάγκη μορφοποίησης και εισαγωγής πληροφοριών, ανά κόμβο, του βασικού τύπου περιεχομένου `movie`. Κάθε σελίδα στο Drupal αποτελεί ένα κόμβο και κάθε κόμβος έχει ένα ID. Τα βήματα για την επεξεργασία του κόμβου είναι τα εξής: Το πρώτο βήμα είναι ο εντοπισμός του κόμβου που θέλουμε να τροποποιήσουμε. Αυτό γίνεται μέσω του μενού διαχείρισης επιλέγοντας `Content management->Content types->Edit` τον τύπο περιεχομένου που θέλουμε να επεξεργαστούμε, στην περίπτωση μας `Edit movie`. Μόλις ανοίξει η σελίδα επεξεργασίας του κόμβου αντιγράφουμε το τελευταίο πεδίο της URL, που είναι και το όνομα του κόμβου, από την γραμμή διευθύνσεων του φυλλομετρητή. Στην συνέχεια στον κατάλογο του θέματος κάνουμε αντιγραφή του αρχείου `node.tpl.php`, του αρχείου δηλαδή που αντιστοιχεί σε κάθε κόμβο και είναι υπεύθυνο για την μορφοποίηση του, και στο νέο αρχείο που δημιουργήθηκε κάνουμε επικόλληση το όνομα του κόμβου πριν από το `node.tpl.php` με την προσθήκη μιας παύλας, δηλαδή `movie-node.tpl.php`. Στο καινούργιο αυτό αρχείο μπορούμε τώρα να κάνουμε οποιαδήποτε αλλαγή χωρίς να αλλάξει τίποτα στους υπόλοιπους κόμβους του ιστότοπου πέραν των κόμβων `movie`.

Στο νέο αρχείο τώρα μπορούμε να κάνουμε οποιαδήποτε αλλαγή απαιτείται για την επίτευξη του επιθυμητού αποτελέσματος. Στο Drupal κάθε πεδίο που ανήκει σε ένα κόμβο μπορεί να προσπελαστεί και τροποποιηθεί μέσω της PHP. Το θέμα είναι να γνωρίζουμε πως θα βρούμε αυτά τα πεδία αλλά και τι τύπου είναι. Την δουλειά αυτή αναλαμβάνει να κάνει η μονάδα Devel. Η μονάδα Devel είναι ένα ανεκτίμητο εργαλείο για την κατασκευή και την κωδικοποίηση ιστοσελίδων στο Drupal. Έχει πολλά χαρακτηριστικά, όπως οι συναρτήσεις `dprint_r()`, `dpm()`, ένα κουμπί `clear_cache`, και ένα χαρακτηριστικό `generate_random_nodes`. Η μονάδα αυτή χρησιμοποιείται συνήθως για χαμηλού επιπέδου ανάπτυξη ενός ιστότοπου καθώς προσφέρει υπό την μορφή πίνακα (βλ. εικόνα 32) όλες της πληροφορίες που αφορούν ένα κόμβο. Η συνάρτηση `dprint_r()` έχει την ίδια λειτουργία με την συνάρτηση `print_r()` της PHP, τυπώνει δηλαδή πληροφορίες για μια μεταβλητή σε μορφή που μπορεί να διαβαστεί από τν άνθρωπο. Η συνάρτηση `dpm()` τυπώνει κάθε μεταβλητή απευθείας στο Drupal messages box στην ιστοσελίδα μέσω της μεταβλητής `$messages`. Το messages box είναι μια ειδικά χρωματισμένη περιοχή σε ότι θέμα Drupal και να χρησιμοποιείται, για να ανακοινώνει νέα γεγονότα όπως "Δημιουργήθηκε νέος όρος ταξινόμησης για node/4312".

field_movie_trairel (Array, 1 element)
field_subtitles (Array, 1 element)
field_duration (Array, 1 element)
field_slide (Array, 1 element)
field_video (Array, 1 element)
field_audio (Array, 1 element)
field_image_cache (Array, 1 element)
field_screenshots (Array, 4 elements)
tags (Array, 5 elements)
last_comment_timestamp (String, 10 characters ) 1333109644
last_comment_name (NULL)
comment_count (String, 1 characters ) 0
taxonomy (Array, 10 elements)
2 (Object) stdClass
tid (String, 1 characters ) 2
vid (String, 1 characters ) 2
name (String, 7 characters ) Blu-ray
description (String, 0 characters )
weight (String, 1 characters ) 0
20 (Object) stdClass
5 (Object) stdClass
177 (Object) stdClass
178 (Object) stdClass
179 (Object) stdClass
180 (Object) stdClass
181 (Object) stdClass
15 (Object) stdClass
164 (Object) stdClass

**Εικόνα 33: Τα πεδία του τύπου movie σε μορφή πίνακα**

Τώρα μπορούμε να επιλέξουμε ποια πεδία θέλουμε να τροποποιήσουμε προσπελάζοντας τα σωστά με τον κατάλληλο PHP κώδικα μέσω του αρχείου `movie-node.tpl.php`. Για να εμφανίσουμε τον πίνακα αυτό χρησιμοποιήσαμε την συνάρτηση `dpm()`, με όρισμα την μεταβλητή `$node` (βλ. εικόνα 32, γραμμή 32). Στην μεταβλητή `$node`, που είναι τύπου `object`, βρίσκονται όλα τα πεδία που αφορούν τον κάθε κόμβο. Για να μορφοποιήσουμε τα πεδία του κόμβου πάλι θα πρέπει να γράψουμε κώδικα, HTML όμως τώρα, αρχικά στο ίδιο αρχείο που θα γράψουμε και PHP, και ύστερα CSS κώδικα στο CSS αρχείο. Τα βήματα θα εναλλάσσονται εξηγώντας μια τον PHP κώδικα και μια τον HTML.

Αρχικά ορίζεται μια κλάση η `mycont` (βλ. εικόνα 32, γραμμή 31) που θα περιλαμβάνει όλα τα πεδία του κόμβου. Αυτό γίνεται με το HTML tag `div` το οποίο ορίζει περιοχές μέσα στην σελίδα που περιλαμβάνουν τα πεδία του κόμβου έτσι ώστε να είναι σαφής ο διαχωρισμός τους αλλά και πιο ευέλικτη η μορφοποίηση, δηλαδή τεμαχίζοντας την σελίδα σε μικρότερα τμήματα ελέγχουμε τις θέσεις των στοιχείων με μεγαλύτερη ακρίβεια. Βάση αυτού κάθε πεδίο ανήκει στην περιοχή του που έχει οριστεί μέσα στον κώδικα, όπου `div` λοιπόν και μια περιοχή για κάθε πεδίο. Η μορφοποίηση των πεδίων γίνεται από το αρχείο CSS και όχι από το `movie-node.tpl.php`, εδώ ορίζονται μόνο οι περιοχές που θα ανήκουν τα πεδία.

Το πρώτο πεδίο που θέλουμε να εμφανίζεται είναι το πεδίο της φωτογραφίας. Αυτό επιτυγχάνεται μέσω της PHP εντολής print. Τα επόμενα πεδία για το βίντεο τον ήχο και τους υπότιτλους έχουν ομαδοποιηθεί σε μια περιοχή για πιο εύκολη διαχείριση τους. Το επόμενο πεδίο είναι αυτό της τιμής του προϊόντος ενώ το επόμενο πεδίο είναι για την εμφάνιση του κουμπιού αγοράς του προϊόντος. Το HTML κουμπί που έχει δημιουργηθεί είναι για την μονάδα book\_movie μια μονάδα που επιτρέπει την κράτηση κάποιου προϊόντος για κάποια ημερομηνία.

```

31 <div class="mycont"> <!--katholikh klash gia ta pedia-->
32 <?php print dpm($node) //synarthsh gia thn emfanish tvn pediwn toy komvou movie
    se morfhn pinaka ?>
33 <!----- my_Code----->
34 <div class="myimage"><!--klash gia to pedio myimage-->
35 <?php print $node->field_image_cache[0][view]; //kwdikas gia thn
    emfanish tou field image ?>
36 </div>
37 <div class="moviespecs"><!--klash gia to pedio moviespecs-->
38 <?php print t('<b>Video</b><br>').$node->content[field_video][field][
    '#children']. "<br />"; //kwdikas gia thn emfanish tou field video?>
39 <?php print t('<b>Audio</b><br>').$node->field_audio[0][value]. "<br />";
    //kwdikas gia thn emfanish tou field audio?>
40 <?php print t('<b>Subtitles</b><br>').$node->field_subtitles[0][value].
    "<br />"; //kwdikas gia thn emfanish tou field subtitles?>
41 </div>
42 <div class="mypprice"><!--klash gia to pedio mypprice-->
43 <?php print $node->content[display_price]['#value']; //kwdikas gia thn
    emfanish tou field price?>
44 </div>
45 <div class="myaddtocart"><!--klash gia to pedio myaddtocart-->
46 <div class="bookbtn"><!--klash gia to pedio bookbtn-->
47 <button type="button" onclick=
    "window.location.href='book_movie/form'">Book</button> <!--kwdikas
    gia thn emfanish toy koumpiou Book-->
48 </div>
49 <?php print $node->content[add_to_cart]['#value']; // kwdikas gia thn
    emfanish tou field add_to_cart?>
50 <!-----movie studio and year----->
51 <div class="studioyear"><!--klash gia to pedio studioyear-->
52 <?php foreach ($node->taxonomy as $tid) // fortwnei ta tid gia kathe lexh
    kleidi toy ekastote komvou
53 $array[] = $tid->tid; // apothikeysh toy kathe tid (taxonomy id) se
    pinaka
54 for($i=0;$i<19;$i++) //mexri 19 lexeis kleidia mporei na exei enas komvos
55 {
56 $term1 = taxonomy_get_term($array[$i]); // epistrofh antikeimenou ma
    wash to tid toy
57 $vid1 = $term1->vid; // epilogh toy vid (vocabulary id)
58 if($vid1 == 4) // an to vid einai 4 dhladh anhkei sto lexiko studios
    emfanise to
59 echo "<a href="."?q=studios/" .rawurlencode($term1->name). ">".
    $term1->name."</a> | "; // emfanish tou onomatos ths lexeis
    kleidi ws link sthn antistoixh kathgoria
60 }
61 for($i=0;$i<19;$i++)
62 {
63 $term1 = taxonomy_get_term($array[$i]);
64 $vid1 = $term1->vid;
65 if($vid1 == 7)
66 echo "<a href="."?q=year/" . $term1->name. ">" . $term1->name. "</a>
    | ";
67 }
68 print date('F j Y', $node->created); ?>
69 </div>

