



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΚΑΛΑΜΑΤΑΣ - ΠΑΡΑΡΤΗΜΑ ΣΠΑΡΤΗΣ  
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

*ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ*

**" Ανάπτυξη επεκτάσεων προστιθέμενης αξίας (value-added) για το περιβάλλον διαχείρισης περιεχομένου Joomla. Εφαρμογή στην υλοποίηση ιστότοπου για ηλεκτρονικά αθλήματα."**

**ΦΟΙΤΗΤΗΣ: ΦΛΕΣΙΑΣ ΒΑΣΙΛΕΙΟΣ - Α.Μ:2007227**

**ΕΠΙΒΛΕΠΩΝ: ΚΑΡΑΜΠΑΤΣΟΣ ΒΑΣΙΛΕΙΟΣ**

**ΑΥΓΟΥΣΤΟΣ 2013**

## ΠΕΡΙΕΧΟΜΕΝΑ

### Contents

1. ΕΙΣΑΓΩΓΗ.....	5
1.1. Σκοπός της εργασίας .....	5
1.2. Σκοπός του είδους των ελεγκτάσεων .....	6
2. ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ .....	7
2.1. Joomla, Σύστημα Διαχείρισης Περιεχομένου .....	7
2.2. Το Model-View-Controller ( Mvc ).....	9
2.2.1. Εφαρμογές.....	10
2.2.2. Πλεονεκτήματα MVC .....	10
2.2.3. Παραλλαγές του μοντέλου .....	11
2.2.4. Mvc Στο Joomla .....	11
2.3. Επεκτασεις Joomla .....	14
2.3.1. Component .....	14
2.3.2. Module.....	16
2.3.3. Plugin.....	17
2.4. Joomla και SEO.....	19
2.4.1. Βελτιστοποίηση για τις μηχανές αναζήτησης (SEO) .....	19
2.4.2. Λειτουργία μηχανών αναζήτησης .....	20
2.4.3. Κατάταξη μιας ιστοσελίδας.....	21
2.4.4. Τεχνικές «White hat» .....	23
2.4.5. Τεχνικές «Black hat».....	24
2.5. Ασφαλεια.....	25
2.5.1. Η εφαρμογή της ασφάλειας στον ιστότοπο .....	25
2.5.2. Ασφάλεια στο Joomla με χρήση άλλων τεχνολογιών .....	25
2.5.3. Ασφάλεια στο Joomla με χρήση του Joomla.....	28
2.5.4. Αναπτύσσοντας μία επέκταση χωρίς κενά ασφαλείας.....	31

2.5.5.	SQL injections (έγχυση SQL) .....	32
2.5.6.	Η διαδικασία του SQL injection.....	33
2.5.7.	Η διαδικασία που ακολουθείται όταν η ασφάλεια έχει παραβιαστεί .....	33
2.6.	PHP.....	35
2.6.1.	Πλεονεκτήματα της PHP.....	35
2.7.	MySQL.....	38
2.7.1.	Πλεονεκτήματα της MySQL.....	38
2.8.	CSS (Cascading Style Sheets).....	40
2.8.1.	Κανόνες CSS.....	40
2.8.2.	Εισαγωγή CSS στη σελίδα.....	40
2.8.3.	Προτεραιότητα .....	41
2.8.4.	Κληρονομικότητα.....	42
3.	ΗΛΕΚΤΡΟΝΙΚΑ ΑΘΛΗΜΑΤΑ .....	44
3.1.	Ιστορία ηλεκτρονικών αθλημάτων.....	44
3.2.	Το παρόν των ηλεκτρονικών αθλημάτων.....	45
3.3.	E-sports στην Ελλάδα.....	45
3.4.	Είδος παιχνιδιού MOBA .....	46
3.4.1.	Ο σκοπός στο είδος παιχνιδιού MOBA.....	47
3.4.2.	Χαρακτηριστικά του Smite .....	48
4.	ΥΛΟΠΟΙΗΣΗ .....	50
4.1.	Component .....	50
4.1.1.	Front-End.....	51
4.1.2.	Back-End .....	54
4.1.3.	Toolbars.....	57
4.1.4.	Pagination .....	60
4.2.	Η χρήση του CSS .....	64
4.2.1.	Εφέ λάμπης γύρω από την φωτογραφία.....	64

4.2.2.	Κρυμμένο κείμενο .....	65
4.2.3.	Διάφανο φόντο για το κείμενο πάνω από την εικόνα.....	65
4.2.4.	Χρήση CSS αντί για Javascript .....	66
4.2.5.	Χρήση Javascript.....	66
4.3.	Module.....	67
4.3.1.	Έξοδος του module.....	67
4.3.2.	Back-End του module.....	67
4.3.3.	Παρουσίαση Module, pshow .....	68
4.3.4.	Ανάλυση κώδικα module, pshow. ....	68
4.4.	Plugin Pshow .....	69
4.4.1.	Ανάλυση κώδικα Plugin Pshow .....	69
4.4.2.	Παράμετροι Plugin Pshow .....	70
4.5.	Plugin Pshow Information.....	71
4.5.1.	Ανάλυση κώδικα plugin Pshow Information.....	73
4.5.2.	Παράμετροι plugin Pshow Information.....	73
4.6.	Η εφαρμογή της ασφάλειας στον ιστότοπο.....	74
4.6.1.	Ασφάλεια στην δημιουργία επέκτασης του Joomla .....	74
4.7.	SEO – Η ανάπτυξη Search Engine Friendly (SEF) συνδέσμων.....	77
4.7.1.	Χτίζοντας τα κομμάτια του URL .....	77
4.7.2.	Κώδικας ανάπτυξης SEF συνδέσμων .....	78
4.7.3.	Ανάλυση του κώδικα ανάπτυξης SEF συνδέσμων.....	79
4.7.4.	Αποκωδικοποιώντας τα URL .....	80
5.	ΕΠΙΛΟΓΟΣ.....	82
5.1.	Συμπεράσματα .....	82
5.2.	Μελλοντικές Επεκτασεις.....	83
	Βιβλιογραφία.....	84

## *Ευχαριστίες*

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω την κοπέλα μου Κατερίνα για όλη της την υποστήριξη και βοήθεια. Τον κ. Τριτάκη που παρόλο που σταμάτησε να είναι υπεύθυνος συνέχισε να με βοηθάει και να με καθοδηγεί. Τους συμφοιτητές Κωνσταντίνο Παύλου, Σωτήρη Τσάβο και Χρήστο Ψαρρα για τις χρήσιμες συμβουλές τους. Την οικογένεια μου που με στήριξε στις σπουδές μου με διάφορους τρόπους όλα αυτά τα χρόνια. Και τέλος τον κ. Καραμπάτσο που ανέλαβε την πτυχιακή μου ενώ βρισκόταν σε κρίσιμο σημείο.

## 1. ΕΙΣΑΓΩΓΗ

Η πρώτη επιλογή η οποία χρειάστηκε να γίνει σχετικά με το θέμα της πτυχιακής ήταν πάνω σε ποιο τομέα της πληροφορικής θα βασιζόταν. Η απάντηση στο παραπάνω ερώτημα, ήταν το ίντερνετ. Ο κύριος λόγος που οδήγησε σε αυτή την απάντηση, ήταν η ευκολία διάδοσης της πληροφορίας στο μεγαλύτερο δυνατό κοινό χρησιμοποιώντας τα μέσα του διαδικτύου. Μία ιστοσελίδα είναι ένα πραγματικός έλεγχος για την πληροφορία την οποία περιέχει, καθώς και τον τρόπο παρουσίασης αυτής. Η ανάπτυξη ιστοσελίδων δημιουργεί μία σειρά από μοναδικές προκλήσεις. Αν παρομοιαστεί μία ηλεκτρονική σελίδα με ένα μαγαζί ο πελάτης είναι ένα κλικ κοντά αλλά ταυτόχρονα και ένα κλικ μακριά από τις άλλες σελίδες. Λαμβάνοντας υπόψη το παγκόσμιο χαρακτήρα του ίντερνετ αντιλαμβάνεται κανείς τον ανταγωνισμό που υπάρχει.

Στην συνέχεια, έπρεπε να αποφασιστεί τι θα αναπτυχθεί σχετικά με το διαδίκτυο. Λαμβάνοντας υπόψη τις γνώσεις που αποκτήθηκαν μέσω των σπουδών σχετικά με τον προγραμματισμό διαδικτύου γίνονταν αναζήτηση για κάτι που να περιλαμβάνει αλλά και να επεκτείνει τις γνώσεις αυτές. Γρήγορα υπήρξε το συμπέρασμα ότι τα Συστήματα Διαχείρισης Περιεχομένου (ΣΔΜ) ήταν η κατάλληλη λύση στον προβληματισμό αφού περιελάμβαναν όλα τα παραπάνω. Όμως, προκειμένου να επεκταθούν οι γνώσεις αυτές, ήταν αναγκαίο να επεκταθούν τα ΣΔΜ.

Τέλος, η τελευταία επιλογή ήταν σχετικά με την θεματολογία της σελίδας. Αφού μία ιστοσελίδα μπορεί να είναι σχετική με τα πάντα επιλέχτηκε κάτι που να κεντρίζει το ενδιαφέρον και να είναι ευχάριστο. Το ενδιαφέρον αυτό μεταφράζεται σε ηλεκτρονικά παιχνίδια και πιο συγκεκριμένα σχετικά με ένα παιχνίδι ανταγωνιστικής φύσεως, και με περιεχόμενα γύρω από αυτό.

### 1.1. Σκοπός της εργασίας

Βασικό κριτήριο για την επιλογή του θέματος της πτυχιακής ήταν η χρήση των προηγούμενων αποκτηθέντων γνώσεων. Ωστόσο, σημαντική ήταν επίσης η απόκτηση νέων γνώσεων. Η εφαρμογή των γνώσεων που υπήρχαν ήδη (PHP, MySQL, HTML, CSS) αλλά και η ανάπτυξη καινούργιων γνώσεων (CMS, MVC, Extentions, Joomla).

Σημαντικό ήταν να δημιουργηθεί κάτι ολοκληρωμένο με δυνατότητες επεκτάσεις, προκειμένου να υπάρξει εξοικείωση με ιστότοπους μεγαλύτερου μεγέθους και είδους. Η σφαιρική γνώση του τρόπου λειτουργίας των νέων τεχνολογιών ήταν απαραίτητη.



Σημαντικό κριτήριο ήταν επίσης να δημιουργηθεί κάτι που θα ήταν ενδιαφέρον και χρήσιμο.

#### 1.2. Σκοπός του είδους των επεκτάσεων

Ο σκοπός της πτυχιακής εργασίας ήταν να δημιουργηθεί ένας ιστότοπος με την θεματολογία ενός παιχνιδιού. Ωστόσο, σκοπός ήταν επίσης η δημιουργία επεκτάσεων που να έχουν εφαρμογές σε περισσότερο από έναν ιστότοπους. Έτσι, οι επεκτάσεις που δημιουργήθηκαν προσπάθησαν να συνδυάσουν τα παραπάνω χαρακτηριστικά.

Περιέχουν την θεματολογία ενός παιχνιδιού, υλικό ενδιαφέρον για το παιχνίδι, ενώ παράλληλα θα μπορούσαν να έχουν εφαρμογή σε οποιοδήποτε ιστότοπο. Προκειμένου να μπορέσουν να έχουν εφαρμογή σε διαφορετικούς ιστότοπους έγινε προσπάθεια ο κώδικας να επιτρέπει την δυνατότητα εφαρμογής σε οποιοδήποτε σύστημα.

## 2. ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

### 2.1. Joomla, Σύστημα Διαχείρισης Περιεχομένου

Το Joomla [1] [2] είναι ένα δημοφιλές Σύστημα Διαχείρισης Περιεχομένου ανοικτού κώδικα. Η χρήση του συνήθως είναι η δημοσίευση περιεχομένου στον παγκόσμιο ιστό. Είναι γραμμένο σε PHP. Χαρακτηρίζεται από τις τεράστιες δυνατότητες του οι οποίες γίνονται απεριόριστες με την χρήση επεκτάσεων. Παρόλο των δυνατοτήτων του είναι ένα απλό αλλά και ευέλικτο περιβάλλον εργασίας ενώ παράλληλα έχει βάθος στην ανάπτυξη του.

Μερικές από τις δυνατότητες του συγκεκριμένου Συστήματος Διαχείρισης Περιεχομένου είναι:

- Νέα, προϊόντα ή υπηρεσίες τμήματα και ενότητες πλήρως επεξεργάσιμα και εύχρηστα.
- Πλήρως παραμετροποιήσιμο περιεχόμενο και περιβάλλον, συμπεριλαμβανομένων των θέσεων του αριστερού, κεντρικού και δεξιού μενού.
- Χρήση του browser για προσθήκη ή αλλαγή του περιεχομένου του ιστότοπου σε οποιοδήποτε σημείο του ιστότοπου.
- Τρέχει σε Linux, FreeBSD, Macos server, Solaris και AIX.
- Πλήρης μηχανισμός διαχείρισης της βάσης δεδομένων του ιστότοπου.
- Archive manager. Η τοποθέτηση των παλαιών άρθρων στην "κατάψυξη" αντί για την διαγραφή αυτών.
- Δυνατότητα κατασκευής πολυγλωσσικής ιστοσελίδας (*Multilanguage*).
- Δυνατότητα βελτιστοποίησης της ιστοσελίδας Joomla στις μηχανές αναζήτησης (SEO) διότι το Joomla είναι ένα Search Engine Friendly CMS (SEF).

Επίσης το Joomla είναι δωρεάν. Μπορεί να χρησιμοποιηθεί, να τροποποιηθεί και να αναπτυχθεί χωρίς να πρέπει να πληρωθεί κάποια άδεια χρήσης σε κανέναν. Το Joomla είναι ελεύθερο λογισμικό με γραμμένο κώδικα υπό την GPL Licence. Η μόνη υποχρέωση, είναι να μην αφαιρεθούν ή αλλοιωθούν τα λογότυπα, ή να οικειοποιηθεί η εφαρμογή. Το Joomla Κέρδισε το Packet Publishing Open Source Content Management System βραβείο το 2006, 2007 και 2011.

Κάποιες από τις εταιρείες και τους οργανισμούς που χρησιμοποιούν το Joomla είναι:





**ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ**  
Υπουργείο Παιδείας και Θρησκευμάτων,  
Πολιτισμού και Αθλητισμού



MINISTRY OF DEFENCE

**SOCA**

SERIOUS ORGANISED CRIME AGENCY



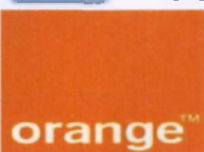
High Court of Australia



Groupama



**MAERSK**  
TRAINING



Ελληνική  
ΥΠΕΠΘ

Δημοκρατία

<http://www.minedu.gov.gr>

MTV

<http://www.mtvgreece.gr>

Υπουργείο Άμυνας Ηνωμένου  
Βασιλείου

<http://www.stabilisationunit.gov.uk>

Κυβέρνηση Ηνωμένου  
Βασιλείου Υπηρεσία σοβαρού  
οργανωμένου εγκλήματος Η.Β

<http://www.soca.gov.uk>

<http://www.hcourt.gov.au>

Ανώτατο  
Αυστραλίας

Δικαστήριο

<http://www.eurunion.org>

European Union delegation to  
the USA

University of Alabama at  
Birmingham

<http://uab.edu>

IKEA

<http://www.ikea.com.kw> <http://www.ikea.com.sa/ar>

General Electric

<http://www.getransportation.com>

GROUPAMA

<http://www.groupama-transport.com>

<http://www.groupamarace.nc>

MAERSK

<http://maersktraining.com>

Orange Telecom

<http://api.orange.com>

Pizza Hut

<http://www.PizzaHut.fr>

## 2.2. Το Model-View-Controller (Mvc)

Το mvc είναι ένα λογισμικό αρχιτεκτονικής - η δομή του συστήματος - το οποίο διαχωρίζει τομέα / εφαρμογή / εργασία (ότι προτιμάς) λογικά από την υπόλοιπη διεπαφή του χρήστη. Αυτό συμβαίνει από το διαχωρισμό των εφαρμογών σε 3 μέρη: μοντέλο (model), αναπαράσταση (view), και διαχείριση (controller).

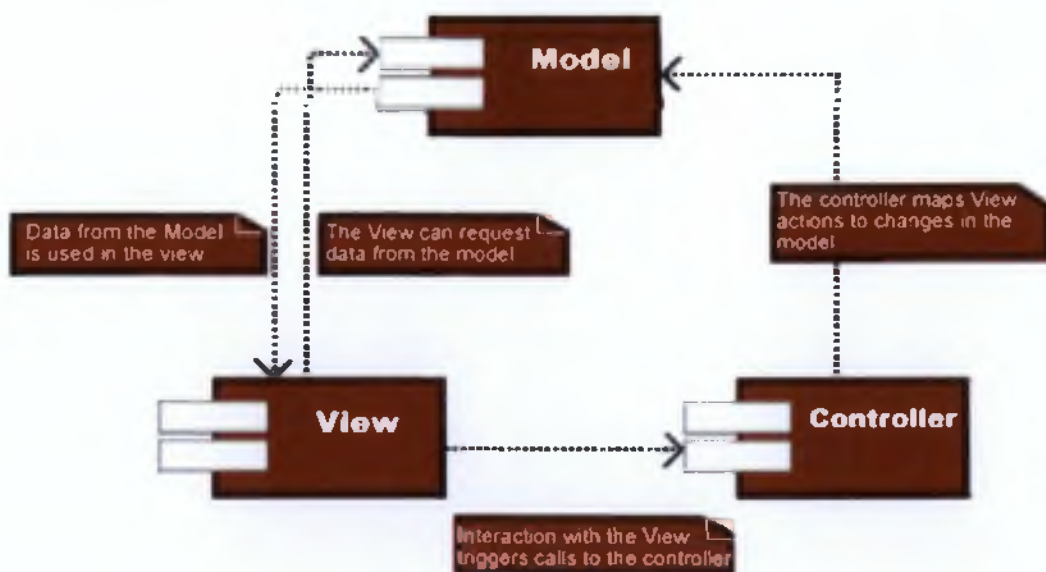
Το μοντέλο (model) διαχειρίζεται βασικές συμπεριφορές και δεδομένα της εφαρμογής. Α ανταποκρίνεται σε αιτήματα για πληροφορίες ακόμη και για να ειδοποιεί τους παρατηρητές όταν οι πληροφορίες αλλάζουν. Αυτό θα μπορούσε να είναι μια βάση δεδομένων ή κάποιος αριθμός από τα δεδομένα των δομών ή αποθηκευτικά συστήματα. Εν συντομία, είναι τα δεδομένα και η διαχείριση των δεδομένων της εφαρμογής.

Η αναπαράσταση (view) αποτελεσματικά παρέχει το στοιχείο στη διεπαφή του χρήστη της εφαρμογής. Θα υπάρχουν δεδομένα από το model με μορφή που ταιριάζει για τη διεπαφή του χρήστη

Η διαχείριση (controller) λαμβάνει είσοδο χρήστη και πραγματοποιεί κλήσεις σε αντικείμενα των Model και των view για να προσαρμόσει κατάλληλα τις δράσεις.

Εν κατακλείδι, αυτά τα 3 στοιχεία δουλεύουν μαζί για να δημιουργήσουν τα 3 βασικά συστατικά του MVC.

*Μία συνολική εικόνα για το MVC.*



### 2.2.1. Εφαρμογές

Το Pattern MVC χρησιμοποιείται όταν ένα συστατικό έχει κάποιο από τα ακόλουθα χαρακτηριστικά:

- Είναι πιθανό να δει το συστατικό με διαφορετικούς τρόπους. Η εσωτερική αναπαράσταση του αντικειμένου μπορεί να είναι εντελώς διαφορετική από την εξωτερική του (στην οθόνη).
- Η συμπεριφορά ή η αναπαράσταση του συστατικού μπορούν να αλλάξουν δυναμικά.
- Υπάρχουν διαφορετικοί τύποι συμπεριφορών, με την έννοια ότι πολλών ειδών συμπεριφορές μπορούν να υλοποιηθούν για το ίδιο αντικείμενο.
- Πολλές φορές είναι επιθυμητό να επαναχρησιμοποιηθεί ένα τέτοιο συστατικό κάτω από άλλες συνθήκες, με τις λιγότερες δυνατές αλλαγές.

### 2.2.2. Πλεονεκτήματα MVC

Διαχωρισμός Προβλημάτων (Separation of Concerns): Αυτό είναι και το πιο βασικό πλεονέκτημα του MVC. Ουσιαστικά δημιουργείται μία εφαρμογή η οποία έχει τρία επίπεδα, το επίπεδο των models , το επίπεδο των controllers και το επίπεδο των views -που θα αναλυθούν παρακάτω- και το κάθε επίπεδο επιτελεί ξεχωριστό έργο και ταυτόχρονα συνεργάζεται με τα άλλα επίπεδα. Μία σωστή MVC εφαρμογή είναι εκείνη που τα τρία επίπεδα είναι ξεκάθαρα καθορισμένα και δεν συμπλέκονται.

Επεκτασιμότητα: Το δεύτερο πλεονέκτημα της MVC αρχιτεκτονικής είναι πολύ σημαντικό επίσης. “Επεκτασιμότητα” είναι η δυνατότητα που διαθέτει μία εφαρμογή , κατά την οποία μπορεί μελλοντικά να προστεθούν λειτουργίες σε αυτή. Μπορούν ακόμα να αλλάξουν κάποιες από τις ήδη υπάρχουσες λειτουργίες και να υπάρξουν διαφορετικά αποτελέσματα. Ένα από παράδειγμα είναι, ότι η πλατφόρμα Joomla είναι επεκτάσιμη με τη χρήση των διάφορων επεκτάσεων διότι προσθέτονται στις ήδη υπάρχουσες λειτουργίες και άλλες λειτουργίες. Τα προγράμματα που είναι φτιαγμένα με MVC αρχιτεκτονική έχουν βασικό χαρακτηριστικό ότι είναι επεκτάσιμα.

Ελεγχιμότητα (Testability): Αυτό είναι ένα πολύ κρίσιμο χαρακτηριστικό. Οι MVC εφαρμογές έχουν την δυνατότητα να είναι ελέγξιμες και με τον τρόπο αυτό συντηρούνται πιο εύκολα.

“Καθαρά” URLs.: Τα περισσότερα MVC frameworks για web applications δίνουν τη δυνατότητα να έχουμε “καθαρά” urls. Συνεπώς, με το MVC δημιουργούνται πιο όμορφα και εύληπτα από το χρήστη αλλά και την μηχανή αναζήτησης URLs.

### 2.2.3. Παραλλαγές του μοντέλου

- **Multiple view targets** – Ένα Model μπορεί να παρέχει πληροφορίες σε περισσότερα από ένα View. Αυτό συνήθως είναι χρήσιμο σε GUI υλοποιήσεις.
- **“Look but don’t touch”** – Τα View δεν χρειάζονται πάντοτε κάποιον Controller. Μπορούν να χρησιμοποιηθούν μονάχα για οπτικές αναπαραστάσεις των πληροφοριών του μοντέλου χωρίς να υποστηρίζουν αλλαγές στις πληροφορίες αυτές.
- **Model push vs view pull** – Το MVC μπορεί να υλοποιηθεί κατά δύο τρόπους. Το Model μπορεί να στέλνει updates στο view (ή στα views) ή ένα view μπορεί να λαμβάνει πληροφορίες όποτε χρειάζεται από το model. Αυτή η επιλογή την υλοποίηση της σχέσης στο σύστημα.

### 2.2.4. Mvc Στο Joomla

Το Joomla κάνει εκτενώς χρήση της αρχιτεκτονικής Model-View-Controller (MVC).

Το MVC στο Joomla [1] [2] [3] [4] στην ουσία αποτελεί ένα μοτίβο προκειμένου να μοιράζεται ο κώδικας σε μικρότερα κομμάτια κάνοντας την επέκταση αλλά και συντήρηση του κώδικα πιο εύκολη. Ο διαχωρισμός της λογικής με την παρουσίαση δεδομένων είναι βασικός για την εφαρμογή του. Ωστόσο, υπάρχουν αρκετές παραλλαγές του MVC που εκτελούνται κατά περίπτωση ανάλογα με τις ανάγκες της κάθε περίπτωσης.

Όταν το Joomla ξεκινάει να επεξεργάζεται ένα αίτημα από τον χρήστη, το πρώτο πράγμα που κάνει είναι να αναλύσει το URL προκειμένου να καθορίσει πιο component θα είναι υπεύθυνο για την επεξεργασία του αιτήματος και να παραδώσει τον έλεγχο σε αυτό το component.

Εάν το component έχει σχεδιαστεί κάνοντας χρήση της αρχιτεκτονικής MVC θα περάσει τον έλεγχο στο controller. Στην συνέχεια, ο controller είναι υπεύθυνος



προκειμένου να αναλύσει το αίτημα. Πρέπει να καθορίσει ποιο μοντέλο θα χρειαστεί προκειμένου να ικανοποιήσει το αίτημα που δέχτηκε, αλλά και ποιο view προκειμένου να εμφανίσει τα αποτελέσματα στον χρήστη.

Το μοντέλο περιλαμβάνει τα δεδομένα που θα χρησιμοποιηθούν από το component. Στις περισσότερες περιπτώσεις τα δεδομένα θα προέλθουν από μία βάση δεδομένων. Συνήθως, θα είναι μία Joomla βάση δεδομένων ή κάποια εξωτερική βάση δεδομένων. Το model είναι υπεύθυνο προκειμένου να ενημερώνει την βάση δεδομένων όπου είναι απαραίτητο. Ο σκοπός του model είναι να απομονώσει το controller και view από τις λεπτομέρειες σχετικά με το πώς αποκτούνται ή τροποποιήθηκαν τα δεδομένα.

Το view είναι υπεύθυνο για την δημιουργία της εξόδου που στέλνεται στον φυλλομετρητή από το component. Χρησιμοποιεί το model για κάθε πληροφορία που χρειάζεται και το μορφοποιεί κατάλληλα. Για παράδειγμα, μία λίστα δεδομένων όπου προέρχονται από το model θα μπορούσε να εμφανιστεί σε ένα πίνακα από το view.

Μιας και το Joomla είναι σχεδιασμένο για να αποτελείται από πολλά κομμάτια, η έξοδος του component που θα δει ο χρήστης είναι μόνο ένα κομμάτι από την ολοκληρωμένη σελίδα. Όταν το view έχει δημιουργήσει την έξοδο το component παραδίδει τον έλεγχο πίσω στο Joomla framework προκειμένου να φορτώσει και εκτελέσει το template. Το template συνδυάζει την έξοδο από το component, αλλά και όποιου άλλου module είναι ενεργοποιημένου στην τρέχουσα σελίδα, έτσι ώστε να μπορεί να παραδοθεί στον φυλλομετρητή σαν μία ενιαία σελίδα.

Προκειμένου να προσφέρει επιπλέον δύναμη και ευελιξία στο σχεδιαστή ιστοσελίδων, του οποίου η μόνη έννοια είναι η δημιουργία νέου κώδικα, παρά τον χειρισμό του υποκείμενου κώδικα το Joomla χωρίζει το παραδοσιακό view σε ένα ξεχωριστό view και layout. Το view αποκτάει δεδομένα από το model, όπως σε ένα παραδοσιακό MVC πρότυπο, αλλά τότε απλά κάνει διαθέσιμα τα δεδομένα στο layout, το οποίο είναι υπεύθυνο για την μορφοποίηση των δεδομένων που θα εμφανιστούν στον χρήστη.

Το πλεονέκτημα αυτού του διαχωρισμού είναι ότι το Joomla template system προσφέρει ένα απλό μηχανισμό για το layout ώστε να παρακάμψει το template (γνωστά ως template overrides). Το template override συνδυάζεται με το template και προσφέρουν στον σχεδιαστή ιστοσελίδων ολοκληρωτικό έλεγχο για την έξοδο του

Joomla που προέρχεται από τον πυρήνα του αλλά και της οποιαδήποτε επέκτασης που συμμορφώνεται με το MVC πρότυπο.

Η εφαρμογή του Joomla MVC γίνεται χάρη στις τρεις εξής κλάσεις Model, View και Controller.



## 2.3. Επεκτασεις Joomla

### 2.3.1. Component

Από όλες τις διαθέσιμες επεκτάσεις το component ή στα ελληνικά διαχείριση είναι το πιο ουσιώδες για το Joomla. Όπως φαίνεται από το όνομα του είναι κεντρική λειτουργική μονάδα του Joomla και μπορεί να περιγραφεί σαν μια μικρή εφαρμογή. Μία εύκολη αναλογία θα ήταν το CMS Joomla ως το λειτουργικό σύστημα, ενώ το component είναι η εφαρμογή. Χρησιμοποιείται για τη δημιουργία, διαχείριση και κατηγοριοποίηση άρθρων.

Χαρακτηριστικό είναι ότι κάθε φορά που το Joomla εκτελείται, ένα component πάντα «καλείται». Συνήθως το Component βρίσκεται συνήθως στο κεντρικό κομμάτι του περιεχόμενου ενός template (βέβαια, αυτό εξαρτάται και από το template που χρησιμοποιείται). Λόγω του βασικού του χαρακτήρα σαν επέκταση, το component είναι γενικά και το πιο σύνθετο. Μπορεί να φορτωθεί μονάχα ένα component σε μια ιστοσελίδα.

Τα περισσότερα component έχουν δύο κύρια κομμάτια, το κομμάτι της διαχείρισης «Back-end» και το μπροστινό κομμάτι της σελίδας «Front-end» που εμφανίζεται το περιεχόμενο αυτού. Το μπροστινό κομμάτι της σελίδας ενός component είναι αυτό που χρησιμοποιείται για να εμφανιστούν οι σελίδες στις οποίες εκτελείται το component, προκειμένου να εμφανιστεί όπως αυτό έχει οριστεί από το κομμάτι της διαχείρισης. Το κομμάτι της διαχείρισης ενός component προσφέρει μία πληθώρα από επιλογές σχετικά με την τροποποίηση χρησιμοποιώντας την εφαρμογή διαχείρισης του Joomla. Με μερικά μόνο κλικ επιτρέπεται η παραμετροποίηση και η διαχείριση μίας πληθώρας επιλογών που επηρεάζουν διαφορετικές πτυχές της επέκτασης.