```

Εικόνα 34: Πρώτο τμήμα κώδικα του αρχείου movie-node.tpl.php

Στην συνέχεια αυτό που θέλουμε να κάνουμε είναι να εμφανίζονται όλες οι λέξεις κλειδιά που αφορούν κάθε, και για κάθε, συγκεκριμένο κόμβο (βλ. γραμμή 52) και όχι όλες οι λέξεις κλειδιά από όλους τους κόμβους σε κάθε ,και για κάθε, κόμβο. Κάθε λέξη κλειδί αντιστοιχεί σε ένα λεξικό και κάθε λεξικό και κάθε λέξη κλειδί έχει ένα ID. Με βάση αυτόν τον κώδικα και τα ID καταφέρνουμε να φιλτράρουμε τις λέξεις κλειδιά για κάθε κόμβο, δηλαδή την εμφάνιση των λέξεων κλειδιών από τα αντίστοιχα λεξικά που ανήκουν μόνο στο τρέχον κόμβο. Εννοείτε ότι ο κώδικας αυτός τρέχει για κάθε κόμβο.

Κάθε κόμβος είναι μια κλάση και μέσω της μεταβλητής \$node μπορούμε να προσπελάσουμε όλα τα πεδία αυτής της κλάσης, το ίδιο συμβαίνει και στις προηγούμενες γραμμές κώδικα. Από την στιγμή που θέλουμε να φιλτράρουμε τις λέξεις κλειδιά με βάση το ID τους πρέπει από κάπου να πάρουμε αυτά τα ID. Τα ID αυτά βρίσκονται στον πίνακα αντικειμένων taxonomy τον οποίο δεν μπορούμε να χρησιμοποιήσουμε άμεσα διότι περιέχει και στοιχεία που δεν μας είναι χρήσιμα, έτσι με την εντολή foreach() επιλέγουμε μόνο τα απαραίτητα στοιχεία που μας χρειάζονται, όπου και τα στοιχεία αυτά αφορούν μόνο τον συγκεκριμένο κόμβο που έχει φορτωθεί κάποια δεδομένη στιγμή, ή αλλιώς τα στοιχεία μόνο του τρέχον κόμβου (έτσι και καταφέρνουμε να παίρνουμε μόνο, όλες, τις λέξεις κλειδιά που αφορούν μόνο τον τρέχον κόμβο· φιλτράρουμε-διαλέγουμε δηλαδή τις λέξεις κλειδιά που θέλουμε). Αποθηκεύουμε λοιπόν όλα τα απαραίτητα ID σε ένα νέο πίνακα, ακεραίων όμως αυτή τη φορά.

Στη συνέχεια μέσα σε μια for (βλ. γραμμή 54) καλούμε την συνάρτηση του Drupal API [taxonomy\\_get\\_term\(\)](#) (βλ. γραμμή 56) όπου η συνάρτηση αυτή παίρνει ως όρισμα το tid κάθε λέξης κλειδί και επιστρέφει το αντικείμενο στο οποίο ανήκει η κάθε λέξη κλειδί. Κάθε \$term1 τώρα είναι ένα αντικείμενο που μέσα του έχει το πεδίο vid (vocabulary ID) το πεδίο δηλαδή με βάση το οποίο θα γίνεται ακόμη ένα φιλτράρισμα των όρων-λέξεων κλειδιών. Αν δηλαδή μια λέξη κλειδί ανήκει στο λεξικό Actors θα εμφανιστεί στο πεδίο με τους ηθοποιούς, αν ανήκει στο λεξικό Director θα εμφανίζεται στο πεδίο με τους σκηνοθέτες. Αυτό επαναλαμβάνεται για όλα τα πεδία (βλ. γραμμές 61, 71, 81 και 91 μέχρι 97) του κάθε κόμβου. Έχει οριστεί κάθε κόμβος να έχει μέχρι και δεκαεννέα λέξεις κλειδιά, αυτό όμως αλλάζει εύκολα ορίζοντας τον έλεγχο στη for να σταματάει όπου θέλουμε. Όπως είναι ορατό μέσα στο movie-node.tpl.php αρχείο μπορούμε να γράφουμε τόσο PHP όσο και HTML κώδικα. Έτσι ορίζονται και κάποιες επικεφαλίδες για τα πεδία του κόμβου.

Ένα ακόμη πεδίο που χρησιμοποιείται είναι το πεδίο δημιουργίας του κόμβου, πότε δηλαδή αυτός δημιουργήθηκε από τον χρήστη. Για την εμφάνιση αυτού του πεδίου χρησιμοποιείται η PHP συνάρτηση [date\(\)](#). Η συνάρτηση (string date (string \$format [, int \$timestamp = time() ] ) ) αυτή επιστρέφει ένα string διαμορφωμένο σύμφωνα με τα δοσμένα ορίσματα μορφοποίησης που εισάγει ο χρήστης χρησιμοποιώντας το δοσμένο ακέραιο timestamp ή την τρέχουσα ώρα αν δεν έχει οριστεί timestamp. Επίσης η συνάρτηση [rawurlencode\(\)](#) (βλ. εικόνα 34) (string rawurlencode ( string \$str )) είναι μια συνάρτηση που κωδικοποιεί ένα string σύμφωνα με το [RFC 3986](#).

```
<?php
echo '<a href="http://example.com/department_list_script/',
    rawurlencode('sales and marketing/Miami'), '>';
?>
```

The above example will output:

```
<a href="http://example.com/department_list_script/sales%20and%20marketing%2FMiami">
```

### Εικόνα 35: Η συνάρτηση rawurlencode() της PHP

Στο δεύτερο τμήμα του κώδικα (βλ. εικόνα 35, γραμμή 98) έχει οριστεί να εμφανίζεται και το πεδίο της μονάδας Fivestar η οποία μονάδα είναι υπεύθυνη για την βαθμολόγηση του περιεχομένου μέσω ενός γραφικού (widget). Στη γραμμή 104 ξεκινάει ο κώδικας του widget tabs της βιβλιοθήκης jQuery της JavaScript. Το γραφικό αυτό των tabs θα εμφανίζεται στο υποσέλιδο της σελίδας κάθε κόμβου και θα περιέχει για το σώμα του κόμβου, την περιγραφή δηλαδή του προϊόντος, το τρέιλερ του προϊόντος και τα στιγμιότυπα του προϊόντος. Ο αριθμός των στιγμιότυπων που θα εμφανίζονται έχει οριστεί να είναι τρία. Στο αρχείο movie-node.tpl.php περιλαμβάνεται μόνο το τι θέλουμε να περιέχει το widget tabs, ο κώδικας του βρίσκεται στη διαδρομή sites/all/themes/mytheme/js στο αρχείο jquery-ui-1.8.20.custom.min.