Επιπλέον, μπορούν να χρησιμοποιηθούν και εφαρμογές τρίτων (3rd party Joomla components) που αναπτύσσονται από τα μέλη της Joomla Κοινότητας. Κάποια διατίθενται δωρεάν και κάποια άλλα από αυτά πωλούνται. Πολλές φορές μάλιστα, υπάρχουν επεκτάσεις όπου προσφέρουν μία δωρεάν έκδοση με περιορισμένες δυνατότητες προκειμένου να προωθήσουν την επί πληρωμή έκδοση που επίσης προσφέρουν.

Συνήθως χρησιμοποιούνται για να δημιουργούν και να αναβαθμίζουν τιμές στις βάσεις δεδομένων. Πραγματικά έχει τις δυνατότητες να κάνει τα πάντα, αρκεί να είναι προγραμματισμένο στην PHP. Μπορεί να χρησιμοποιηθεί και για το «ανέβασμα» φωτογραφιών ή βίντεο.

Τέλος, ακόμα και το ίδιο το Joomla χρησιμοποιεί τα components στον πυρήνα του με τα εξής χαρακτηριστικά παραδείγματα: content management system, contact forms και Web Links.

### 2.3.2. *Module*

Το *module* είναι μία ακόμα μορφή επέκτασης του Joomla. Χρησιμοποιείται συχνά για να εμφανίσει ένα μικρό κομμάτι περιεχομένου, συνήθως στα άκρα μίας σελίδας (πάνω, κάτω, αριστερά, δεξιά). Συνήθως, είναι στοιχεία ενός καταλόγου περιεχομένου ή μίας πλαϊνής μπάρας. Αντίθετα με το *component*, οποιοσδήποτε αριθμός *module* μπορεί να εμφανιστεί σε μία σελίδα. Δεν προορίζονται για να είναι το κύριο περιεχόμενο μίας σελίδας, αλλά για να το συμπληρώνουν. Το Joomla υποστηρίζει επίσης την ύπαρξη *module* περιεχομένου, χωρίς να περιλαμβάνεται καθόλου προγραμματισμός και μπορεί να είναι απλή στατική HTML ή ακόμα και κείμενο. Παρόμοια με το *component* υπάρχουν *module* στον πυρήνα του Joomla. Γνωστότερο παράδειγμα ίσως να είναι το *login module* που επιτρέπει την είσοδο των χρηστών στον λογαριασμό τους σε μία σελίδα που κάνει χρήση της πλατφόρμας Joomla.

Τα «κουτιά» αυτά όπως συνηθίζεται να είναι η μορφή τους μπορούν να «δεθούν» σε ένα *component*. Όπως για παράδειγμα το «latest news» *module* του πυρήνα συνδέεται με το *component com\_content* του πυρήνα προκειμένου να εμφανίσει υπερσυνδέσμους στα νεότερα αντικείμενα περιεχομένου. Επίσης το *module* μπορεί να τοποθετηθεί σε μία θέση για *module* ενός *template*. Με αυτό τον τρόπο το *module* «εξηγεί» στο *template* που να τοποθετηθεί. Στη συνέχεια το *template* «εξηγεί» στο Joomla που να τοποθετήσει το *module* ανάλογα με το που έχει οριστεί η συγκεκριμένη θέση του *module* στο *template* αυτό. Για παράδειγμα εάν είχε οριστεί η θέση ενός *module* στα αριστερά ενός *template* αυτό δεν σημαίνει απαραίτητα ότι το *module* θα εμφανιστεί στην αριστερή πλευρά της σελίδας. Αλλά, όπου αυτό είναι ορισμένο βάση του *template*.

Τα *module* διαχειρίζονται από το κομμάτι διαχείρισης του Joomla και πιο συγκεκριμένα από το *Module Manager*. Εκεί μπορούν να βρεθούν οι παράμετροι. Αλλάζοντας τις παραμέτρους επιτυγχάνεται ο έλεγχος του περιεχομένου της εξόδου ή τον τρόπο που εμφανίζεται η έξοδος.

### 2.3.3. *Plugin*

Το plugin (παλαιότερα γνωστό ως Mambot) είναι άλλη μία μορφή επέκτασης του Joomla. Τα plugin προσφέρουν λειτουργίες οι οποίες είναι συσχετισμένες με την ενεργοποίηση γεγονότων. Ο πυρήνας του Joomla παρέχει μία πληθώρα από γεγονότα ενεργοποίησης, ωστόσο οποιαδήποτε επέκταση μπορεί να προκαλέσει την ενεργοποίηση ενός προσαρμοσμένου γεγονότος. Όταν ένα συγκεκριμένο γεγονός συμβαίνει, όλες οι λειτουργίες των plugin που είναι συνδεδεμένες με αυτό το γεγονός εκτελούνται διαδοχικά. Χάρη σε αυτό, τα plugin επεκτείνουν με έναν ιδιαίτερα ισχυρό τρόπο την λειτουργικότητα του Joomla πλαισίου (Framework). Επίσης προσφέρει στους προγραμματιστές επεκτάσεων ένα τρόπο να επιτρέψουν σε άλλες επεκτάσεις να ανταποκριθούν στις πράξεις του, κάνοντας τις επεκτάσεις επεκτάσιμες.

Τα plugin όπως και οι υπόλοιπες επεκτάσεις του Joomla, χρησιμοποιούνται επίσης από τον πυρήνα. Τα plugin μας επιτρέπουν να τροποποιήσουμε τις λειτουργίες του συστήματος χωρίς να πειράζουμε τον ήδη υπάρχον κώδικα. Για παράδειγμα, ένα plugin θα μπορούσε να χρησιμοποιηθεί για να αλλάξει το περιεχόμενο πριν αυτό εμφανιστεί. Ή λόγω χάρη να επεκτείνει τις λειτουργίες αναζήτησης. Ή ακόμα και να εφαρμόσει ένα προσαρμοσμένο μηχανισμό ταυτοποίησης.

Τα plugin χωρίζονται σε ομάδες προκειμένου να χειρίζονται συγκεκριμένα γεγονότα ανάλογα με την ομάδα στην οποία ανήκουν. Υπάρχουν οι εξής ομάδες στον πυρήνα:

- Authentication
- Content
- Editors
- Editors-xtid
- Search
- System
- User
- Xmlrpc

Αλλά όπως αναφέρθηκε μπορεί να δημιουργηθεί ένα plugin που ανήκει σε προσαρμοσμένες ομάδες. Για παράδειγμα, αν κάποιος ήθελε να φτιάξει ένα plugin ειδικά για ένα συγκεκριμένο component, θα μπορούσε να δημιουργήσει μία προσαρμοσμένη ομάδα plugin με το όνομα του component.

## 2.4. Joomla και SEO

### 2.4.1. Βελτιστοποίηση για τις μηχανές αναζήτησης (SEO)

Τα αρχικά SEO σημαίνουν: Search Engine Optimization [5], δηλαδή η βελτιστοποίηση της αναζήτησης των μηχανών αναζήτησης. Διαφορετικά, η βοήθεια στις μηχανές αναζήτησης, προκειμένου να αναγνωρίζουν καλύτερα το περιεχόμενο μίας σελίδας.

Οι τεχνικές βελτιστοποίησης της αναγνωρισιμότητας μιας ιστοσελίδας και κατά επέκταση μιας επιχείρησης μέσω των μηχανών αναζήτησης, ή εν συντομία (S.E.O), συνίστανται σε ένα σύνολο μεθόδων που αποσκοπούν στην προώθηση και διαφήμιση των ιστοσελίδων μέσω της ευνοϊκής κατάταξής τους στις μηχανές αναζήτησης σε συνάρτηση με τις στοχοθετημένες λέξεις - φράσεις κλειδιά που είναι σημαντικές για μια επιχείρησή.

Επίσης, στο Joomla υπάρχει η επιλογή για Search Engine Friendly (SEF) , το οποίο επιτρέπει φιλικές προς τις μηχανές αναζήτησης υπερσυνδέσεις. Αυτό βοηθάει την σελίδα, την οποία τα χρησιμοποιεί, να ανεβαίνει υψηλότερα σε σχέση με μία σελίδα με ίδιο περιεχόμενο που δεν χρησιμοποιεί SEF ή δεν τα χρησιμοποιεί το ίδιο αποδοτικά. Αλλά για να εξηγήσουμε τι ακριβώς κάνει το SEF στις διευθύνσεις μίας ιστοσελίδας θα χρησιμοποιήσουμε ένα παράδειγμα.

- [http://paradeigma.com/Joomla15/index.php?option=com\\_content&view=article&id=19&Itemid=27](http://paradeigma.com/Joomla15/index.php?option=com_content&view=article&id=19&Itemid=27)

Το παραπάνω είναι η διεύθυνση ενός ιστότοπου που δεν χρησιμοποιεί SEF υπερσυνδέσμους, ενώ σε περίπτωση που χρησιμοποιούσε θα ήταν κάπως έτσι:

- [http://paradeigma.com/Joomla15/index.php/onoma\\_arthou](http://paradeigma.com/Joomla15/index.php/onoma_arthou)

Το οποίο όπως είναι εμφανές πέρα από την βοήθεια που προσφέρει στις μηχανές αναζήτησης είναι πιο ευανάγνωστο και για τους ίδιους τους χρήστες άρα το όφελος είναι διπλό.



#### 2.4.2. Λειτουργία μηχανών αναζήτησης

Αρχικά, μια μηχανή αναζήτησης χρησιμοποιεί προγράμματα όπως είναι οι «αράχνες», spiders ή web crawlers, οι οποίοι αυτόματα ανακαλύπτουν τις διάφορες ιστοσελίδες του Παγκοσμίου Ιστού, ακολουθώντας όλους τους υπερσυνδέσμους που εντοπίζουν. Τα περιεχόμενα κάθε ιστοσελίδας αναλύονται και η πληροφορία αυτή, που εξάγεται από τις επικεφαλίδες, το κείμενο, την μετά-πληροφορία κ.α, αποθηκεύεται στις βάσεις των μηχανών αναζήτησης και κατατάσσεται. Όταν ο χρήστης πραγματοποιήσει μια αναζήτηση με βάση λέξεις ή φράσεις κλειδιά, του επιστρέφονται οι ιστοσελίδες οι οποίες έχουν μεγαλύτερη συσχέτιση με την αναζήτηση του.

Καθώς στα κριτήρια αναζήτησης του μπορεί να ανταποκρίνονται σε χιλιάδες ιστοσελίδες, κάθε μηχανή αναζήτησης χρησιμοποιεί ειδικούς αλγορίθμους οι οποίοι αξιολογούν και βαθμολογούν τις διάφορες ιστοσελίδες ώστε να τις παρουσιάσουν με συγκεκριμένη σειρά στην λίστα με τα αποτελέσματα της αναζήτησης. Η κατάταξη πραγματοποιείται συνήθως ανάλογα με την συνάφεια των κριτηρίων του χρήστη με το περιεχόμενο των ιστοσελίδων. Τα αποτελέσματα αυτά αναφέρονται στην βιβλιογραφία ως οργανικά ή φυσικά αποτελέσματα. Αν και κάθε μηχανή αναζήτησης επιστρέφει χιλιάδες αποτελέσματα, έχει αποδειχθεί ότι οι τελικοί χρήστες επισκέπτονται μόνο τα αρχικά.

Σκοπός των τεχνικών βελτιστοποίησης μέσω των μηχανών αναζήτησης είναι η βελτίωση της θέσης και του πλήθους εμφάνισης μιας ιστοσελίδας στα φυσικά αποτελέσματα μιας αναζήτησης, για διάφορες λέξεις και φράσεις κλειδιά. Έχει αποδειχθεί ότι οι SEO τεχνικές επιτυγχάνουν να αυξήσουν τόσο το πλήθος όσο και την ποιότητα των επισκεπτών μιας ιστοσελίδας, οι οποίοι είτε προβαίνουν σε αγορές είτε ζητούν κι άλλες πληροφορίες.

Οι SEO τεχνικές διαχωρίζονται σε καλόβουλες, «*white hat SEO*» και σε κακόβουλες, «*black hat SEO*». Οι πρώτες είναι αυτές οι οποίες είναι αποδεκτές από τις διάφορες μηχανές αναζήτησης και περιλαμβάνουν την αναβάθμιση και τον εμπλουτισμό των περιεχομένων και της δομής μιας ιστοσελίδας. Οι black hat SEO χρησιμοποιούν ορισμένες μη αποδεκτές τεχνικές όπως είναι οι cloaking και spamdexing.

### 2.4.3. Κατάταξη μιας ιστοσελίδας

Οι σχεδιαστές ιστοσελίδων έχουν ως στόχο την ανάδειξη των ιστοσελίδων όσο το δυνατόν ψηλότερα στη λίστα με τα φυσικά αποτελέσματα που επιστρέφονται από μια αναζήτηση. Για τον λόγο αυτό προσπαθούν με διάφορους τρόπους να προσελκύσουν αλλά και να «εξαπατήσουν» τις μηχανές αναζήτησης. Αρχικά η κατηγοριοποίηση των ιστοσελίδων από τις μηχανές αναζήτησης βασιζόταν στο περιεχόμενο και μόνο σε αυτό μιας ιστοσελίδας. Οι σχεδιαστές ιστοσελίδων απλά υπέβαλλαν την ιστοσελίδα στις μηχανές αναζήτησης κι αυτές χρησιμοποιώντας τα ειδικά προγράμματα crawlers, έψαχναν τις ιστοσελίδες, αποθήκευαν την πληροφορία που συνέλεξαν και τις κατηγοριοποιούσαν ανάλογα με το θέμα τους. Οι πρώτες μηχανές αναζήτησης βασιζόνταν στα στοιχεία που τους παρείχαν οι σχεδιαστές ιστοσελίδων. Αργότερα όμως και καθώς ο αριθμός των ιστοσελίδων αυξήθηκε με γρήγορους ρυθμούς, οι μηχανές αναζήτησης άρχισαν να χρησιμοποιούν πιο «έξυπνους» αλγόριθμους οι οποίοι βασιζόνταν σε πληροφορία όπως είναι:

1. Κείμενο μέσα στον τίτλο
2. Όνομα domain
3. Κατάλογοι διεύθυνσης ιστοσελίδων (URL) και ονόματα αρχείων
4. HTML tags: κεφαλίδες και κείμενο έντονης γραφής
5. Συχνότητα όρων
6. Εγγύτητα λέξεων κλειδιών
7. Γειτνίαση λέξεων κλειδιών
8. Ακολουθία λέξεων κλειδιών
9. Κείμενο μέσα σε NOFRAMES tags
10. Ιδιότητες εικόνων

Οι σχεδιαστές ιστοσελίδων επιθυμώντας την καλή κατάταξη των ιστοσελίδων τους εμπλέκονται σε μια διαδικασία ανταλλαγής υπερσυνδέσεων μεταξύ διαφόρων ιστοσελίδων, έτσι ώστε να αυξήσουν την εισερχόμενη ροή τους από το Google. Καθώς οι μηχανές αναζήτησης και οι σχεδιαστές ιστοσελίδας βρίσκονται σε μια διαρκή «μάχη» για την παροχή από τη μία έγκυρων αποτελεσμάτων και βελτιστοποίηση της κατάταξης των ιστοσελίδων από την άλλη, πιο «έξυπνων» αλγορίθμων κατάταξης. Καθώς όλες οι μεγάλες εταιρείες που προσφέρουν την υπηρεσία των μηχανών αναζήτησης δεν αποκαλύπτουν τους αλγόριθμους που χρησιμοποιούν για την βαθμολόγηση των

ιστοσελίδων, οι τεχνικές βελτιστοποίησης, SEO, έχουν καταλήξει ότι τέσσερα είναι τα πιο σημαντικά μέτρα για την αξιολόγηση:

1. Λέξεις κλειδιά σε tag κεφαλίδων
2. Λέξεις κλειδιά σε υπερσυνδέσμους προς μια ιστοσελίδα
3. Λέξεις κλειδιά που εμφανίζονται στο κείμενο
4. Δημοτικότητα μιας ιστοσελίδας όπως μετράται από τον αλγόριθμο PageRank της Google

Οι πρώτες τεχνικές βελτιστοποίησης της κατάταξης μιας ιστοσελίδας για τις μηχανές αναζήτησης έκαναν την εμφάνιση τους το 1997 στο Usenet, λίγα χρόνια μετά από την εμφάνιση των πρώτων μηχανών αναζήτησης. Οι χειριστές των μηχανών αναζήτησης συνειδητοποίησαν ότι οι σχεδιαστές των ιστοσελίδων προσπαθούσαν να επηρεάσουν τα φυσικά αποτελέσματα μιας αναζήτησης, πολλές φορές παραπλανώντας τις μηχανές αναζήτησης και κατά συνέπεια τον τελικό χρήστη. Το 2005 δημιουργήθηκε το AirWeb ετήσιο συνέδριο το οποίο έχει ως θέμα του την ελαχιστοποίηση των επιβλαβών συνεπειών των «επιθετικών» σχεδιαστών ιστοσελίδων. Διάφορες επιχειρήσεις χρησιμοποιούν «επιθετικές» SEO τεχνικές που έχουν ως συνέπεια την τιμωρία και την απαγόρευση των ιστοσελίδων τους από τις μηχανές αναζήτησης.

Πιο συγκεκριμένα, εταιρείες οι οποίες εμπορεύονται SEO τεχνικές σε διάφορες άλλες επιχειρήσεις για να βελτιώσουν την αναγνωρισιμότητα τους, κάνουν χρήση SEO στρατηγικών αυξημένου ρίσκου που έχουν σαν αποτέλεσμα όχι μόνο να τιμωρηθούν οι εταιρείες που παρέχουν SEO υπηρεσίες αλλά και οι επιχειρήσεις που τις προσέλαβαν. Για παράδειγμα, η Wall Street Journal σε ένα άρθρο της αναφέρει ότι η Google τιμώρησε την Traffic-Power.com και τους πελάτες της, αφαιρώντας τις ιστοσελίδες τους από το ευρετήριο της. άλλες μηχανές αναζήτησης όπως η Yahoo και η MSN εφαρμόζουν πιο ελαστική πολιτική όσον αφορά τις SEO τεχνικές, κερδίζοντας έτσι μεγάλο μερίδιο της αγοράς όσον αφορά τις μηχανές αναζήτησης. Όλες οι μηχανές αναζήτησης παρέχουν στους χρήστες έναν οδηγό με επιτρεπόμενες SEO ενέργειες καθώς και οδηγίες για την αποτελεσματικότερη βελτίωση της κατάταξης των ιστοσελίδων τους.

#### 2.4.4. Τεχνικές «White hat»

Η αποδοτική σχεδίαση μιας ιστοσελίδας είναι μόνο η αρχή της διαδικασίας SEO. Οι μηχανές αναζήτησης δεν ενδιαφέρονται για την εμφάνιση της σελίδας, ενδιαφέρονται όμως για τις τεχνικές λεπτομέρειες. Οι τεχνικές λεπτομέρειες περιλαμβάνουν δυο ειδών στοιχεία. Αυτά διακρίνονται γενικότερα σε on-page και off-page. Οπότε, έχουμε αντίστοιχα on-page optimization και off-page optimization. Επιπλέον, όπως έχει ήδη αναφερθεί οι τεχνικές βελτιστοποίησης διαχωρίζονται στις καλόβουλες, white hat, και στις κακόβουλες, black hat.

Οι «white hat» περιλαμβάνουν και κινούνται στις γενικές οδηγίες και αρχές για του τι είναι αποδεκτό και τι όχι από τις μηχανές αναζήτησης. Γενικά, επικεντρώνονται στην δημιουργία κατάλληλου περιεχομένου ώστε να είναι πιο εύκολα προσβάσιμο από τα προγράμματα crawlers της μηχανής αναζήτησης. Πολύ συχνά, οι σχεδιαστές ιστοσελίδων πραγματοποιούν «σοβαρά» λάθη και παραλείψεις κατά τον σχεδιασμό των ιστοσελίδων με αποτέλεσμα την χαμηλή κατάταξη τους στα φυσικά αποτελέσματα μιας αναζήτησης.

Οι τεχνικές «white hat SEOs» προσπαθούν να ανακαλύψουν και να διορθώσουν τα λάθη αυτά, όπως είναι μη προσπελάσιμα από την μηχανή αναζήτησης μενού, «σπασμένους» υπερσυνδέσμους και προσωρινές αποστολές σε νέες διευθύνσεις. Καθώς οι μηχανές αναζήτησης βασίζονται κυρίως στο κείμενο, πολλές από τις τεχνικές που χρησιμοποιούνται για τη εύκολη πρόσβαση σε μια ιστοσελίδα αποτελούν και αποτελεσματικές τεχνικές SEO.

Οι παρακάτω μέθοδοι θεωρούνται και οι πιο αποδοτικοί και δημοφιλείς για την βελτίωση της κατάταξης μιας ιστοσελίδας:

- Χρήση μοναδικού και σχετικού τίτλου για ονομασία κάθε ιστοσελίδας
- Αντικατάσταση ασαφών όρων με συγκεκριμένη ορολογία σχετική με το θέμα της ιστοσελίδας
- Παροχή μοναδικού και ποιοτικού περιεχομένου το οποίο να απευθύνεται στα ενδιαφέροντα των επισκεπτών
- Χρήση ακριβής περιγραφής στα meta tags ώστε να είναι πιο πληροφοριακά και αποδοτικά

- Πρόσβαση σε όλες τις ιστοσελίδες ενός δικτυακού χώρου με χρήση υπερσυνδέσμων
- Δυνατότητα των web crawlers των μηχανών αναζήτησης να προσπελάσουν μια ιστοσελίδα χωρίς τη χρήση cookies και καταγραφής (logging in)
- Ανάπτυξη “link bait” στρατηγικών: τεχνικές που δελεάζουν άλλες ιστοσελίδες για εισαγωγή υπερσυνδέσμου προς της ιστοσελίδα της επιχείρησης

Όσον αφορά την αποδοτική περιγραφή μετά-πληροφορίας, ένα εξαιρετικά δημοφιλές SEO εργαλείο είναι ο *Advanced Meta-Tags Generator*, ο οποίος παρέχει χρήσιμη βοήθεια στους σχεδιαστές ιστοσελίδων ώστε να προσθέσουν meta-tag κώδικα στην ιστοσελίδα τους για να βελτιώσουν την «ορατότητα» τους από της μηχανές αναζήτησης.

#### 2.4.5. Τεχνικές «Black hat»

Οι μηχανές αναζήτησης και οι σχεδιαστές ιστοσελίδων εμπλέκονται σε μία αέναη διαμάχη. Οι πρώτοι επιθυμούν την παράδοση έγκυρου περιεχομένου ενώ οι δεύτεροι την κατάταξη των ιστοσελίδων τους στις υψηλότερες θέσεις. Για τον λόγο αυτό, πολλοί προσπαθούν με δόλιους τρόπους και μηχανισμούς να παραπλανήσουν τόσο τις μηχανές αναζήτησης όσο και τους τελικούς χρήστες.

Η χρήση των «black hat» τεχνικών είναι τόσο παλιά όσο και οι μηχανές αναζήτησης. Οι πιο δημοφιλής είναι οι προαναφερθέντες cloaking και spamdexing



## 2.5. Ασφάλεια

Η ασφάλεια [6] [7] είναι μία διαδικασία που χρειάζεται συνεχή προσπάθεια και σε πολλαπλά πεδία. Από την στιγμή που κάποιος ασχολείται με την ανάπτυξη σελίδων για το διαδίκτυο έρχεται αντιμέτωπος με την ασφάλεια της σελίδας του. Αυτό συμβαίνει γιατί υπάρχουν αρκετοί κακόβουλοι χρήστες που για διάφορους λόγους έχουν σκοπό να βλάψουν μία σελίδα που εκτίθεται στο διαδίκτυο.

Η ασφάλεια στο διαδίκτυο είναι μια γρήγορα κινούμενη πρόκληση με συνεχείς απειλές. Δεν υπάρχει κανένας σωστός τρόπος προκειμένου να διασφαλιστεί μια ιστοσελίδα και όλες οι μέθοδοι ασφαλείας είναι αντικείμενο συνεχούς αλλαγής, σταθερής βελτίωσης και συνεχούς επανεξέτασης. Δυστυχώς, όλες οι ιστοσελίδες είναι ανοιχτές σε συνεχείς επιθέσεις λόγω του δημόσιου χαρακτήρα τους.

Συνεπώς, όπως κάθε λογισμικό, έτσι και ένα Joomla ιστοτόπος έχει προβλήματα ασφαλείας. Ακόμα και το παραμικρό ολίσθημα στο κομμάτι της ασφαλείας μπορεί να οδηγήσει στην καταστροφή, δηλαδή στο να χάσουμε τον έλεγχο της σελίδας μας.

### 2.5.1. Η εφαρμογή της ασφαλείας στον ιστότοπο

Το Joomla είναι ένα αρκετά ασφαλές σύστημα. Αλλά από την στιγμή που είναι μία εφαρμογή του διαδικτύου, με δυνατή πρόσβαση από όλους, είναι αναγκαία η κάλυψη των κενών ασφαλείας. Σε συνδυασμό με το ότι γίνεται χρήση διάφορων τεχνολογιών τα πιθανά κενά ασφαλείας που μπορούν να δημιουργηθούν είναι πάρα πολλά. Από μία μη ενημερωμένη έκδοση PHP ή SQL, έως τις λάθος ρυθμίσεις από τον πάροχο φιλοξενίας της ιστοσελίδας, ή την κακογραμμένη επέκταση μπορούν να εκθέσουν την ασφάλεια της σελίδας.

### 2.5.2. Ασφάλεια στο Joomla με χρήση άλλων τεχνολογιών

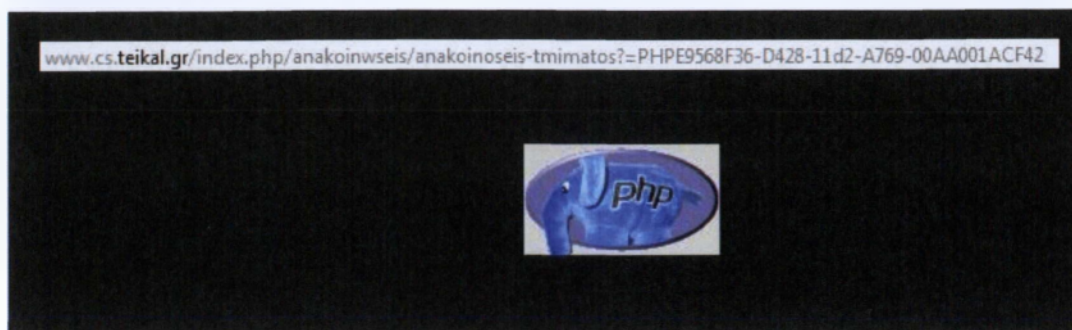
Υπάρχουν κάποια βασικά βήματα που μπορεί κάποιος να ακολουθήσει προκειμένου να προβεί σε κάποια μέτρα ασφαλείας, τα οποία είναι τα εξής:

- Αναβάθμιση της SQL, PHP, Apache, FTP Server ή όποιας άλλης τεχνολογίας γίνεται χρήση.
- Δικαιώματα αρχείων ρυθμισμένα κατάλληλα δηλαδή:



1. αποφυγή χρήσης δικαιωμάτων 777 (οποιοσδήποτε μπορεί να γράψει σε αυτά τα αρχεία)
  2. Τα δικαιώματα στους καταλόγους πρέπει να είναι 755
  3. Τα δικαιώματα των αρχείων είναι 644.
- Προσεκτική επιλογή του πάροχου φιλοξενίας (host provider), αφού πολλά κομμάτια της ασφάλειας είναι στα χέρια του πάροχου.
  - Χρήση του αρχείου .htaccess προκειμένου να αποτραπεί η άμεση πρόσβαση σε αρχεία που δεν είναι απαραίτητα η πρόσβαση τους. Σε περίπτωση που κάποια επέκταση δεν απαιτεί πρόσβαση σε κάποιο συγκεκριμένο αρχείο, η απόρριψη όλων των αρχείων εκτός του index.php και index2.php δεν θα δημιουργήσει κανένα πρόβλημα στην λειτουργία της σελίδας. Σε αντίθετη περίπτωση μπορεί να προστεθούν οι κατάλληλοι κανόνες προκειμένου να τους δοθεί πρόσβαση. Με τον παραπάνω αποκλεισμό της άμεσης πρόσβασης των αρχείων επιτυγχάνεται ένα σημαντικό επίπεδο ασφαλείας.
  - Mod\_security, ένα mod του Apache κατάλληλα ρυθμισμένο με χρήση κάποιων κανόνων που επιτρέπει την ταύτιση γνωστών hack μέσω της υπογραφής του από το να φτάσουν μέχρι την σελίδα μας.
  - Η PHP επιτρέπει την διαρροή πληροφοριών. Χρησιμοποιώντας «?=RHP9568F36-D428-11d2-A769-00AA001ACF42» πίσω από ένα σύνδεσμο μίας σελίδας σε ένα φυλλομετρητή μπορούμε να καταλάβουμε την έκδοση της PHP που υπάρχει εγκατεστημένη. Υπάρχουν διαφορετικές εικόνες για διαφορετικές ομάδες εκδόσεων άρα ανάλογα με την εικόνα την οποία θα εμφανιστεί μπορούμε να συμπεράνουμε την έκδοση.

Ένα παράδειγμα είναι το κάτωθι..



Το οποίο μας οδηγεί στο να συμπεράνουμε ότι η έκδοση που υπάρχει εγκατεστημένη στο παράδειγμα βάση της εικόνας που εμφανίστηκε αντιστοιχεί στην PHP Version 5.3.0 .

Προκειμένου να μην μοιράζεται ούτε αυτή την πληροφορία κάποιος με ολόκληρο το διαδίκτυο πρέπει να ακολουθήσει μία παρόμοια διαδικασία και να προσθέσει στο αρχείο .htaccess τα εξής:

```
RewriteCond    %{QUERY_STRING}    ^%3F=PHPE9568F36-D428-11d2-A769-00AA001ACF42 [OR]
RewriteCond    %{QUERY_STRING}    ^%3F=PHPE9568F34-D428-11d2-A769-00AA001ACF42 [OR]
RewriteCond    %{QUERY_STRING}    ^%3F=PHPE9568F35-D428-11d2-A769-00AA001ACF42 [OR]
RewriteCond    %{QUERY_STRING}    ^%3F=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 [OR]
RewriteRule    ^(.*)$ - [R=404,L]
```

Έχοντας απενεργοποιήσει έτσι όλες τις πιθανές εξόδους της PHP.