Επίσης για την εισαγωγή JavaScript και της βιβλιοθήκης jQuery στο θέμα που χρησιμοποιούμε απαιτείται η συνάρτηση `drupal_add_js()`. Η συνάρτηση αυτή χρησιμοποιείται για την προσθήκη JavaScript αρχείων, ρυθμίσεων ή και γραμμών κώδικα σε μια σελίδα. Η συνάρτηση αυτή καλείται από το αρχείο που υπάρχει στο κατάλογο εγκατάστασης του θέματος του ιστοτόπου `template.php`. Ακόμη μια απαραίτητη συνάρτηση είναι η `drupal_add_css()`. Η συνάρτηση αυτή χρησιμοποιείται για την προσθήκη CSS αρχείων στο προεπιλεγμένο θέμα.

```

70 <!-------movie actors----->
71 <div class="actsdirgrn"><!--klash gia ta pedia actors director genre-->
72 <h3>Actors</h3>
73 <?php for($i=0;$i<19;$i++)
74 {
75     $sterm1 = taxonomy_get_term($array[$i]);
76     $vid1 = $sterm1->vid;
77     if($vid1 == 5)
78         echo "<a href='\".\"?q=actors/\".rawurlencode($sterm1->name).
79             \"%20\".\">\".\"$sterm1->name.\"</a><br />\";
80 }?>
81 <!-------movie director----->
82 <h3>Director</h3>
83 <?php for($i=0;$i<19;$i++)
84 {
85     $sterm1 = taxonomy_get_term($array[$i]);
86     $vid1 = $sterm1->vid;
87     if($vid1 == 6)
88         echo "<a href='\".\"?q=director/\".rawurlencode($sterm1->name).\"%20\".
89             \"%>\".\"$sterm1->name.\"</a><br />\";
90 }?>
91 <!-------movie genre----->
92 <h3>Movie genre</h3>
93 <?php for($i=0;$i<19;$i++)
94 {
95     $sterm1 = taxonomy_get_term($array[$i]);
96     $vid1 = $sterm1->vid;
97     if($vid1 == 3)
98         echo "<a href='\".\"?q=categorys/\".$sterm1->name.\">\".\"$sterm1->name.
99             \"</a><br />\";
100 }?>
101 <?php print "<br />\".t('<b>Movie rating</b><br />').$node->content[fivestars_widget
102 ]['#value']; ?>
103 </div>
104 <script> <!-- eisagwgh tou jquery widget tabs->
105     $(function() {
106         $( "#tabs" ).tabs();
107     });
108 </script>
109 <p><br /><br /><br /><br /><br />
110 <div class="demo"><!--klash gia to pedio demo-->
111 <div id="tabs">
112     <ul>
113         <li><a href="#tabs-1">Overview</a></li>
114         <li><a href="#tabs-2">Trailer</a></li>
115         <li><a href="#tabs-3">Screenshots</a></li>
116     </ul>
117     <div id="tabs-1"><?php print "<b>\".$node->title.\" Overview :</b>\".$node
118     ->content[body]['#value']; //eisagwgh tou pediou body ?></div>
119     <div id="tabs-2"><?php print "<b>\".$node->title.\" Trailer :</b>\".$node->
120     field_movie_trairel[0][view]; //eisagwgh tou pediou trailer ?></div>
121     <div id="tabs-3"><?php print "<b>\".$node->title.\" Screenshots :</b>\";
122     for($i=0;$i<3;$i++) print "<p>\".$node->field_screenshots[$i][view].
123     "</p>\"; //eisagwgh tou pediou screenshot(mexri tria epitrepetai)?></div>
124 </div>
125 </div></p>

```

Εικόνα 36: Δεύτερο τμήμα κώδικα του αρχείου `movie-node.tpl.php`

Τέλος όλος ο απαραίτητος CSS κώδικας (βλ. εικόνα 36) που γράφτηκε για την μορφοποίηση του τελικού κόμβου βρίσκεται στο αρχείο `style.css` πάλι στον κατάλογο εγκατάστασης του προεπιλεγμένου θέματος. Το προεπιλεγμένο θέμα είναι το `picture_reloaded`. Οι τελική μορφοποίηση, δηλαδή οι τελικές θέσεις των στοιχείων, τα περιγράμματα, οι γραμματοσειρές, τα χρώματα, τα πλάτη και τα μήκη των στοιχείων καθορίστηκαν μέσω του σημαντικότερου για web development και design εργαλείου της Mozilla τον Firebug. Κάθε στοιχείο που έχει οριστεί μέσα σε μια κλάση μέσω της HTML μπορεί να μορφοποιηθεί μέσω του firebug στιγμιαία (on the fly) και έπειτα να γραφούν οι αλλαγές αυτές στο κατάλληλο αρχείο, το αρχείο `style.css` στην παρούσα περίπτωση.



Το τελικό αποτέλεσμα όλης της διαδικασίας που αναλύθηκε, όσο το δυνατό σε μεγαλύτερο αλλά και ταυτόχρονα σε κατανοητό βάθος, στις παραγράφους των προηγούμενων σελίδων αυτού του κεφαλαίου παρουσιάζεται στην εικόνα 37. Κατά την επίσκεψη του ο χρήστης στην σελίδα του τελικού κόμβου μπορεί να βρει **συγκεντρωτικά** και **άμεσα** όλες τις απαραίτητες **πληροφορίες**, άλλωστε αυτός είναι και ο σκοπός, για το προϊόν που τον ενδιαφέρει αλλά και να προχωρήσει στην αγορά ή την κράτηση αυτού. Οι **πληροφορίες** αυτές αφορούν τα χαρακτηριστικά της ταινίας, όπως η κωδικοποίηση και η ανάλυση του βίντεο, τις γλώσσες για τις οποίες είναι διαθέσιμη η ταινία και τα format του ήχου, τους υπότιτλους της ταινίας, την περιγραφή της ταινίας που βρίσκεται στο tab description, το trailer της ταινίας που βρίσκεται στο tab trailer και τα στιγμιότυπα της ταινίας που βρίσκονται στο tab screenshot.

```

1438  /*-----node NewMovieview Theming-----*/
1439  div.myimage {
1440  border-style:double;
1441  height:240px;
1442  position:relative;
1443  top:26px;
1444  width:190px;
1445  }
1446  div.moviespecs {
1447  left:675px;
1448  position:absolute;
1449  top:79px;
1450  width:189px;
1451  }
1452  div.mypprice{
1453  color:red;
1454  font-size:10px;
1455  left:301px;
1456  position:absolute;
1457  top:324px;
1458  }
1459  div.myaddtocart {
1460  left:313px;
1461  position:absolute;
1462  top:343px;
1463  }
1464  div.myfivestars {
1465  left:675px;
1466  position:absolute;
1467  top:325px;
1468  }
1469  div.studioyear{
1470  position:absolute;
1471  top:53px;
1472  }
1473  div.actsdirgrn{
1474  left:475px;
1475  position:absolute;
1476  top:64px;
1477  width:auto;
1478  }
1479  div.bookbtn{
1480  left:14px;
1481  position:absolute;
1482  top:28px;
1483  }

```

**Εικόνα 37: Ο κώδικας CSS για το movie-node.tpl.php**

Επίσης ο χρήστης μπορεί άμεσα να **πληροφορηθεί** για τους πρωταγωνιστές της ταινίας, τον σκηνοθέτη, το είδος ή τα είδη στα οποία ανήκει η ταινία, την εταιρία και το έτος παραγωγής της ταινίας, την ημερομηνία δημοσίευσης της ταινίας στον ιστότοπο, την τιμή πώλησης της ταινίας αλλά και την βαθμολογία της ταινίας που έχει διαμορφωθεί από άλλους χρήστες, **πληροφορίες** ικανοποιητικά αρκετές ώστε κάποιος να έχει μια πλήρη εικόνα του προϊόντος και να προχωρήσει στην αγορά του. Σημαντική πληροφόρηση μπορεί επίσης να λάβει ο χρήστης με την χρήση των λέξεων κλειδιά. Αυτές αφορούν τους ηθοποιούς, τον σκηνοθέτη, την εταιρία και το έτος παραγωγής, αλλά και το είδος της ταινίας. Μπορεί δηλαδή, άμεσα ο χρήστης, επιλέγοντας μεταξύ των λέξεων κλειδιά, να πληροφορηθεί για το ποιες ταινίες έχουν παραχθεί στο ίδιο έτος με την ταινία που κάνει προεπισκόπηση, ποιες ταινίες έχουν παραχθεί από την ίδια εταιρία παραγωγής, σε ποιες άλλες ταινίες έχει πρωταγωνιστήσει ο κάθε ηθοποιός, ποιες άλλες ταινίες έχει σκηνοθετήσει ο ίδιος σκηνοθέτης, αλλά και ποιες άλλες ταινίες ανήκουν στο ίδιο είδος.

The screenshot shows the eVideoclub website interface. At the top, there is a navigation bar with links for Home, Blu ray, DVD, Latest movies, Top rated, and Contact. A search bar is located in the top right corner. The main content area is divided into several sections:

- By genre:** A list of movie genres including Action (11), Adventure (12), Animation (3), Comedy (4), Crime (3), Documentary (1), Drama (6), Epic (2), Family (6), Fantasy (2), Historical (2), Horror (1), Music (3), Musical (2), Mystery (3), Romance (2), Sci-Fi (2), Superhero (2), and Thriller (2).
- By studio:** A list of studios including 20th Century Fox (4), Australian New Wave (1), Columbia Pictures (3), DreamWorks Pictures (1), Eagle Rock Entertainment (2), Hollywood Pictures (1), Magnolia Pictures (1), Orion Pictures (1), Paramount Pictures (4), United Artists (1), Walt Disney Pictures (1), and Warner Bros. (6).
- User login:** A section with fields for Username and Password, and a Log in button. Below it are links for 'Create new account' and 'Request new password'.
- Home:** The main featured movie, 'Forrest Gump', is displayed. It includes a movie poster, the title, release year (1994), and date (January 20 2012). Below the poster are buttons for 'Add to cart' (price 29.99 €) and 'Book'. To the right of the poster, there are sections for 'Actors' (Mykelti Williamson, Sally Field, Tom Hanks), 'Director' (Robert Zemeckis), 'Movie genre' (Comedy, Drama, Romance), and 'Movie rating' (Average: 5.3 (4 votes)).
- Technical details:** A section on the right provides information about the video codec (MPEG-4 AVC), resolution (1080p), aspect ratio (2.35:1), original aspect ratio (2.39:1), audio (DTS-HD Master, 5.1 French, 5.1 Spanish, 5.1 Digital), and subtitles (English, French, Spanish, Portuguese).
- Overview:** A section titled 'Forrest Gump Overview:' provides a synopsis of the film, mentioning it is based on the 1986 novel by Winston Groom and directed by Robert Zemeckis. It includes a photo of young Forrest Gump and Jenny sitting on a tree branch.
- Shopping cart:** A 'Shopping cart' button is located in the top right, with a link to 'View your shopping cart.' Below it is a 'Get daily eVideoclub news' section with an email subscription form.