- ❖ Το ποια είναι η αντιστοιχία της κάθε εικόνας καθώς και επιπλέον πληροφορίες σχετικά υπάρχουν στο παρακάτω σύνδεσμο:  
[http://www.0php.com/php\\_easter\\_egg.php](http://www.0php.com/php_easter_egg.php)

### 2.5.3. Ασφάλεια στο Joomla με χρήση του Joomla

Ίσως ένα από τα πιο βασικά κενά ασφαλείας είναι οι ανενημέρωτες εκδόσεις. Όχι μόνο του Joomla, αλλά όλων των επεκτάσεων (components, modules, plugins αλλά ακόμα και των templates αφού και αυτά χρησιμοποιούν PHP άρα μπορούν να περιέχουν κενά ασφαλείας). Όταν το Joomla ή κάποια επέκταση αναβαθμίζει την έκδοση του λόγω κάποιου κενού ασφαλείας, τότε αυτό το κενό ασφαλείας σύντομα μπορεί να διαδοθεί και να γίνει ευρέως γνωστό. Αυτό συμβαίνει διότι με μία απλή σύγκριση των δύο εκδόσεων μπορεί κάποιος να βρει και να εκμεταλλευτεί το κενό ασφαλείας. Άρα, περισσότεροι πιθανοί κίνδυνοι για να αποκτήσει κάποιος μη εξουσιοδοτημένη πρόσβαση στην ιστοσελίδα. Συνεπώς, οι ενημερωμένες εκδόσεις και ειδικά οι ενημερώσεις ασφαλείας θα πρέπει να εφαρμόζονται όσο πιο άμεσα είναι δυνατόν.

Με την χρήση ειδικών επεκτάσεων που ο σκοπός τους είναι η ασφάλεια. Οι λειτουργίες του μπορεί να είναι κάτι απλό όπως το να μην εμφανίζουν τα πεδία για εισαγωγή στο back-end της σελίδας μέχρι να εισαχθεί στο σύνδεσμο ο κωδικός. Ή να προστατεύει ευάλωτα κομμάτια του Joomla.

Με την μη χρήση επεκτάσεων αμφιβόλου ποιότητας, αφού μία κακογραμμένη επέκταση είναι αρκετή για να δημιουργήσει κενό ασφαλείας.

Συχνό έλεγχο της λίστας επεκτάσεων που έχουν εκτεθεί (Vulnerable Extensions List) γιατί σε οποιαδήποτε επέκταση μπορεί να βρεθεί κάποιο κενό ασφαλείας και από την στιγμή που αυτό γίνει γνωστό ο κίνδυνος αυξάνεται.

Η λίστα μπορεί να βρεθεί στον εξής σύνδεσμο:  
[http://docs.joomla.org/Vulnerable\\_Extensions\\_List](http://docs.joomla.org/Vulnerable_Extensions_List)

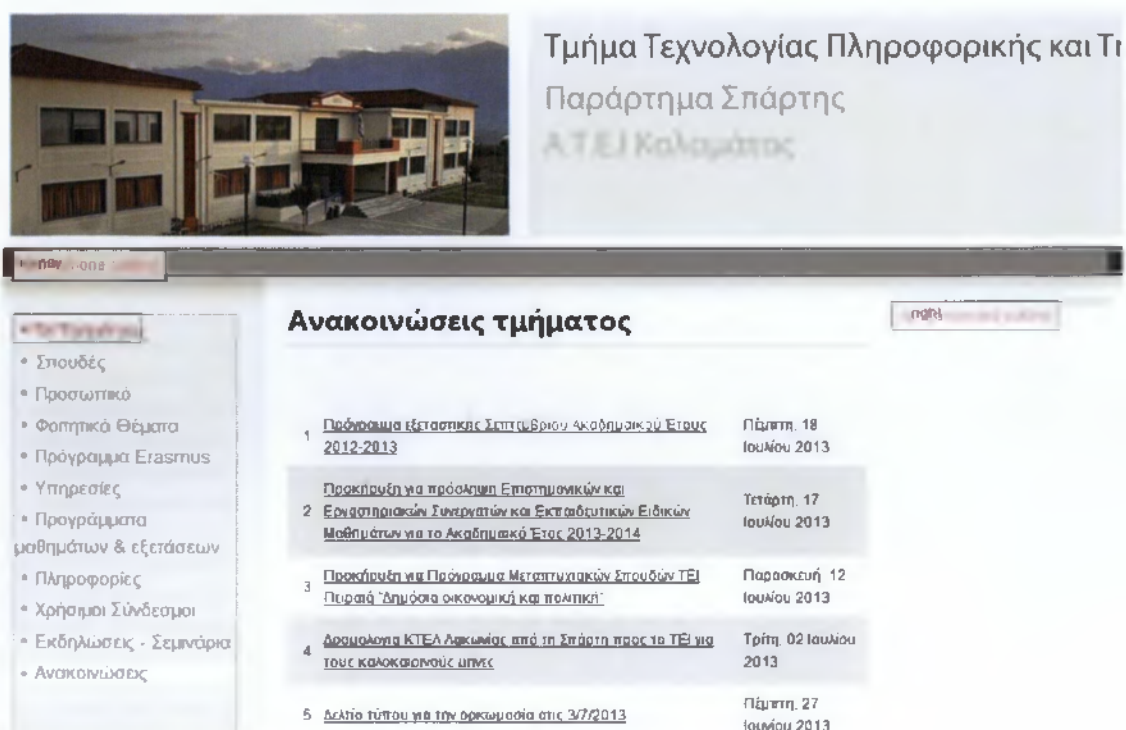
Η χρήση δυνατού κωδικού προστασίας της διαχείρισης του back-end.

Αλλαγή διάφορων προεπιλεγμένων ρυθμίσεων που περιλαμβάνουν, αλλά δεν σταματούν στο πρόθεμα των πινάκων της βάσης δεδομένων του Joomla (jos\_), το όνομα αλλά και το ID του Super Administrator (62), αλλαγή των εικόνων του πυρήνα κλπ.

- ✓ Τα παραπάνω πρέπει να αλλάζουν προκειμένου να μην γίνουν εκμεταλλεύσεις με χρήση βασισμένη στη γνώση του συστήματος και στα προκαθορισμένα

χαρακτηριστικά αυτά. Όσα λιγότερα γνωρίζει κάποιος για την σελίδα μας τόσο το καλύτερο για την ασφάλεια της.

Κρύψιμο όσο των δυνατών περισσότερων «μη-αναγκαίων» λειτουργιών που φανερώνουν πληροφορίες, όπως την εμφάνιση της έκδοσης του Joomla ή την έκδοση ενός component. Ακόμα και η πληροφορία για το εάν μία σελίδα χρησιμοποιεί ένα συγκεκριμένο CMS όπως για παράδειγμα Joomla θα μπορούσε να δώσει σε κάποιον πληροφορίες σχετικά με το τι έχει απέναντι του και τι ακριβώς να περιμένει / ψάχνει για να βρει το κενό ασφαλείας. Προσθέτοντας το `?tp=1` στην κατάληξη ενός συνδέσμου θα μας εισάγει στο debugging mode για τα module, δείχνοντας τις θέσεις των module στο template. Έτσι, αν αυτό λειτουργούσε σε μία σελίδα θα ήταν μία καλή ένδειξη ότι η συγκεκριμένη ιστοσελίδα χρησιμοποιεί Joomla, για παράδειγμα...



The screenshot shows a Joomla website interface. At the top right, there is a header with the text: "Τμήμα Τεχνολογίας Πληροφορικής και Τι Παράρτημα Σπάρτης Α.Τ.Ε.Ι. Καλαμάτας". Below this is a navigation menu with a "ΚΑΤΗΓΟΡΙΕΣ" dropdown menu containing items like "Σπουδές", "Προσωπικό", "Φοιτητικά Θέματα", "Πρόγραμμα Erasmus", "Υπηρεσίες", "Προγράμματα μαθημάτων & εξετάσεων", "Πληροφορίες", "Χρήσιμοι Σύνδεσμοι", "Εκδηλώσεις - Σεμινάρια", and "Ανακοινώσεις". The main content area is titled "Ανακοινώσεις τμήματος" and contains a list of five announcements with their respective dates.

Ανακοίνωση	Ημερομηνία
1. Πρόγραμμα εξεταστικής Σεπτεμβρίου-Ακτοβριακού Έτους 2012-2013	Πέμπτη, 18 Ιουλίου 2013
2. Προκήρυξη για πρόσληψη Επιστημονικών και Ερευνητικών Συνεργατών και Εκπαιδευτικών Ειδικών Μορφών για το Ακαδημαϊκό Έτος 2013-2014	Τετάρτη, 17 Ιουλίου 2013
3. Προκήρυξη για Πρόγραμμα Μεταπτυχιακών Σπουδών ΤΕΙ Πυραυλά "Δημόσια οικονομικά και πολεμική"	Παρασκευή, 12 Ιουλίου 2013
4. Διομολογή ΚΤΕΛ Αρκαμίας από τη Σπάρτη προς το ΤΕΙ για τους καλοκαιρινούς μήνες	Τρίτη, 02 Ιουλίου 2013
5. Δελτίο τύπου για την ορκωμοσία στις 3/7/2013	Πέμπτη, 27 Ιουλίου 2013

➤ Κάνοντας ότι περιγράφεται παραπάνω εμφανίστηκε η παραπάνω εικόνα στον φυλλομετρητή. Συμπέρασμα η παραπάνω σελίδα χρησιμοποιεί Joomla.

Θα μπορούσαμε ακόμα να χρησιμοποιήσουμε το `?template=onomatemplate` προκειμένου να αλλάζουμε το template σε κάποιο άλλο. Αυτό θα μας οδηγούσε στο ίδιο συμπέρασμα ότι δηλαδή στη σελίδα που θα γίνει "επιτυχημένα" η αλλαγή του template χρησιμοποιεί Joomla. Δοκιμάζοντας διάφορες πιθανές αλλαγές μπορούμε ακόμη και να αποκλείσουμε μερικές εκδόσεις που χρησιμοποιεί ο συγκεκριμένος ιστότοπος, αφού τα



template που έρχονται προ-εγκατεστημένα με το Joomla αλλάζουνε από έκδοση σε έκδοση.

Για παράδειγμα, σε περίπτωση που προσπαθούσαμε να αλλάξουμε το template και αυτό δε γινόταν με επιτυχία, θα συνειδητοποιούσαμε ότι αυτό το template δεν υπάρχει. Βρίσκοντας σε ποιες εκδόσεις απουσιάζει θα μπορούσαμε να αποκλείσουμε αυτές τις εκδόσεις ως τις πιθανές που χρησιμοποιούνται από αυτό τον ιστότοπο.

Τέλος, θα μπορούσαμε να προσθέσουμε το ?tmpl=offline το οποίο μας δείχνει την σελίδα σε περίπτωση που βγάξαμε την σελίδα μας εκτός σύνδεσης, σαν μία ακόμα ένδειξη σχετικά με το Joomla.

Για να αποτραπούν όλα τα παραπάνω θα πρέπει να προσθέσουμε κανόνες στο .htaccess αρχείο προκειμένου να εμφανίζει μία 404 σελίδα λάθους. Έτσι όπως θα συμπεριφερόταν δηλαδή και σε περίπτωση που δεν ήταν Joomla η σελίδα μας. Οι κανόνες που χρειάζονται είναι οι εξής:

```
RewriteCond %{QUERY_STRING} (&|%3F){1,1}tp= [OR]
RewriteCond %{QUERY_STRING} (&|%3F){1,1}template= [OR]
RewriteCond %{QUERY_STRING} (&|%3F){1,1}tmpl= [NC]
RewriteRule ^(.*)$ - [R=404,L]
```

Τα αποτυπώματα του Joomla είναι άκρως αποκαλυπτικά για την ταυτότητα του. Αλλά πως μπορεί κάποιος να μαζέψει τα αποτυπώματα ενός προγράμματος; Δυστυχώς, γίνεται πιο εύκολα από ότι ακούγεται. Χρησιμοποιώντας ένα πρόγραμμα finger printing πχ (Blind Elephant) μπορούν να συλλεχθούν αλλά και να ταυτιστούν τα αποτυπώματα του Joomla. Αυτό συμβαίνει με τα στατικά αρχεία του Joomla τα οποία αλλάζουν από έκδοση σε έκδοση και μπορούν να προδώσουν την πολύτιμη πληροφορία σχετικά με το ποια έκδοση είναι εγκατεστημένη στην σελίδα μας. Η προδοσία συμβαίνει με την γνώση αυτών των αλλαγών, οπότε σε περίπτωση αναγνώρισης κάποιας αλλαγής γίνεται μία καλή μαντεψιά της έκδοσης που υπάρχει εγκατεστημένη, κάνοντας χρήση του CSS, Javascript και εικόνων.

Για άλλη μία φορά οι κανόνες στο .htaccess παρέχουν την λύση αφού ένα κοινό χαρακτηριστικό των finger printing scripts είναι ότι δεν υποκρίνονται ότι μία σελίδα από τον ιστότοπο ζητάει τα στατικά αρχεία. Άρα, ένας κανόνας που να εμποδίζει τα στατικά

αρχεία της σελίδας να καλούνται από εξωτερική διεύθυνση από τον ιστότοπο είναι αρκετός για να τα εμποδίσει.

```
RewriteCond %{REQUEST_FILENAME} -f  
RewriteCond %{HTTP_REFERER} !^http[s]{0,1}://(.\.)*www\.example\.com [NC]  
RewriteRule \.(jpe[g,2]?|jpg|png|gif|bmp|css|js|swf|htm[lI]?)$ - [R=404,L]
```

Τα αντίγραφα ασφαλείας αν και δεν είναι μέτρο πρόληψης, θα είναι χρήσιμα που συμβεί οτιδήποτε στον ιστότοπο μας. Καλό είναι να υπάρχει συχνή λήψη αντιγράφων, ανάλογα με την συχνότητα ανανέωσης της σελίδας μας στην περίπτωση απώλειας δεδομένων. Παρόλο που αρκετοί πάροχοι φιλοξενίας παρέχουν αντίγραφα ασφαλείας καλό θα είναι να μην βασίζεται κάποιος πάνω τους. Μπορεί να είναι ελλιπή ή όχι κατάλληλα επειδή η σελίδα έχει μεγάλη συχνότητα ενημέρωσης σε σχέση με το τι θεωρεί ο πάροχος ικανοποιητικό για τους πελάτες του. Επίσης, συχνά οι επεκτάσεις που βοηθούν την λήψη αντιγράφων ασφαλείας προσφέρουν σύγκριση με παλαιότερα αντίγραφα ασφαλείας προκειμένου να ανιχνευτούν αλλαγές. Με την βοήθεια και γνώση των αλλαγών που έχουν ζητηθεί από τον χρήστη (λόγου χάρη μίας ενημέρωσης) είναι δυνατή η ταυτοποίηση ύποπτων/κακόβουλων αλλαγών. Έτσι θα μπορούσαμε να ανιχνεύσουμε ένα κενό ασφαλείας που δεν θα ήταν απαραίτητα εμφανές διαφορετικά. Όπως για παράδειγμα κλοπή των δεδομένων της βάσης ή εγκατάσταση κακόβολου λογισμικού και διαμερισμός αυτού από την σελίδα.