At the bottom of the page, there is a footer with the text '©epp1365'.

Εικόνα 38: Τελικό αποτέλεσμα του movie-node.tpl.php

Στην δεξιά πλευρά της ιστοσελίδας υπάρχει το block του καλαθιού αγοράς που ενημερώνει τον χρήστη για τις αγορές που πρόκειται να κάνει. Ακριβώς από κάτω βρίσκεται το block του newsletter του ιστότοπου. Το block αυτό ενεργοποιείται μέσω της μονάδας [Simplenews](#). Η μονάδα Simplenews δημοσιεύει και στέλνει ενημερωτικά δελτία σε μια λίστα από συνδρομητές. Και τα δύο είδη χρηστών, ανώνυμοι και εξουσιοδοτημένοι, μπορούν να επιλέξουν να εγγραφούν, δίνοντας την διεύθυνση του ηλεκτρονικού τους ταχυδρομείου, σε διάφορες λίστες. Μερικά χαρακτηριστικά της μονάδας αυτής είναι η αποστολή ενός ολόκληρου κόμβου ως newsletter στους συνδρομητές, πολλαπλές κατηγορίες newsletters με ξεχωριστές ρυθμίσεις, block ανά κατηγορία και σελίδα, μαζική διαχείριση όλων των εγγεγραμμένων χρηστών στη λίστα, συμπεριλαμβανομένης μαζικής εγγραφής και εξαγωγής της λίστας αλλά και ρυθμίσεις για επιβεβαίωσης εγγραφής σε ανώνυμους χρήστες.

#### 4.3.4 Υλοποίηση της custom μονάδας book\_form

Όπως έχει ήδη αναφερθεί και σε προηγούμενα κεφάλαια της παρούσας εργασίας, στο Drupal μπορεί να κατασκευαστεί μια οποιαδήποτε μονάδα που θα λειτουργεί με βάση τις απαιτήσεις του σχεδιαστή του συστήματος. Στην παρούσα εργασία υπήρχε η ανάγκη για την δημιουργία μιας φόρμας μέσω της οποίας ο χρήστης θα μπορούσε να κάνει κράτηση ενός προϊόντος για κάποια συγκεκριμένη ημερομηνία. Τα βήματα κατασκευής της μονάδας αυτής περιγράφονται στις επόμενους παραγράφους.

Για την κατασκευή μιας οποιαδήποτε μονάδας στο Drupal ακολουθείτε μια συγκεκριμένη προπαρασκευαστική διαδικασία. Η διαδικασία αυτή θα αναλυθεί μέσω της κατασκευής της μονάδας που δημιουργήθηκε για την παρούσα εργασία. Το πρώτο που πρέπει να δημιουργηθεί είναι ένας κατάλογος με το όνομα της μονάδας που πρόκειται να κατασκευαστεί στην διαδρομή sites/all/modules της βασικής εγκατάστασης του Drupal. Το όνομα αυτό είναι book\_form (χωρίς κενά διαστήματα). Μέσα στον κατάλογο αυτό θα πρέπει υποχρεωτικά να υπάρχουν δυο αρχεία, ένα αρχείο με την κατάληξη .info και ένα αρχείο με την κατάληξη .module. Το όνομα και στα δύο αρχεία θα πρέπει να είναι το ίδιο με το όνομα του καταλόγου. Το .info (βλ. εικόνα 38) αρχείο είναι ένα αρχείο που περιέχει πληροφορίες για την μονάδα όπως, το όνομα της, την έκδοση του Drupal με την οποία θα είναι συμβατή, την περιγραφή του τι κάνει η μονάδα αλλά και κάποιες ακόμη όχι τόσο σημαντικές πληροφορίες. Το άλλο αρχείο (βλ. εικόνα 39,40) περιέχει τον κώδικα που θα γραφεί από τον προγραμματιστή και θα εκτελεστεί από το Drupal.

```

1 | ; $Id$
2 |
3 | name = Book movie form
4 | description = A form for booking movies.
5 | core = 6.x
6 | package = My module
7 |
8 |

```

**Εικόνα 39: Αρχείο .info της custom μονάδας book\_form**

Στο πρώτο τμήμα κώδικα (βλ. εικόνα 39 γραμμή 7) το πρώτο που πρέπει να γίνει είναι να κληθεί η συνάρτηση [hook\\_menu\(\)](#) (όπου hook αντικαθιστάτε με το όνομα της μονάδας). Η συνάρτηση αυτή ορίζει το μονοπάτι μέσω του οποίου θα εμφανίζεται η φόρμα στον χρήστη, με άλλα λόγια δημιουργεί μια νέα σελίδα στον ιστότοπο όπου θα βρίσκεται η φόρμα κρατήσεων. Στο σώμα της συνάρτησης ορίζεται ένας πίνακας και μια μεταβλητή πίνακα με το μονοπάτι προς τη φόρμα. Στη μεταβλητή πίνακα ορίζεται ο τίτλος της σελίδας, η συνάρτηση που θα καλείται από την σελίδα αυτή, τα δικαιώματα πρόσβασης στην σελίδα αυτή, μια περιγραφή και η κατοχύρωση του μονοπατιού ώστε να κληθεί η σωστή συνάρτηση όταν προσπελαθεί το μονοπάτι αυτό. Η επόμενη συνάρτηση είναι η [hook\\_form\(\)](#) (βλ. γραμμή 24). Η συνάρτηση αυτή καλεί τον κατασκευαστή φορμών του Drupal μέσω της συνάρτησης [drupal\\_get\\_form\(\)](#) η οποία παίρνει ως όρισμα το όνομα της φόρμας που πρόκειται να κατασκευαστεί. Η επόμενη συνάρτηση είναι η συνάρτηση που κατασκευάζει την φόρμα όταν κληθεί από την [drupal\\_get\\_form\(\)](#). Στο σώμα της συνάρτησης ορίζεται μια μεταβλητή (βλ. γραμμή 36) \$script που περιέχει τον κώδικα για το jQuery widget Datepicker. Το script ορίζεται ως string και εισάγεται στην φόρμα μέσω της συνάρτησης [drupal\\_add\\_js\(\)](#). Στον κώδικα για το script αυτό έχουν οριστεί οι επιλογές (options) έτσι ώστε, η ημερομηνία να έχει την μορφή ημέρα-μήνα-έτος, ο χρήστης να μπορεί να επιλέξει από την τρέχων ημερομηνία και μετά και όχι ημερομηνίες προηγούμενες της τρέχων ημερομηνίας και η προεπιλεγμένη ημερομηνία να είναι η τρέχων. Στη γραμμή 45 ορίζεται το όνομα της φόρμας και από κει και κάτω τα πεδία της φόρμας. Το πρώτο πεδίο είναι το μικρό όνομα, το επόμενο πεδίο το επίθετο, το επόμενο πεδίο είναι το email, το επόμενο πεδίο είναι ο κωδικός μέλους και το τελευταίο πεδίο είναι το πεδίο για την ημερομηνία. Όλα τα πεδία είναι textfields με μέγιστο μήκος και αριθμό χαρακτήρων είκοσι. Επίσης τα πεδία email, κωδικός και ημερομηνία είναι υποχρεωτικά. Το τελευταίο πεδίο (βλ. γραμμή 91) είναι το κουμπί submit.

```

1 <?php
2 // $Id:$
3 /**
4  * This function defines the URL to the page created etc.
5  * See http://api.drupal.org/api/function/hook_menu/6
6  */
7 function book_form_menu() {
8     $items = array();
9     $items['book_movie/form'] = array(
10      'title' => t('Movies booking form'),
11      'page callback' => 'book_form_form',
12      'access arguments' => array('access content'),
13      'description' => t('Movies rental form'),
14      'type' => MENU_CALLBACK,
15    );
16    return $items;
17  }
18  /**
19  * This function gets called in the browser address bar for:
20  * "http://yourhost/book_form/form" or
21  * "http://yourhost/?q=book_form/form". It will generate
22  * a page with this form on it.
23  */
24  function book_form_form() {
25
26      // This form calls the form builder function via the
27      // drupal_get_form() function which takes the name of this form builder
28      // function as an argument. It returns the results to display the form.
29      return drupal_get_form('book_form_my_form');
30  }
31  /**
32  * This function is called the "form builder". It builds the form.
33  * Notice, it takes one argument, the $form_state
34  */
35  function book_form_my_form($form_state) {
36      $script = '$(function() {
37          $( "#edit-date" ).datepicker({
38              dateFormat: "dd/mm/yy" ,
39              changeMonth: true,
40              changeYear: true,
41              yearRange: "2012:2013",
42              minDate: "+0d" });});';
43      drupal_add_js($script, 'inline');
44      // This is the first form element. It's a textfield with a label, "Name"
45      $form['name'] = array(
46          '#type' => 'fieldset',
47          '#title' => t('Book a movie'),
48          '#collapsible' => TRUE,
49          '#collapsed' => FALSE,
50      );
51
52      $form['name']['first'] = array(
53          '#type' => 'textfield',
54          '#title' => t('First name'),
55          '#description' => "Please enter your first name.",
56          '#required' => FALSE,
57          '#size' => 20,
58          '#maxlength' => 20,
59      );
60      $form['name']['last'] = array(
61          '#type' => 'textfield',
62          '#title' => t('Last name'),
63          '#description' => "Please enter your last name.",
64          '#required' => FALSE,
65          '#size' => 20,
66          '#maxlength' => 20,
67      );
68      $form['name']['email'] = array(
69          '#type' => 'textfield',
70          '#title' => 'Email',
71          '#description' => "Please enter your email address.",
72          '#required' => TRUE,
73          '#size' => '20',
74      );
75      $form['name']['code'] = array(
76          '#type' => 'textfield',
77          '#title' => t('Member code'),
78          '#description' => "Please enter your member code.",
79          '#required' => TRUE,
80          '#size' => 20, // added
81          '#maxlength' => 20, // added
82      );