#### 2.5.4. Αναπτύσσοντας μία επέκταση χωρίς κενά ασφαλείας

Το πλαίσιο του Joomla (framework) περιλαμβάνει πολλές λειτουργίες οι οποίες βοηθούν με τον σκοπό της ασφάλειας. Σε περίπτωση που κάποιος κάνει χρήση αυτών των λειτουργιών επιτυγχάνει όχι μόνο το να έχει μία ασφαλή επέκταση, αλλά και να διατηρήσει την ασφάλεια ολόκληρου του ιστότοπου. Αυτό συμβαίνει επειδή οι επεκτάσεις είναι κομμάτι του Joomla, με αποτέλεσμα ένα κενό ασφαλείας σε μία επέκταση να είναι ικανό να εκθέσει ολόκληρο τον ιστότοπο σε κίνδυνο. Άρα, κάνοντας χρήση και σωστή επέκταση του Joomla αξιοποιείται η δομή του προσφέροντας ένα στρώμα ασφαλείας κάνοντας τις επεκτάσεις ασφαλείς.



Αλλά ας δούμε πως μπορεί να προστατέψει την επέκταση η χρήση του Joomla με την κατάλληλη χρήση:

- Προστασία των φορμών της επέκτασης με την χρήση μίας ειδικής τυχαίας ένδειξης που εξασφαλίζει ότι θα γίνει ταύτιση της ένδειξης του χρήστη με την σύνδεση του χρήστη με την φόρμα που υπέβαλε.
- Προστασία από άμεση πρόσβαση στα αρχεία της επέκτασης, δηλαδή πρόσβαση μόνο μέσω του Joomla είναι επιτρεπτή.
- Προστασία της επέκτασης από το να συμπεριληφθεί και να εκτελεστεί κώδικας PHP από τον χρήστη (remote file inclusion).
- Προστασία της επέκτασης από την έγχυση SQL (SQL injections) η οποία αναλύεται παρακάτω.

#### 2.5.5. *SQL injections (έγχυση SQL)*

Τα SQL injections είναι μια διαδεδομένη μέθοδος επίθεσης σε εφαρμογές web που χρησιμοποιούν συστήματα βάσεων δεδομένων. Με τη μέθοδο των SQL injections κακόβουλοι χρήστες επιδιώκουν να εκμεταλλευτούν ευπάθειες στην ασφάλεια μιας εφαρμογής web, ώστε να επέμβουν στα δεδομένα της βάσης. Ως μέθοδος επίθεσης δεν είναι νέα ούτε πολύπλοκη. Για την εφαρμογή των SQL injections αρκεί απλά ένας web browser και κοινή λογική – έτσι οι επιθέσεις SQL injections εμφανίστηκαν ταυτόχρονα με τις πρώτες εφαρμογές web.

Σκοπός των επιθέσεων SQL injections είναι η εκμετάλλευση συγκεκριμένων αδυναμιών μιας εφαρμογής web, ώστε να εκτελεστούν στη βάση δεδομένων επερωτήσεις SQL (SQL queries) που δεν έχουν προβλεφθεί από τους προγραμματιστές.

Παραδείγματα τέτοιων επερωτήσεων είναι η προβολή (SELECT) δεδομένων που δεν έχει προβλεφθεί να προβάλλονται στους χρήστες, η ανεξέλεγκτη εισαγωγή (INSERT) δεδομένων στη βάση και η διαγραφή (DELETE) δεδομένων από τη βάση.

Παρά το αυξημένο ρίσκο, θεωρείται σήμερα μεγάλος ο αριθμός των συστημάτων τα οποία είναι ευάλωτα στις επιθέσεις SQL Injections. Το ευχάριστο είναι ότι ως απειλή εύκολα μπορεί να διερευνηθεί σε μια εφαρμογή και με κατάλληλη πρόβλεψη να αποφευχθεί.

### 2.5.6. *Η διαδικασία του SQL injection*

Στα SQL injections ένας κακόβουλος χρήστης προσπαθεί να εκμεταλλευτεί την ελλιπή ή λανθασμένη επαλήθευση (validation) των δεδομένων εισόδου (input data) μιας εφαρμογής. Τυπικά η ευπάθεια των SQL injections μπορεί να εμφανίζεται και σε μη web εφαρμογές, όμως εκεί είναι σαφώς πιο σπάνιο. Συνήθεις τρόποι όπου οι χρήστες δίνουν δεδομένα εισόδου σε μια εφαρμογή web είναι οι φόρμες (με τις μεθόδους HTTP GET και POST) και τα links (μέθοδος HTTP GET). Στην περίπτωση ελλιπούς επαλήθευσης των δεδομένων εισόδου είναι σε κάποιες περιπτώσεις δυνατόν να περαστούν επιπλέον εντολές SQL προς εκτέλεση στην εφαρμογή ή να αλλαχτούν μέρη μιας επερώτησης SQL με τρόπο που δεν έχει προβλεφθεί.

### 2.5.7. *Η διαδικασία που ακολουθείται όταν η ασφάλεια έχει παραβιαστεί*

Όταν όλα τα άλλα έχουν αποτύχει να κρατήσουν το κίνδυνο μακριά. Έχει χαθεί ο έλεγχος της σελίδας και πρέπει να έχει υπάρξει κάποια προετοιμασία πριν συμβεί αυτό, τα αντίγραφα ασφαλείας. Αλλά στην περίπτωση όπου έχουμε αντιληφθεί ότι η ασφάλεια βρίσκεται σε κίνδυνο τα αντίγραφα ασφαλείας από μόνα τους δεν είναι αρκετά για να διορθώσουν την κατάσταση. Ότι οδήγησε στο να χαθεί ο έλεγχος της σελίδας μας, κατά πάσα πιθανότητα υπάρχει και στα αντίγραφα ασφαλείας, άρα χωρίς να έχει καθοριστεί το κενό ασφαλείας δεν έχει μεγάλο νόημα να επαναφέρουμε την σελίδα μας. Συνεπώς, πρέπει να ταυτοποιηθεί πρώτα ο κίνδυνος, πριν γίνει επαναφορά της σελίδας.

Το πρώτο και σημαντικότερο είναι να ταυτοποιήσουμε ότι υπήρχε κενό ασφαλείας στην σελίδα και όχι στον σέρβερ όπου φιλοξενείται αυτή, αφού τότε η κατάσταση είναι εκτός του ελέγχου. Με ανίχνευση των πρόσφατων αλλαγών στα αρχεία ή ακόμα και εύρεση νέων θα μπορούσαμε να βρούμε την απειλή. Ο έλεγχος των καταγραφών είναι επίσης χρήσιμο για εύρεση ύποπτων αιτημάτων προς την σελίδα και αναγνώριση της μεθόδου που χρησιμοποιήθηκε. Αφού λοιπόν βρεθεί η απειλή, μπορεί να γίνει η επαναφορά του ιστότοπου. Στην συνέχεια η καταπολέμηση αυτής με διαγραφή οποιουδήποτε μολυσμένου αρχείου ή εγγραφής στην βάση δεδομένων ή διόρθωση του κενού ασφαλείας. Τέλος, στενότερη παρατήρηση της ασφαλείας της σελίδας για το

επόμενο διάστημα, καθώς και αυξημένη όπου είναι δυνατό προκειμένου να αποφευχθούν μελλοντικές καταστροφές.

## 2.6. PHP

Η PHP [8] είναι μια γλώσσα script από την πλευρά του διακομιστή, σχεδιασμένη ειδικά για το Web. Μέσα σε μια HTML σελίδα μπορεί να ενσωματωθεί PHP κώδικας, που θα εκτελείται κάθε φορά που θα γίνεται επίσκεψη στη σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης.

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

Η PHP αρχικά σήμαινε Personal Home Page ( προσωπική αρχική σελίδα ), αλλά άλλαξε σύμφωνα με την σύμβαση GNU και τώρα σημαίνει PHP Hypertext Preprocessor ( προεπεξεργαστής κειμένου PHP ). Η τρέχουσα βασική έκδοση της PHP είναι η 4. Αυτή η έκδοση έχει μερικές μεγάλες βελτιώσεις στη γλώσσα.

### 2.6.1. Πλεονεκτήματα της PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι ο Perl, Microsoft Active ServerPages (ASP), JavaServerPages ( JSP ) και AllaireColdFusion. Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα όπως:

- Υψηλή απόδοση
  - Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
  - Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
  - Χαμηλό κόστος
  - Ευκολία μάθησης και χρήσης
  - Μεταφερσιμότητα
  - Διαθεσιμότητα του κώδικα προέλευσης
- ❖ Ακολουθεί μια πιο λεπτομερή περιγραφή αυτών των πλεονεκτημάτων.

### **1) Απόδοση**

Η PHP είναι πολύ αποτελεσματική. Με ένα φθηνό διακομιστή μπορεί να εξυπηρετήσει εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies ( <http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

### **2) Ολοκλήρωση με Βάσεις δεδομένων**

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, είναι δυνατή η σύνδεση κατευθείαν με τις βάσεις δεδομένων PostgreSQL, mSQL, Oracle, dbm, filePro, Informix, InterBase, Sybase, μεταξύ άλλων.

Χρησιμοποιώντας το Open Database Connectivity Standard (ODBC) μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC . Αυτό περιλαμβάνει και τα προϊόντα της Microsoft products, μεταξύ άλλων

### **3) Ενσωματωμένες Βιβλιοθήκες**

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web , έχει πολλές ενσωματωμένες βιβλιοθήκες , που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web. είναι δυνατή η δημιουργία εικόνων GIF δυναμικά, η σύνδεση με άλλες υπηρεσίες δικτύων, η αποστολή ηλεκτρονικού ταχυδρομείου, η ενασχόληση με cookies και η δημιουργία PDF εγγράφων: όλα αυτά με λίγες γραμμές κώδικα.

### **4) Κόστος**

Η PHP είναι δωρεάν. Είναι δυνατή η ανάκτηση (κατέβασμα) της τελευταίας έκδοσης από το <http://www.php.net>, χωρίς χρέωση.

### **5) Εκμάθηση της PHP**

Η σύνταξη της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού, βασικά στην C και στην Perl.

### **6) Μεταφερσιμότητα**

Η PHP είναι διαθέσιμη για πολλά λειτουργικά συστήματα. Είναι δυνατή η γραφή κώδικα PHP για δωρεάν συστήματα τύπου Unix, όπως LINUX και FreeBSD, για εμπορικές εκδόσεις του UNIX, όπως το Solaris και το IRIX ή για διαφορετικές εκδόσεις

των Microsoft Windows. Ο κώδικας σας συνήθως θα δουλεύει χωρίς αλλαγές στα συστήματα που τρέχουν την PHP.

## 7) Κώδικας προέλευσης

Υπάρχει πρόσβαση στον κώδικα προέλευσης της PHP. Αντίθετα με εμπορικά, κλειστά προγράμματα, αν υπάρχει κάτι που θέλει αλλαγή ή προσθήκη στη γλώσσα, μπορεί να γίνει. Δεν χρειάζεται να περιμένει κανείς τον κατασκευαστή να εμφανίσει διορθώσεις. Δεν θα υπάρχει ανησυχία αν ο κατασκευαστής θα σταματήσει να υπάρχει ή αν θα σταματήσει να υποστηρίζει το προϊόν.



## 2.7. MySQL

Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων σας επιτρέπει την αποθήκευση, την αναζήτηση, την ταξινόμηση και την ανάκληση των δεδομένων αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση.

Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια. Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινά από το 1979

### 2.7.1. Πλεονεκτήματα της MySQL

Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL, Microsoft SQL και Oracle.

Η MySQL έχει πολλά πλεονεκτήματα, όπως χαμηλό κόστος, εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος.

#### 1) Απόδοση

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Είναι δυνατή η επίσκεψη στην σελίδα δοκιμών <http://web.mysql.com/benchmark.html>. Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

#### 2) Χαμηλό κόστος

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

#### 3) Ευκολία Χρήσης

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

#### **4) Μεταφερσιμότητα**

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

#### **5) Κώδικας Προέλευσης**

Όπως και με την PHP, μπορεί κάποιος να πάρει και να τροποποιήσει τον κώδικα προέλευσης της MySQL.

## 2.8. CSS (Cascading Style Sheets)

Το CSS [9] [10] είναι μια απλή γλώσσα που βοηθάει να οριστεί με σαφήνεια και ιδιαίτερη ευελιξία τον τρόπο με τον οποίο θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα. Στην ουσία πρόκειται για κανόνες εμφάνισης επιτρέποντας έναν εύκολο τρόπο για έλεγχο της εμφάνισης.

Πλεονεκτήματα της χρήσης CSS έναντι της μορφοποίησης μέσω HTML attributes?

- Πολύ μεγαλύτερη ευελιξία. Το CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασική HTML.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου ιστότοπου μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στο ιστότοπο.
- Μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO (Search engine optimization). Οι μηχανές αναζήτησης δεν «μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).
- Γρηγορότερες σελίδες. Όταν γίνεται χρήση εξωτερικού αρχείου CSS, ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του ιστότοπου το αποθηκεύει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα του ιστότοπου.

### 2.8.1. Κανόνες CSS

Ένας «κανόνας» CSS αποτελείται από 2 μέρη: Τον επιλογέα (CSS selector) ο οποίος καθορίζει το τι θα μορφοποιηθεί και τις ιδιότητες (CSS properties) οι οποίες αφορούν το πως αυτό θα μορφοποιηθεί.

### 2.8.2. Εισαγωγή CSS στη σελίδα

Μπορεί να εισαχθεί CSS στη σελίδα με τρεις διαφορετικούς τρόπους, αναλόγως την περίπτωση. Οι τρόποι αυτοί, κατά σειρά φθίνουσα προτεραιότητας, είναι:

**Inline CSS:** Αν είναι επιθυμητό να μορφοποιηθεί ένα στοιχείο μόνο, και δεν πρόκειται να χρειαστεί αυτό το είδος μορφοποίησης για τίποτε άλλο στο ιστότοπο, μπορεί να γραφτεί άμεσα (δηλ. χωρίς το κομμάτι του CSS selector και χωρίς αγκύλες) CSS properties μέσα στο attribute style, το οποίο το διαθέτει σχεδόν κάθε στοιχείο html. Αξίζει να σημειωθεί, ότι αν κάποιες από τις ιδιότητες που θα γραφτούν εντός της attribute style ενός στοιχείου «συγκρούονται» με κανόνες CSS που έχουν οριστεί για αυτό αλλού και το αφορούν, τότε ο browser θα επιλέξει να εφαρμόσει αυτά που γράφτηκε εντός του attribute style, ως πιο συγκεκριμένα για το στοιχείο αυτό.

**CSS για μια συγκεκριμένη σελίδα:** Πολλές φορές, μπορεί να είναι επιθυμητό να εφαρμοστούν κάποιους κανόνες CSS μόνο για τη συγκεκριμένη σελίδα και όχι για όλο το ιστότοπο (ή να διαφοροποιηθούν σε κάποιες ιδιότητες τους υπάρχοντες). Ένας τρόπος για να γίνει αυτό και να μην δημιουργηθεί ξεχωριστό αρχείο CSS είναι να εισαχθούν εντός των tags <head>...</head> της σελίδας οι κανόνες CSS μέσα σε <style> tags (με την html attribute type τους σε text/css).