```

Εικόνα 40: Πρώτο τμήμα κώδικα μονάδας book\_form

```

83 $form['name']['date'] = array(
84   '#type' => 'textfield',
85   '#title' => t('Pick date'),
86   '#description' => "DVDs can be rent for five days,Blu-rays for 2.",
87   '#required' => TRUE,
88   '#size' => 20, // added
89   '#maxlength' => 20, // added
90 );
91 $form['submit'] = array(
92   '#type' => 'submit',
93   '#value' => 'Book it',
94 );
95 return $form;
96 }
97 function book_form_my_form_validate($form, &$form_state) {
98
99   //email validation
100   $valid_email = $form_state['values']['email'];
101   if (!valid_email_address($valid_email)) {
102     form_set_error('email', 'Your email address ' . $valid_email . ' is invalid');
103   }
104   //code member validation
105   $valid_code= $form_state['values']['code'];
106   if (!is_numeric($valid_code)) {
107     form_set_error('code', 'Your member code must be numeric');
108   }
109   //code member validation
110   if ($valid_code < 0 || $valid_code > 5000) {
111     form_set_error('code', 'Your member code must be between 0 and 5000');
112   } //firts name validation
113   $valid_name = $form_state['values']['first'];
114   $valid_name1 = $form_state['values']['last'];
115   if (is_numeric($valid_name) || is_numeric($valid_name1)) {
116     form_set_error('name', 'Your first name and last name must be no numeric');
117   }
118 }
119 function book_form_my_form_submit($form, &$form_state) {
120   drupal_set_message(t('The movie has been successfully booked at ' . date('d/m/y') . ' for the ' .
    $form_state['values']['date']));
121 }

```

Εικόνα 41: Δεύτερο τμήμα κώδικα μονάδας book\_form

Στη γραμμή 97 ορίζεται η συνάρτηση που θα ελέγχει την εγκυρότητα της φόρμας. Η συνάρτηση αυτή είναι η [hook\\_validate\(\)](#) (όπου hook το όνομα της φόρμας και το όρισμα \$form ο κόμβος που πρόκειται να ελεγχτεί-βλ. και \$items). Πρώτα ελέγχεται το πεδίο email (βλ. γραμμή 100) μέσω της συνάρτησης [valid\\_email\\_address\(\)](#). Αν δεν είναι έγκυρη η διεύθυνση email τότε καλείται η συνάρτηση [form\\_set\\_error\(\)](#) η οποία εμφανίζει ένα μήνυμα λάθους στην περιοχή μηνυμάτων του Drupal. Έπειτα ελέγχεται το πεδίο με τον κωδικό για το αν αυτό που εισήχθηκε στο πεδίο είναι αριθμητικό ή όχι μέσω της PHP συνάρτησης [is\\_numeric\(\)](#) (βλ. γραμμή 105), αν δεν είναι εμφανίζεται ένα μήνυμα λάθους. Στην συνέχεια ελέγχεται πάλι το πεδίο με τον κωδικό μέλους (βλ. γραμμή 110) αλλά αυτή την φορά για το αν η τιμή του κωδικού βρίσκεται μέσα στα επιτρεπτά όρια, μεταξύ μηδέν δηλαδή και πέντε χιλιάδες. Τελευταία ελέγχονται τα πεδία του μικρού και του μεγάλου ονόματος για το αν αυτά είναι αριθμοί ή όχι. Ελέγχονται από την συνάρτηση [is\\_numeric\(\)](#) και αν είναι αριθμοί (αρκεί ένα από τα δυο να είναι) τότε εμφανίζεται ένα μήνυμα λάθους. Τέλος η φόρμα κατοχυρώνεται (βλ. γραμμή 119), αφού έχει κριθεί έγκυρη, από την συνάρτηση [node\\_form\\_submit\(\)](#) (όπου node\_form το όνομα της φόρμας) και εμφανίζει ένα μήνυμα μέσω της συνάρτησης [drupal\\_set\\_message\(\)](#) που ενημερώνει το χρήστη ότι η κράτηση του ολοκληρώθηκε επιτυχώς καθώς και την ημερομηνία που έγινε η κράτηση αλλά και την ημερομηνία για την οποία έγινε η κράτηση. Το τελικό αποτέλεσμα της φόρμας φαίνεται στην εικόνα 41.

The screenshot shows the 'Movies booking form' on the eVideoclub website. The form is titled 'Home Movies booking form' and includes a 'Book a movie' dropdown menu. The form fields are: 'First name' (filled with 'simon'), 'Last name' (filled with 'templar'), 'Email' (filled with 's\_temp@yahoo.com'), 'Member code' (filled with '1234'), and 'Pick date' (with a calendar widget showing November 2012). The calendar is currently displaying the 5th of November. On the left, there are navigation menus for 'By genre' and 'By studio'. On the right, there is a 'Shopping cart' section showing 0 items and a total of 0.00 €, and a 'Get daily eVideoclub news' section with an 'Unsubscribe' button.

**Εικόνα 42: Τελικό αποτέλεσμα της μονάδας book\_form**

Τέλος στο βασικό μενού υπάρχει και το link επικοινωνίας contact. Πρόκειται για μια ακόμη φόρμα (βλ. εικόνα 42) η οποία όμως αυτή την φορά δημιουργήθηκε από την μονάδα Webform. Η μονάδα Webform είναι για την παραγωγή φορμών σε Drupal. Μετά την υποβολή κάποιου αιτήματος, οι χρήστες μπορούν να στείλουν ένα e-mail "απόδειξη", καθώς και να αποστείλουν μια κοινοποίηση στους διαχειριστές. Τα αποτελέσματα μπορούν να εξαχθούν στο Excel ή άλλες εφαρμογές λογιστικών φύλλων. Η μονάδα Webform παρέχει επίσης κάποιες βασικές στατιστικές αναφορές και έχει και μεγάλο API για την επέκταση των δυνατοτήτων της. Μερικά καλά παραδείγματα θα μπορούσαν να είναι διαγωνισμοί, εξατομικευμένες μορφές επικοινωνίας, ή αναφορών. Κάθε ένα από αυτά θα μπορούσε να έχει μια προσαρμοσμένη μορφή για να συμπληρωθεί από τους τελικούς χρήστες.

The screenshot shows the 'Contact' form on the eVideoclub website. The form is titled 'Home Contact' and includes the instruction 'Fill the fields of this form in order to contact us.' and a note '\* means Required'. The form fields are: 'First name', 'Last name', 'E-mail', 'Phone number', 'Subject', and 'Message'. There is a 'Send it' button at the bottom of the form. On the left, there are navigation menus for 'By genre' and 'By studio'. On the right, there is a 'Shopping cart' section showing 'View your shopping cart.' and a 'Get daily eVideoclub news' section with an 'Email' field and a 'Subscribe' button.

**Εικόνα 43: Φόρμα επικοινωνίας του eVideoclub**

## 5 Αποτελέσματα

Τα αποτελέσματα της παρούσας πτυχιακής εργασίας είναι πολλά και ωφέλιμα. Το βασικότερο των αποτελεσμάτων είναι η υλοποίηση ενός λειτουργικού και σύγχρονου ηλεκτρονικού καταστήματος-βάση των τεχνολογιών που εφαρμόζονται σήμερα και είναι ήδη δοκιμασμένες-η οποία προέκυψε μέσω της ανάλυσης των απαιτήσεων, τόσο από πλευράς χρήστη όσο και διαχειριστή, και της σχεδίασης του συστήματος, δηλαδή ποια ενέργεια γίνεται που, τότε και πως.

Ως αποτέλεσμα λαμβάνεται και η εκμάθηση τεχνικών γύρω από το σύστημα διαχείρισης περιεχομένου, όπως η δημιουργία λεξικών και λέξεων κλειδιών μέσω της μονάδας Taxonomy, η δημιουργία κατηγοριών συνδυάζοντας τις μονάδες Views και Taxonomy, η εισαγωγή πεδίων στους κόμβους μέσω της μονάδας CCK, η μορφοποίηση του περιεχομένου μέσω της CSS και του Firebug, η κατασκευή custom μονάδων μέσω της PHP για την κράτηση προϊόντων καθώς και την βαθμολόγηση των προϊόντων μέσω της μονάδας Fivestar.

Επίσης ως αποτελέσματα της παρούσας εργασίας εννοούνται η προσωπική κατάκτηση γνώσεων του συγγραφέα γύρω από τον προγραμματισμό διαδικτύου (PHP,CSS,jQuery,MySQL) και το Drupal (το οποίο χρησιμοποιείται ευρέως στο διαδίκτυο), όπου μελλοντικά μπορούν να αξιοποιηθούν και στην αγορά εργασίας, η πολύτιμη αναζήτηση και εύρεση πληροφοριών που μπορεί να αντλήσει ο αναγνώστης ξεφυλλίζοντας την, αλλά και η απόκτηση ενός ακόμη ενεργού μέλους του συστήματος διαχείρισης περιεχομένου (Drupal) και ευρύτερα της πληροφορικής και της κοινότητας του ανοιχτού κώδικα.

Τέλος αν και το ηλεκτρονικό κατάστημα που υλοποιήθηκε στην εργασία αυτή πραγματεύεται την πώληση ταινιών μπορεί εύκολα να τροποποιηθεί για την πώληση οποιουδήποτε προϊόντος μέσω διαδικτύου. Αυτός είναι και ένας από τους σκοπούς της εργασίας αυτής, να αποτελέσει δηλαδή έναν οδηγό κατασκευής ηλεκτρονικών καταστημάτων.

### 5.1 Συμπεράσματα

Το γενικό συμπέρασμα της παρούσας εργασίας είναι ότι το Drupal, ένα ολοκληρωμένο σύστημα διαχείρισης περιεχομένου, προσφέρει τη δυνατότητα κατασκευής ιστότοπων, τόσο για χρήστες με μηδενική εμπειρία στο χώρο κατασκευής διαδικτυακών ιστότοπων μέσω απλών επιλογών-κουμπιών και πολλαπλές πηγές βοήθειας αλλά τόσο και για αρχάριους στο προγραμματισμό διαδικτύου όσο και για προχωρημένους. Από την παρούσα εργασία τα συμπεράσματα είναι ότι η κατασκευή ενός λειτουργικού ηλεκτρονικού καταστήματος είναι μια επίπονη εργασία που απαιτεί τεχνικές γνώσεις από πλευράς προγραμματισμού αλλά και μορφοποίησης του επιθυμητού αποτελέσματος. Το σύστημα διαχείρισης περιεχομένου διευκολύνει την όλη εργασία που απαιτείται αλλά μέχρι ένα βαθμό, όχι μέχρι το τελικό αποτέλεσμα, από κει και πέρα είναι θέμα τεχνικών δεξιοτήτων. Η παρούσα εργασία προτείνει ένα βασικό τρόπο κατασκευής ιστότοπων, από προγραμματιστικής πλευράς και μορφοποίησης, που στρέφεται γύρω από τα ηλεκτρονικά καταστήματα.

### 5.2 Μελλοντική εργασία και επεκτάσεις

Μελλοντικά θα μπορούσε να προστεθεί στο ηλεκτρονικό κατάστημα η τεχνική SEO (search engine optimization) μια τεχνική που περιγράφει όλες εκείνες τις διαδικασίες-επεμβάσεις που πρέπει να γίνουν στη δομή και το περιεχόμενο μιας ιστοσελίδας ώστε να είναι όσο το δυνατό πιο φιλική στις μηχανές αναζήτησης. Τελικός στόχος είναι η υψηλή κατάταξη του ιστοχώρου και η αύξηση της επισκεψιμότητας της ιστοσελίδας μέσω οργανικών αποτελεσμάτων, δηλαδή αποτελεσμάτων χρηστών του Internet που ψάχνουν στις μηχανές αναζήτησης με τις λέξεις-κλειδιά (key words) που αφορούν το περιεχόμενο της ιστοσελίδας. Επίσης στον ιστότοπο θα μπορούσαν να συμπεριληφθούν, εξειδικευμένη αναζήτηση, επιπλέον προϊόντα, προτάσεις για παρόμοια προϊόντα, προσφορές μεταξύ των χρηστών, μεταφράσεις σε άλλες γλώσσες του περιεχομένου καθώς και αλφαβητική ταξινόμηση ή ταξινόμηση ανά τιμή, προτάσεις οι οποίες θα επέκτειναν ακόμη περισσότερο την λειτουργικότητα του ιστότοπου.

## Βιβλιογραφία

- [1] Ric Shreve's. Drupal 6 Themes. Publisher: BIRMINGHAM - MUMBAI. First published: September 2008
- [2] George Papadogonas, Yiannis Doxaras. Drupal e-commerce with Ubercart 2.x. Publisher: BIRMINGHAM - MUMBAI. First published: March 2010
- [3] Emma Jane Hogbin, Konstantin Kafer. Front End Drupal Designing, Theming, Scripting. Publisher: Prentic Hall. First published: April 2009
- [4] John K. Vandyk, Matt Westgate. Pro Drupal Development. Publisher: Apress.
- [5] Matt Butcher. Drupal 6 JavaScript and jQuery. Publisher: BIRMINGHAM – MUMBAI. First published: February 2009.
- [6] About Drupal: <http://drupal.org/about>
- [7] Understanding Drupal: <http://drupal.org/documentation/understand>
- [8] The Drupal overview: <http://drupal.org/getting-started/before/overview>
- [9] Drupal documentation: <http://drupal.org/documentation>
- [10] Drupal's menu building mechanism: <http://drupal.org/node/10901>
- [11] Drupal's node building mechanism: <http://drupal.org/node/10902>
- [12] Theming Guide: <http://drupal.org/documentation/theme>
- [13] Overview of theme files: <http://drupal.org/node/171194>
- [14] Structure Guide: <http://drupal.org/documentation/structure>
- [15] API reference: <http://api.drupal.org/api/drupal/6>
- [16] Form API Quick start Guide: <http://drupal.org/node/751826>
- [17] Hooks: <http://api.drupal.org/api/drupal/includes!module.inc/group/hooks/6>
- [18] Understanding the hook system for Drupal modules: <http://drupal.org/node/292>
- [19] Node object reference: <http://drupal.org/node/49768>
- [20] Working with JavaScript and jQuery: <http://drupal.org/node/171213>
- [21] W3schools Learn to Create Websites: <http://www.w3schools.com/>
- [22] jQuery: <http://jqueryui.com/> , <http://jquery.com/>
- [23] PHP: <http://en.wikipedia.org/wiki/PHP>
- [24] MySQL: <http://en.wikipedia.org/wiki/MySQL>
- [25] Cascading Style Sheets: [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [26] JavaScript: <http://en.wikipedia.org/wiki/JavaScript>
- [27] jQuery: <http://en.wikipedia.org/wiki/JQuery>
- [28] Drupal: <http://en.wikipedia.org/wiki/Drupal>
- [29] Luke Welling, Laura Thomson. Ανάπτυξη Web Εφαρμογών με PHP και MySQL. Publisher: Γκιούρδας. Published: Φεβρουάριος 2011.
- [30] Drupal: <http://el.wikipedia.org/wiki/Drupal>
- [31] Εγκατάσταση του Drupal από το 0: <http://edutorials.gr/>



## 6 Παράρτημα – Παρουσίαση

### Slide 1:



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**  
**Σχολή Τεχνολογικών Εφαρμογών**  
**Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



**Πτυχιακή Εργασία**  
**Τίτλος:**  
**Ηλεκτρονική Video Λέσχη με δυνατότητα on-line αγορών και ενοικιάσεων ταινιών μέσω ιστοσελίδας (e-shop).**

**ΧΑΡΙΑΛΑΣ ΕΜΜΑΝΟΥΗΛ (ΑΜ: 1365)**

**Επιβλέπων Καθηγητής: ΓΙΩΡΓΟΣ Μ. ΠΑΠΑΔΟΥΡΑΚΗΣ Ph.D.**

### Slide 2:

## Σκοπός της πτυχιακής εργασίας

- **Μελέτη και παρουσίαση του CMS Drupal:** των βασικών χαρακτηριστικών του, του τρόπου λειτουργίας και των δυνατοτήτων του με αποτέλεσμα την
- **Κατασκευή ενός ηλεκτρονικού καταστήματος (e-shop) παροχής υπηρεσιών πώλησης και κράτησης video ταινιών με τα εξής χαρακτηριστικά:**
- **Απλό στη χρήση:** από έμπειρους χρήστες και μη.
- **Λειτουργικό:** να παρέχει γρήγορη πλοήγηση στο περιεχόμενο με εύρηστα και όχι πολύπλοκα μενού.
- **Τεχνολογικά σύγχρονο:** XHTML, CSS, jQuery, MySQL, PHP, Apache.
- **Επεκτάσιμο:** με την κατασκευή και την προσθήκη μονάδων.
- **Ευέλικτο:** να μπορεί να εγκατασταθεί σε οποιαδήποτε πλατφόρμα.