**Εξωτερικό αρχείο CSS:** Η πιο «σωστή» χρήση του CSS και αυτή που θα έπρεπε να είναι η πρώτη λύση στην οποία θα καταφύγει κάποιος είναι η χρήση εξωτερικού αρχείου CSS. Αξίζει να σημειωθεί ότι μπορούν πάντα να υπάρχουν πολλά αρχεία για την ίδια σελίδα με μεγαλύτερη προτεραιότητα να έχει πάντα αυτό που έχει γραφτεί τελευταίο.

### 2.8.3. Προτεραιότητα

Πολλές φορές για το ίδιο στοιχείο μπορεί να ταιριάζουν αρκετοί κανόνες CSS, ορισμένοι εκ των οποίων να περιέχουν ιδιότητες που «συγκρούονται».

Οι βασικότεροι είναι οι εξής:

- 1) Οι ιδιότητες CSS που βρίσκονται μέσα στην ιδιότητα style ενός στοιχείου έχουν πάντα μεγαλύτερη προτεραιότητα από οποιοσδήποτε άλλες (εκτός από όσες χρησιμοποιούν !important, το οποίο θα αναλυθεί παρακάτω)
- 2) Οι κανόνες CSS που βρίσκονται μέσα στα tags <style>...</style> έχουν μεγαλύτερη προτεραιότητα από αυτούς που βρίσκονται σε ένα εξωτερικό αρχείο CSS.
- 3) Όταν δύο ή περισσότερα αρχεία CSS συμπεριλαμβάνονται στην ίδια ιστοσελίδα, οι κανόνες αυτού που γράφτηκε τελευταίο στη σελίδα έχουν μεγαλύτερη προτεραιότητα από αυτούς του πρώτου.

- 4) Ένας κανόνας CSS έχει μεγαλύτερη προτεραιότητα από όσους κανόνες CSS βρίσκονται γραμμένοι πιο πάνω από αυτόν αν δεν συντρέχει άλλος λόγος για να μετρήσει παραπάνω κάποιος από αυτούς.
- 5) Οι κανόνες με επιλογή του τύπου `.όνομα_κλάσης` έχουν μεγαλύτερη προτεραιότητα από κανόνες με επιλογή του τύπου στοιχείο, μιας και είναι πιο συγκεκριμένοι.
- 6) Οι επιλογείς του τύπου `#όνομα_id` έχουν μεγαλύτερη προτεραιότητα από επιλογείς του τύπου `.όνομα_κλάσης`, μιας και είναι πιο συγκεκριμένοι.

Σε κάθε περίπτωση, αν είναι επιθυμητό μια ιδιότητα ενός κανόνα να έχει μεγαλύτερη προτεραιότητα από αυτή που της αντιστοιχεί, μπορεί πριν το να γράψουμε `!important`. Ωστόσο γενικά πρέπει να χρησιμοποιείται αυτή η τεχνική με φειδώ μιας και μπορεί να οδηγήσει σε μπερδεμένο κώδικα CSS.

#### 2.8.4. Κληρονομικότητα

Στην html, κάθε στοιχείο αποτελεί «απόγονο» (descendant) κάποιου άλλου στοιχείου, εκτός από το στοιχείο `<html>` που αποτελεί πρόγονο (ancestor) για κάθε στοιχείο στη σελίδα. Το κάθε στοιχείο μπορεί να έχει και δικούς του απογόνους, εκτός αν πρόκειται για self-closing στοιχεία όπως το `img` ή το `input`, τα οποία δεν διαθέτουν tag κλεισίματος (end tag). Έτσι διαμορφώνεται μια ιεραρχία, η οποία αποτελεί το DOM tree (Document Object Model tree )

Όταν εφαρμόζουμε μια ιδιότητα CSS σε ένα στοιχείο, μπορεί να κληρονομηθεί και στα στοιχεία-απογόνους του ή μπορεί και να μην κληρονομηθεί, αναλόγως την ιδιότητα. Όσον αφορά τις ιδιότητες που αναφέρθηκαν παραπάνω, αυτές που κληρονομούνται και στα στοιχεία απογόνους είναι οι:

- color
- font-size
- font-family
- font-style
- font-weight
- text-align
- text-decoration

Φυσικά αν οριστεί μέσω κάποιου κανόνα CSS μια διαφορετική τιμή για μια συγκεκριμένη ιδιότητα, θα εφαρμοστεί αυτή η τιμή και όχι η κληρονομημένη (inherited).



Όταν επιθυμείται ένα στοιχείο να αποκτήσει για μια ιδιότητα την τιμή της ιδιότητας αυτής του γονέα του ενώ αυτή κανονικά δεν κληρονομείται, είναι δυνατό ως τιμή της ιδιότητας για το στοιχείο αυτό να ορίσουμε την inherit.

### 3. ΗΛΕΚΤΡΟΝΙΚΑ ΑΘΛΗΜΑΤΑ

Ηλεκτρονικά αθλήματα ή γνωστά και ως e-sport είναι ένας όρος που αναφέρεται στα ηλεκτρονικά παιχνίδια που παίζονται σε ανταγωνιστικό επίπεδο. Τα πιο συνηθισμένα ηλεκτρονικά παιχνίδια που παίζονται ανταγωνιστικά είναι τα παιχνίδια στρατηγικής, μάχης, βολής πρώτου προσώπου (fps), αγώνων και τέλος αρένα μάχης για πολλούς παίκτες στο διαδίκτυο (MOBA). Τα παραπάνω είδη παίζονται ανταγωνιστικά σε ερασιτεχνικό, ημιεπαγγελματικό και επαγγελματικό επίπεδο. Μερικά από αυτά μάλιστα έχουν οργανωθεί προκειμένου να δημιουργήσουν πρωταθλήματα και τουρνουά. Μερικά από τα γνωστότερα τουρνουά ηλεκτρονικών αθλημάτων είναι τα εξής:

- Major League Gaming (MLG)
- European Gaming League (EGL)
- Global Starcraft II League (GSL)
- World Cyber Games (WCG)
- Dreamhack
- The Evolution Champion Series (EVO)
- Intel Extreme Masters

Μερικά απευθύνονται σε ένα μόνο παιχνίδι ή είδος αλλά μερικές διοργανώσεις όπως το Dreamhack, MLG, GSL, IntelExtremeMasters έχουν πληθώρα από είδη παιχνιδιών.

#### 3.1. Ιστορία ηλεκτρονικών αθλημάτων

Τα παιχνίδια παίζονται ανταγωνιστικά από το ξεκίνημα τους. Παρόμοια ισχύει και για τα ηλεκτρονικά παιχνίδια. Το είδος του παιχνιδιού που δημιούργησε τον πρώτο διαγωνισμό ωστόσο ήταν τα Arcade. Πιο συγκεκριμένα το τουρνουά που έγινε από την Atari το 1980, με περισσότερους από 10.000 συμμετέχοντες, οδήγησε στην εδραίωση του ανταγωνισμού στα παιχνίδια. Η έννοια του σκορ στα παιχνίδια καθώς και πίνακας κατάταξης στα παιχνίδια οδήγησε στην δημιουργία ακόμα μεγαλύτερου ανταγωνισμού.

Χάρη στο διαδίκτυο είναι δυνατή η γρήγορη και εύκολη εύρεση παιχνιδιών που παλαιότερα ήταν δυνατή μόνο με την φυσική παρουσία των παικτών στον ίδιο χώρο.

### 3.2. Το παρόν των ηλεκτρονικών αθλημάτων

Ο ρυθμός ανάπτυξης των ηλεκτρονικών αθλημάτων είναι πολύ υψηλός τα τελευταία χρόνια. Χαρακτηριστικά είναι τα νούμερα του δημοφιλέστερου παιχνιδιού League of Legends. Για την δεύτερη μόλις σεζόν, τα συνολικά βραβεία ανέρχονταν στα 2.000.000 αμερικάνικα δολάρια. Επίσης, συνολικά οι μοναδικοί επισκέπτες όπου παρακολούθησαν το γεγονός διαδικτυακά ή μέσω τηλεόρασης ανέρχονταν στους 8,2 εκατομμύρια. Ενώ οι περισσότεροι ταυτόχρονοι χρήστες ήταν 1,1 εκατομμύρια θεατές.

Πρόσφατα μάλιστα (Αύγουστο 2013), η Αμερική έδωσε το δικαίωμα στους παίκτες που αγωνίζονται σε ηλεκτρονικά αθλήματα να αναγνωρίζονται σαν αθλητές προκειμένου να υπάρχει διευκόλυνση στο θέμα εισόδου στη χώρα για συμμετοχή σε τουρνουά.

### 3.3. E-sports στην Ελλάδα

Παρόλο που η Ελλάδα είναι πίσω στο τομέα των ηλεκτρονικών αθλημάτων υπάρχουν προσπάθειες που γίνονται. Για παράδειγμα, ο Μάριος 'Nivek' Μπιτσικόκος κέρδισε το Παγκόσμιο Πρωτάθλημα στο Injustice: Gods Among Us που πραγματοποιήθηκε στο Παρίσι. Ταυτόχρονα διάφορες οργανώσεις προσπαθούν να συντελέσουν στην οργάνωση ελληνικών τουρνουά, με αρκετή επιτυχία. Ένα παράδειγμα διοργάνωσης στα ελληνικά εδάφη, είναι το γεγονός που διοργανώθηκε από το League of Legends στην Αθήνα. Το γεγονός συμπεριελάμβανε ένα ελληνικό τουρνουά με τον κόσμο που προσήλθε να ξεπερνάει τις 5.000 άτομα.



*Φωτογραφία της προσέλευσης της διοργάνωσης League Of Legends Launch Event στην Αθήνα / Γκάζι*

Επίσης, υπάρχουν εβδομαδιαία πλέον τουρνουά από μεγάλες οργανώσεις όπως το ESL (Electronic Sport League) στα δημοφιλέστερα παιχνίδια. Οι συμμετέχοντες αυξάνονται όλο και περισσότερο δίνοντας θετικά δείγματα για την εξέλιξη των ηλεκτρικών αθλημάτων στην χώρα μας.

#### **3.4. Είδος παιχνιδιού MOBA**

Moba ονομάζεται το είδος το οποίο ανήκει στην υποκατηγορία των παιχνιδιών δράσης πραγματικού χρόνου. Η διαφορά ωστόσο του είδους moba με τα παιχνίδια στρατηγικής είναι ότι δεν υπάρχει κατασκευή μονάδων και οι παίκτες ελέγχουν μονάχα ένα χαρακτήρα. Αυτή την στιγμή αποτελεί ένα από τα δημοφιλέστερα είδη. Οι ρίζες του είδους ξεκινάνε από το Aeon of Strife, ένα προσαρμοσμένο χάρτη του γνωστού παιχνιδιού Starcraft. Η έννοια του προσαρμοσμένου χάρτη αναφέρεται στο ότι κάνοντας χρήση εργαλείων που παρέχονται από το παιχνίδι, στο οποίο βασίζεται ο

προσαρμοσμένος χάρτης, γίνεται η δημιουργία του συγκεκριμένου χάρτη. Οι προσαρμοσμένοι χάρτες συνήθως είναι δημιουργήματα του community και όχι της εταιρίας που έφτιαξε το παιχνίδι.

Στην συνέχεια, ένας επίσης προσαρμοσμένος χάρτης δημιουργημένος για το παιχνίδι Warcraft III: Reign of Chaos & The Frozen Throne έγινε τόσο δημοφιλής που οδήγησε στην δημιουργία των πρώτων παιχνιδιών του είδους. Πολλά παιχνίδια δημιουργήθηκαν στην συνέχεια το καθένα με τις δικές του ιδιαιτερότητες. Demigod, Heroes of Newerth, Rise of the Immortal είναι μερικά από αυτά. Μεγάλο κεφάλαιο της κατηγορίας των MOBA αποτελεί το League Of Legends όπου σε μικρό χρονικό διάστημα κατάφερε να γίνει το δημοφιλέστερο παιχνίδι στον κόσμο. Αριθμεί περισσότερους από 12.000.000 παίκτες την μέρα και ξεπερνάει τους 5.000.000 ταυτόχρονους παίκτες.

#### *3.4.1. Ο σκοπός στο είδος παιχνιδιού MOBA*

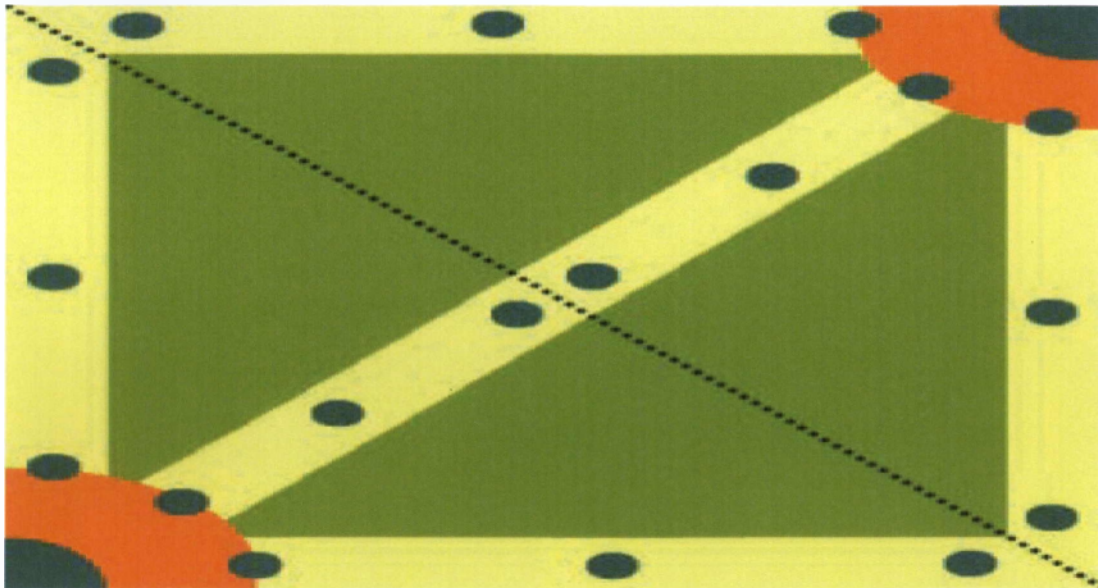
Υπάρχουν δύο αντίπαλες ομάδες των οποίων ο σκοπός είναι να καταστρέψουν την αντίπαλη βάση. Συνήθως αυτό σημαίνει ένα κεντρικό κτίριο το οποίο πρέπει να καταστραφεί προκειμένου να κερδίσει η ομάδα. Ωστόσο, υπάρχουν και άλλα κτίρια μέσα στην βάση όπου όταν καταστραφούν προσφέρουν κάποιου άλλου είδους όφελος. Προκειμένου να αποτραπεί η καταστροφή των κτιρίων υπάρχουν αμυντικά κτίρια, καθώς και σχετικά αδύναμες μονάδες που ελέγχονται από τον υπολογιστή. Οι μονάδες αυτές εμφανίζονται περιοδικά από κάθε βάση και ταξιδεύουν σε προκαθορισμένα μονοπάτια προς την αντίπαλη βάση.