## Slide 3:

## Τί είναι CMS; (Content Management System)

- Στην απλή μορφή του, ένα Σύστημα Διαχείρισης Περιεχομένου είναι μια διαδικτυακή εφαρμογή κατασκευασμένη σε μια γλώσσα προγραμματισμού όπως Java, Python, PHP, Ruby.
- Συνεργάζεται με μια βάση δεδομένων, όπως η MySQL, κάτω από έναν Web Server, όπως ο Apache.
- Αυτοματοποιεί διαδικασίες μαζικής παραγωγής, οργάνωσης και δημοσίευσης περιεχομένου, μέσω ιστοσελίδων, χωρίς την απαίτηση γνώσεων προγραμματισμού.



## Slide 4:

## Τί είναι Drupal;

- Σύστημα διαχείρισης περιεχομένου ανοιχτού κώδικα γραμμένο στην PHP κάτω από την GNU General Public License.
- Λογισμικό που οργανώνει, ενημερώνει, ελέγχει, ρυθμίζει και παρουσιάζει ένα web site.
- Εγκαθίσταστε σε οποιαδήποτε πλατφόρμα (Windows, Linux κ.α.) που μπορεί ταυτόχρονα να υποστηρίξει ένα Web Server που μπορεί να τρέξει την PHP και μια βάση δεδομένων, π.χ. ο Apache.
- Χρησιμοποιείται από χρήστες με μηδενική εμπειρία προγραμματισμού διαδικτύου αλλά και από έμπειρους προγραμματιστές.

Slide 5:

## Βασικά χαρακτηριστικά του Drupal (1)

- **Πεδία-Fields:** η μικρότερη μονάδα πληροφορίας στο Drupal, π.χ. ακέραιος, πραγματικός, κείμενο κ.τ.λ.
- **Κόμβοι-Nodes:** ένα κομμάτι περιεχομένου αποτελούμενο από πεδία ή και από άλλους κόμβους, π.χ. η πρώτη σελίδα.
- **Τύποι περιεχομένου-Content types:** κόμβοι που ανήκουν σε ένα τύπο περιεχομένου με εξ ορισμού ρυθμίσεις για όσους κόμβους ανήκουν σε αυτόν τον τύπο.
- **Ταξινόμια-Taxonomy:** δυνατότητα κατασκευής λεξικών με λέξεις κλειδιά για την ταξινόμια του περιεχομένου.
- **Χρήστες-Users:** κάθε χρήστης έχει ένα User ID. Οι μη εγγεγραμμένοι έχουν ID ίσο με μηδέν ενώ οι εγγεγραμμένοι από το ένα και μετά, το ID ίσο με ένα ανήκει στον διαχειριστή του συστήματος.

Slide 6:

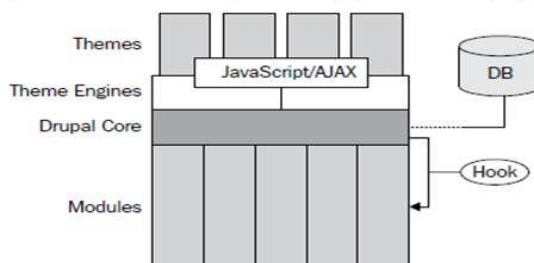
## Βασικά χαρακτηριστικά του Drupal (2)

- **Δικαιώματα-Permissions:** τα δικαιώματα ορίζονται από τον διαχειριστή ανάλογα το είδος χρήστη.
- **Ρόλοι-Roles:** δυνατότητα ομαδοποίησης χρηστών κάτω από ένα ρόλο με συγκεκριμένα δικαιώματα.
- **Μονάδες-Modules:** custom και contributed, λογισμικό που επεκτείνει την λειτουργικότητα και τις δυνατότητες του Drupal, π.χ. η μονάδα views.
- **Αφαιρετικό επίπεδο βάσης δεδομένων-Database abstraction layer:** επιτρέπει την χρήση διαφορετικών βάσεων δεδομένων χρησιμοποιώντας τον ίδιο βασικό κώδικα.
- **Θεματικές παραλλαγές-Themes:** custom και contributed, ορίζουν την εμφάνιση (ή αλλιώς το θέμα) της ιστοσελίδας. Στο κατάλογο του θέματος βρίσκονται όλα τα CSS, PHP και JavaScript αρχεία μέσω των οποίων μπορεί να μορφοποιηθεί η ιστοσελίδα.

## Slide 7:

## Βασικά χαρακτηριστικά του Drupal (3)

- **Πυρήνας-Core:** σύνολο από μονάδες που παρέχουν βασικές λειτουργίες για την υποστήριξη των τμημάτων του συστήματος, όπως φόρτωση βιβλιοθηκών με κοινής χρήσης συναρτήσεις, εκκίνηση του συστήματος, διαχείριση των χρηστών κ.τ.λ.
- **API:** σύνολο από σταθερές και καθολικές μεταβλητές, κλάσεις, αρχεία και συναρτήσεις για την επέκταση και διαχείριση του Drupal σε όλα του τα τμήματα, θεματικές παραλλαγές, βάση δεδομένων, κόμβους κ.τ.λ.
- **Hook system:** μέρος του API (σύνολο συναρτήσεων) μέσω με του οποίου αλληλεπιδρούν οι custom μονάδες με τον πυρήνα.



## Slide 8:

## Εισαγωγή στην εφαρμογή Είδη χρηστών

- **Administrator:** είναι υπεύθυνος για την κατασκευή, τον έλεγχο, την ενημέρωση, την συντήρηση, την επέκταση και τη λειτουργικότητα του ιστότοπου σε κάθε τμήμα του. Έχει καθολικά δικαιώματα.
- **Webmaster:** είναι υπεύθυνος για την δημιουργία, διαγραφή και ενημέρωση του περιεχομένου του ιστότοπου την διαχείριση παραγγελιών και του Forum. Τα δικαιώματα του ορίζονται από τον διαχειριστή. Μπορεί να υπάρχουν περισσότεροι του ενός.
- **Registered user:** μπορεί να πλοηγηθεί στο περιεχόμενο του ιστότοπου και να χρησιμοποιήσει όλες τις παρεχόμενες λειτουργίες του καθώς και την πραγματοποίηση παραγγελιών, τον σχολιασμό στο forum και την βαθμολόγηση προϊόντων.
- **Anonymous user:** μπορεί να πλοηγηθεί στο περιεχόμενο του ιστότοπου και να χρησιμοποιήσει όλες τις παρεχόμενες λειτουργίες του αλλά όχι να πραγματοποιήσει παραγγελίες ή σχολιασμό ή βαθμολόγηση.

Slide 9:

## Λειτουργίες της εφαρμογής

- **Πώληση και κράτηση ταινιών:** μέσω καλαθιού και υποβολή φορμών αγοράς και κράτησης.
- **Κατηγοριοποίηση ταινιών:** ανά είδος δίσκου, δηλαδή DVD ή Blu-ray και μέσω λέξεων κλειδιών, ανά είδος ταινίας, έτος, παραγωγή, ηθοποιό και σκηνοθέτη.
- **Αναζήτηση περιεχομένου:** βασική και σύνθετη.
- **Βαθμολόγηση ταινιών:** ανά χρήστη και συγκεντρωτικά αποτελέσματα ψήφων ανά ταινία
- **Εγγραφή σε Newsletter:** για αποστολή μηνυμάτων ειδοποίησης για νέα προϊόντα.
- **Εγγραφή σε Forum:** για δημιουργία συζητήσεων γύρω από τα προϊόντα.
- **Φόρμα επικοινωνίας :** για αποστολή ερωτήσεων, απαιτήσεων κ.τ.λ.
- **Slideshow:** με τις πιο πρόσφατες ταινίες που ανέβηκαν στο e-shop.

Slide 10:

## Αρχική σελίδα της εφαρμογής

- **Επικεφαλίδα:** περιλαμβάνει τον τίτλο του ιστότοπου, το βασικό μενού μέσω του οποίου μπορεί ο κάθε χρήστης να πλοηγηθεί στις σελίδες που τον ενδιαφέρουν και το πεδίο αναζήτησης.
- **Αριστερή περιοχή:** περιλαμβάνει τρία μπλοκ. Το πρώτο **φιλτράρει** όλες τις ταινίες ανά είδος. Το δεύτερο **μπλοκ φιλτράρει** όλες τις ταινίες ανά εταιρία παραγωγής και το τρίτο **μπλοκ είναι** για την εισαγωγή των χρηστών στο σύστημα.
- **Δεξιά περιοχή:** περιλαμβάνει δύο μπλοκ. Το πρώτο **μπλοκ αποτελεί** το καλάθι αγορών και το δεύτερο **μπλοκ αποτελεί** το newsletter όπου κάθε χρήστης μπορεί να εγγραφεί εισάγοντας το e-mail του.
- **Περιοχή περιεχομένου:** αποτελείται από τρία μέρη, το πρώτο μέρος περιλαμβάνει ένα **slideshow** με όλες τις τελευταίες ταινίες που ανέβηκαν στο site, το δεύτερο μέρος τις ταινίες εκείνες οι οποίες έχουν την μεγαλύτερη βαθμολογία και το τρίτο μέρος όλες τις ταινίες εκείνες οι οποίες έχουν χαμηλή τιμή.

## Slide 11:

## Τελικός κόμβος της εφαρμογής

- **Πληροφορίες:** ο χρήστης μπορεί να αντλήσει πληροφορίες για κάθε ταινία όπως, χαρακτηριστικά Video και Audio, διαθέσιμους υπότιτλους, περιγραφή της ταινίας, trailer, στιγμιότυπα, πρωταγωνιστές, σκηνοθέτη, έτος και εταιρία παραγωγής καθώς και την βαθμολογία της ταινίας.
- **Κατηγοριοποίηση:** ο χρήστης μπορεί να κατηγοριοποιήσει τις ταινίες, μέσω λέξεων κλειδιών, ανά ηθοποιό, σκηνοθέτη, είδος, παραγωγή και έτος και τέλος να βαθμολογήσει την ταινία.
- **Κράτηση:** ο χρήστης μπορεί να κάνει κράτηση μιας ταινίας μέσω της φόρμας κράτησης συμπληρώνοντας τα πεδία όνομα, επώνυμο, e-mail, κωδικό μέλους καθώς και την ημερομηνία που επιθυμεί να γίνει η κράτηση, ξεκινώντας από την τρέχουσα.
- **Αγορά:** στη φόρμα αγοράς ο χρήστης μπορεί να αγοράσει μια ταινία συμπληρώνοντας τα πεδία όπως και στην φόρμα κράτησης συν τα πεδία οδός και αριθμός, περιοχή, T.K., τηλέφωνο και το είδος πληρωμής. Και στις δύο περιπτώσεις, κράτησης και αγοράς, **απαιτείτε** η έγγραφη του χρήστη.

## Slide 12:

## Development Μονάδες της εφαρμογής (1)

- **Content Construction Kit (CCK):** μονάδα για την δημιουργία πεδίων. Τα πεδία: Διάρκεια (τύπου **integer**), Video, Audio, Υπότιτλοι, Περιγραφή (τύπου **textarea**), Image (εξώφυλλο), Screenshots (τύπου **file**) και Movie trailer (URL προς το YouTube-τύπου **Embedded Video**) δημιουργήθηκαν όλα με την μονάδα **CCK**.
- **Ubercart:** μονάδα για την δημιουργία κλάσεων προϊόντων και καλαθιού αγοράς. Τα Πεδία: Sell price (τιμή πώλησης προϊόντος- τύπου **Float**) και SKU δημιουργήθηκαν με την μονάδα **Ubercart**.
- **Taxonomy:** μονάδα για την δημιουργία λεξικών με λέξεις κλειδιά. Τα πεδία **Director, Production, Year, Movie genre** και **Actors** είναι λεξικά και κάθε όρος τους αποτελεί λέξη κλειδί. Τα πεδία: Title, Actors, Movie genre, Director, Production, Year (τύπου **textfield**), Disk type (υποχρεωτικό τύπου **Dropdown**) δημιουργήθηκαν όλα με την μονάδα **Taxonomy**.  
\* Το σύνολο των πεδίων αυτών αποτελούν την βασική δομή για κάθε νέο κόμβο που δημιουργείτε και ανήκει στο βασικό τύπο περιεχομένου **new movie**.

## Slide 13:

## Μονάδες της εφαρμογής (2)

- **Views:** είναι ένας ισχυρός δημιουργός σύνθετων ερωτημάτων προς τη βάση δεδομένων. Μπορεί να φιλτράρει, να ταξινομεί, να μορφοποιεί να συνδέει πεδία και κόμβους, βάση κριτηρίων που ορίζονται από τον χρήστη. Το σύνολο των πεδίων εμφανίζεται είτε σε μια σελίδα είτε σε ένα μπλοκ.
  - Η αρχική σελίδα, το slideshow, τα μπλοκ, οι λίστες με τα DVD και Blu-ray, τους ηθοποιούς, τους σκηνοθέτες, τις παραγωγές, το έτος, τις πρόσφατες ταινίες και την συγκεντρωτική βαθμολογία, είναι όλα **κατασκευασμένα** μέσω των **views**.
- **ImageCache:** παρέχει ρυθμίσεις εικόνων όπως Crop, Resize, Rotate κ.α. δημιουργώντας πρότυπα εικόνας τα οποία μπορούν να εφαρμοστούν σε άλλες εικόνες.
- **Voting-Fivestar:** μονάδα για την καταγραφή ψήφων και του widget Fivestar.
- **Webform:** μονάδα για την δημιουργία φορμών επικοινωνίας.

## Slide 14:

## Μονάδες της εφαρμογής (3)

- Η **custom μονάδα book\_form:** είναι μια μονάδα που χρησιμοποιείτε για την κράτηση ταινιών υπό την μορφή φόρμας.
- **Πεδία:** τα πεδία της φόρμας είναι όνομα, επίθετο, e-mail, κωδικός μέλους και ημερομηνία κράτησης.
- **Σύνολο συναρτήσεων της μονάδας book\_form:**
- **drupal\_get\_form(\$form\_id)** καλεί την συνάρτηση κατασκευής της φόρμας με όρισμα το (ID) όνομα της φόρμας.
- **book\_form\_my\_form(\$form\_state):** συνάρτηση κατασκευής της φόρμας υπό την μορφή πίνακα. Το όρισμα χρησιμοποιείτε για την προσπέλαση των πεδίων της φόρμας.
- **book\_form\_my\_form\_validate(\$form, &\$form\_state):** συνάρτηση επικύρωσης των πεδίων της φόρμας, με όρισματα το τι θέλουμε να επικυρώσουμε και την μεταβλητή για την προσπέλαση των πεδίων της φόρμας.
- **book\_form\_my\_form\_submit(\$form, &\$form\_state):** συνάρτηση υποβολής της φόρμας.
- **db\_query(\$query):** Εκτελεί ένα βασικό ερώτημα στην ενεργό βάση δεδομένων με όρισμα το ερώτημα.
- **form\_set\_error():** συνάρτηση εκτύπωσης μηνύματος λάθους, με όρισμα ένα string.
- **drupal\_set\_message():** συνάρτηση εκτύπωσης μηνύματος με όρισμα ένα string.

## Slide 15:

## Μορφοποίηση της εφαρμογής

- Όλοι οι όροι των λεξικών στο Drupal βρίσκονται στην **μεταβλητή \$terms**, όμως αυτό δεν είναι ευέλικτο όσον αφορά την **μορφοποίηση**, έτσι πρέπει να γραφεί **κώδικας** (PHP,CSS,HTML) για να πάρουμε μεμονωμένα κάθε όρο και να τον μορφοποιήσουμε ανάλογα.
- Το ίδιο ακριβώς συμβαίνει και με το **περιεχόμενο** κάθε κόμβου, βρίσκεται στην **μεταβλητή \$content** και πρέπει πάλι να **προσπελάσουμε** κάθε πεδίο χωριστά ώστε να μορφοποιηθεί κατάλληλα.
- Για την **προσπέλαση** κάθε πεδίου κάθε κόμβου χωριστά, το Drupal προσφέρει την **συνάρτηση dsm()**. Η συνάρτηση αυτή παρέχει όλα τα πεδία του εκάστοτε κόμβου **υπό τη μορφή πίνακα**, με αποτέλεσμα να μπορούν να διαχειριστούν από τον developer (μέσω της PHP) ανάλογα με το τι θέλει να κάνει, μορφοποίηση, αλλαγή τιμής, εμφάνιση-απόκρυψη, ταξινόμηση κ.τ.λ.
- **Σημαντικό** ρόλο στην μορφοποίηση της εφαρμογής έπαιξε το **add-on Firebug**, του ανοιχτού κώδικα φυλλομετρητή, **Firefox** της **Mozilla**. Το add-on αυτό επιτρέπει να γίνονται **αλλαγές on-the-fly** (που δεν αποθηκεύονται) σε έναν ιστότοπο όσον αφορά τις HTML, CSS και JavaScript.

## Slide 16:

## Συμπεράσματα

- Η παρούσα εφαρμογή προσφέρει υπηρεσίες αγοράς και κράτησης και μπορεί να υιοθετηθεί από **οποιοδήποτε** ιστότοπο που έχει ως σκοπό την παροχή τέτοιων υπηρεσιών.
- Ο σχεδιασμός και η κατασκευή ενός διαδικτυακού τόπου είναι μια **πολυσύνθετη διαδικασία** που συνδυάζει **τεχνικές** από διάφορα πεδία, όπως: του προγραμματισμού, των βάσεων δεδομένων, των δικτύων, του ηλεκτρονικού εμπορίου και του marketing, του Search Engine Optimization (SEO), και των πολυμέσων.
- Το Drupal είναι ένα ανοιχτού κώδικα λογισμικό, ασφαλές, σύγχρονο και συνεχώς εξελισσόμενο για την κατασκευή οποιουδήποτε διαδικτυακού τόπου με μια μεγάλη κοινότητα υποστήριξης για κάθε χρήστη. Οι λειτουργίες του μπορούν να επεκταθούν και να τροποποιηθούν με οποιονδήποτε τρόπο και χωρίς **κανένα κόστος**.
- Στο Drupal μπορεί ο οποιοσδήποτε να κατασκευάσει ένα ιστότοπο, αλλά για να βρίσκεται σε ανταγωνιστικό επίπεδο σε σχέση με τους ήδη υπάρχοντες, απαιτείται πολύ καλό **προγραμματιστικό υπόβαθρο** στις τεχνολογίες διαδικτύου.