Ο παίκτης ελέγχει μία δυνατή μονάδα μέσα στο παιχνίδι. Αυτή η μονάδα αναφέρεται σαν ήρωας, πρωταθλητής ή διαφορετικά ανάλογα την θεματολογία του παιχνιδιού. Ο κάθε ήρωας έχει διαφορετικά χαρακτηριστικά με αποτέλεσμα να δημιουργείται μία μεγάλη ποικιλία στους πιθανούς συνδυασμούς. Όταν ο ήρωας βρίσκεται κοντά στην εξόντωση μίας εχθρικής μονάδας αποκτάει εμπειρία. Η εμπειρία βοηθάει στο να αποκτήσει επίπεδα που τον κάνουν πιο ισχυρό και να μάθει νέες ικανότητες ή να εξελίξει τις ικανότητες που έχει ήδη. Στην περίπτωση μάλιστα που την εξόντωσε ο ίδιος αποκτάει και λεφτά. Τα λεφτά χρησιμοποιούνται στην αγορά αντικειμένων προκειμένου να γίνει πιο ισχυρός και βοηθούν στην εξέλιξη του παιχνιδιού.



Σε περίπτωση που η εχθρική μονάδα ήταν ήρωας τα χρήματα είναι ακόμα περισσότερα σε σχέση με την εξολόθρευση μονάδας ελεγχόμενη από τον υπολογιστή. Σε περίπτωση θανάτου ενός ήρωα, υπάρχει χρόνος αναμονής ο οποίος αυξάνεται ανάλογα με την πρόοδο του παιχνιδιού.

Παρόλο που το κάθε παιχνίδι έχει τις διαφορές του, όλα ακολουθούσαν τον αρχικό σχεδιασμό του χάρτη που προήλθε από τους προσαρμοσμένους χάρτες.



Ο παραπάνω χάρτης δείχνει το σύνηθες σχέδιο των χαρτών. Οι πράσινες τελείες αναφέρονται σε αμυντικά κτίρια, οι λευκές γραμμές (lanes) είναι το σημείο όπου προχωρούν οι μονάδες που ελέγχουν οι υπολογιστές. Τα πορτοκαλί σημεία είναι οι βάσεις τις κάθε ομάδας. Οι πράσινες περιοχές είναι ουδέτερες περιοχές όπου μπορούν να χρησιμοποιηθούν προκειμένου να ταξιδέψουν οι παίκτες από το ένα lane στο άλλο. Φιλοξενούν ουδέτερα τέρατα που μπορούν να εξοντωθούν για λεφτά, αλλά και διάφορα πλεονεκτήματα.

#### 3.4.2. Χαρακτηριστικά του Smite

Το Smite είναι το θέμα του ιστότοπου. Πρόκειται για ένα γνωστό ανερχόμενο ηλεκτρονικό παιχνίδι, όπου ανήκει στο είδος στρατηγικής δράσης πραγματικού χρόνου ή γνωστότερο ως αρένα μάχης για πολλούς παίκτες στο διαδίκτυο (MOBA). Η θεματολογία του είναι η χρήση μυθολογιών, συνεπώς οι ήρωες του ονομάζονται θεοί. Το παιχνίδι βρίσκεται αυτή την στιγμή στο στάδιο της ανοιχτής δοκιμής (Open-Beta). Το

παιχνίδι αναμένεται να κυκλοφορήσει στα τέλη του καλοκαιριού του 2013. Η κύρια διαφοροποίηση του με τα υπόλοιπα παιχνίδια είναι η οπτική γωνία του παίκτη.

Η συνηθισμένη οπτική στα παιχνίδια αυτού του είδους είναι η ισομετρική από πάνω, κληρονομιά από τα παιχνίδια στρατηγικής. Το Smite αντίθετα, κάνει χρήση της κάμερας τρίτου προσώπου φέρνοντας τον παίκτη πιο κοντά στην δράση. Η αλλαγή της γωνίας της κάμερας επιφέρει πολλές αλλαγές στη συνηθισμένη εμπειρία που προσφέρουν τα παιχνίδια του είδους. Μια από αυτές είναι ότι περιορίζοντας το οπτικό πεδίο γίνεται περιορισμός των πληροφοριών που έχει ο παίκτης.

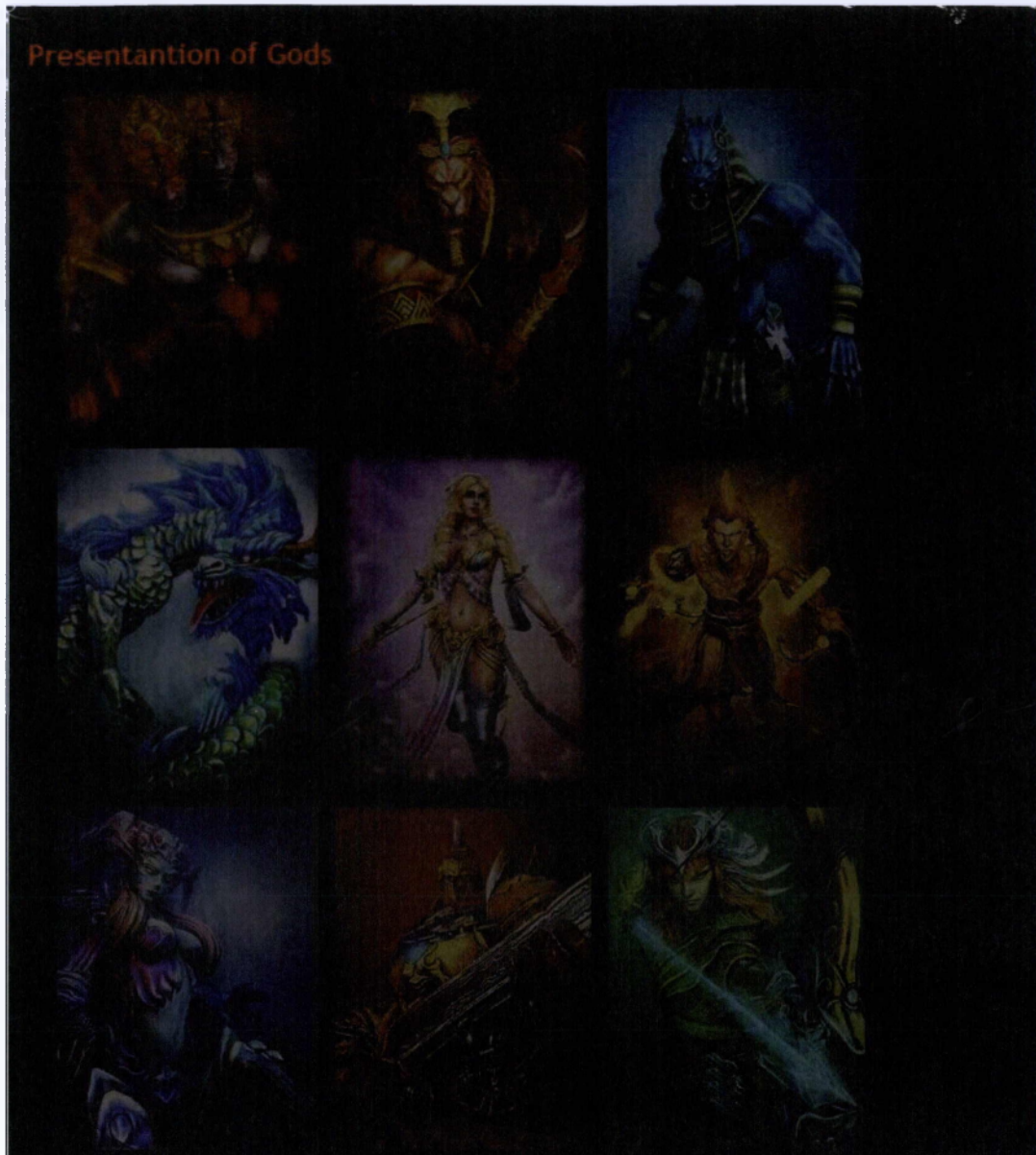
Σαν αποτέλεσμα για να ανταπεξέλθει ο παίκτης στον στρατηγικό χαρακτήρα του παιχνιδιού πρέπει να έχει τις πληροφορίες αυτές. Το οποίο οδηγεί στο ότι η επικοινωνία μεταξύ των παικτών γίνεται ακόμα πιο σημαντική καθώς και η προσωπική αντίληψη του παίκτη για το τι υπάρχει γύρω του.

## 4. ΥΛΟΠΟΙΗΣΗ

Σε αυτό το κεφάλαιο ξεκινά η παρουσίαση και η επεξήγηση του κώδικα, καθώς και η παρουσίαση τεχνικών που χρησιμοποιήθηκαν

### 4.1. Component

Τα components, αποτελούν την βασικότερη επέκταση του Joomla. Άρα και ο βαθμός πολυπλοκότητας τους είναι ο μεγαλύτερος. Όλα τα component του Joomla πρέπει να ακολουθούν τον ίδιο τρόπο ονομασίας, το οποίο είναι το όνομα τους με το πρόθεμα com\_. Το component που δημιουργήθηκε ονομάστηκε pshow. Όπως όλα τα components, έτσι και το pshow component αποτελείται από δύο κομμάτια, το front-end και το back-end.




### Περιγραφή default front-end του component

Στην πρώτη σελίδα που θα βρεθεί κάποιος όταν εισέλθει στην σελίδα του component θα είναι μία πλήρης λίστα. Η λίστα περιέχει όλους τους διαθέσιμους ήρωες ταξινομημένη κατά όνομα. Εμφανίζονται εικόνες των χαρακτήρων οι οποίες είναι ταξινομημένες σε σειρές των τριών. Οι εικόνες εμφανίζονται μόνο στην περίπτωση που ταυτίζονται με το όνομα ενός χαρακτήρα.



Μέχρι πάνω από την εικόνα να περάσει το ποντίκι. Τότε θα αποκαλυφθεί το όνομα όπου η εικόνα και το όνομα λειτουργούν σαν σύνδεσμοί.





## Ares

The Greek God of War

Released :Thursday, 04 October 2012

### Base Stats

Health	575	Mana	237	Movement	360	Range	12
Health regen	8.67	Mana regen	5.02	Attacks/second	0.91	Favor	5500
Magic Protection	30.9	Physical Protection	20	Physical Power	40.15	Gem Cost	200

The attack of Ares in the game is melee, magical. Ares is a tank. The strenght of this God is high crowd control, high defense while the weakness is low damage.

### Lore

Feared by his enemies, reviled by his father, adored by the Goddess of Beauty, and worshipped by warriors, Ares, a tortured and lonely soul, is the bloodthirsty Greek God of War. Ares is not loved by the people. He represents everything they fear and despise: war, strife, chaos, and murder. It's said that the sky darkens when Ares takes the field. That he rides to battle on a chariot drawn by fire-breathing steeds and delights only in the dash of battle and smell of bloodshed. Ares is not loved by his father. Zeus has many sons, but only two with his wife Hera. However, the thunder God has long suspected Hera of having an affair that led to the birth of Ares, perhaps as revenge for all of Zeus' infidelity. For this, Zeus is disgusted by him. Ares is not loved by his sister, Athena and Ares are both Gods of War. Where Ares represents violence, force, and slaughter, Athena is the embodiment of intelligence, strategy, and leadership. Ares is quick to act and leave a bloody trail. Athena is thoughtful and careful with lives. Often they have been on the opposite sides, most notably at the Battle of Troy, where Athena viciously wounded him and forced Ares to retreat. Ares is the lover of the Goddess of Beauty. Though Aphrodite is married to Ares' brother, Hephaestus. Ares takes what he wants, and Aphrodite gives in willingly. Their union created offspring almost as terrible as Ares himself: Phobos and Deimos, Fear and Terror. Ares is loved only by the Spartans, who revere the God for his masculinity, his merciless warrior spirit, and tenacity against all odds. The Spartans, however, know the unpredictable fury of their patron, and bind his statues in chains. This is, in part, to prevent Ares' spirit of victory from escaping Sparta, and a vain effort to keep the God of War contained.

[return to the pantheon](#)

Οι σύνδεσμοι οδηγούν στην παρουσίαση του συγκεκριμένου χαρακτήρα. Για τη συγκεκριμένη σελίδα έχουν δημιουργηθεί ξεχωριστά view, model. Σε αυτή την σελίδα υπάρχουν αναλυτικά όλα τα στοιχεία σχετικά με τον ήρωα. Τέλος, αφού τελειώσουν οι πληροφορίες σχετικά με το ήρωα, υπάρχει ένας σύνδεσμος προκειμένου να επιστρέψει ο χρήστης στην προβολή όλων των ηρώων.

- ❖ Όμως σε περίπτωση που το id του ήρωα δεν υπάρχει ή το id αντιστοιχεί σε έναν μη δημοσιευμένο ήρωα θα εμφανιστεί μήνυμα λάθους 404.

απορρίπτοντας όλα τα δεδομένα και επιστρέφοντας στη αρχική σελίδα του back-end του component.

#### *4.1.2.2. Τροποποίηση εγγραφής*

Παρόμοια με την δημιουργία ο διαχειριστής μεταφέρεται σε νέα σελίδα όπου περιέχεται η φόρμα εισαγωγής δεδομένων. Σε περίπτωση που ο διαχειριστής επιλέξει να αλλάξει μία εγγραφή μπορεί να το κάνει με εύκολο τρόπο. Αυτό συμβαίνει γιατί οι τιμές που έχει ορίσει ήδη στην εγγραφή υπάρχουν συμπληρωμένες στα πεδία της φόρμας προκειμένου να διευκολύνουν την αλλαγή. Οι επιλογές που έχει ο διαχειριστής όταν τελειώσει με την τροποποίηση των δεδομένων είναι ίδιες με την δημιουργία νέας εγγραφής, δηλαδή σώσιμο, εφαρμογή και ακύρωση.

#### *4.1.2.3. Διαγραφή, δημοσίευση, απόκρυψη*

Επιλέγοντας τις εγγραφές που είναι προς αλλαγή της κατάστασης δημοσίευσης τους ή προς διαγραφή, ο διαχειριστής μπορεί να κάνει αλλαγές σε μεγάλο νούμερο εγγραφών με ελάχιστο κόπο.

#### *4.1.3. Toolbars*

Το Joomla έχει μία ποικιλία έτοιμων κουμπιών για την διαχείριση περιεχομένου από το back-end. Χάρη σε αυτά τα κουμπιά διατηρείται μία ομοιόμορφη εμφάνιση σε όλα τα components. Έτσι, οι χρήστες αποκτούν εξοικείωση με τα κουμπιά τόσο σε εμφάνιση, όσο και στις λειτουργίες που προσφέρει το καθένα από αυτά. Σε περίπτωση που χρειάζεται όμως δίνεται η δυνατότητα να προστεθούν καινούργια κουμπιά καθώς και να αλλάξουν τα ήδη υπάρχον κουμπιά.

#### 4.1.3.1. Κώδικας toolbar

##### Αρχείο toolbar.pshow.html.php

Class TOOLBAR\_pshow

```
{  
  
    function _NEW()  
  
    {  
  
        JToolBarHelper::title(JText::_('Pantheon of Change'),'generic.png');  
  
        JToolBarHelper::save();  
  
        JToolBarHelper::apply();  
  
        JToolBarHelper::cancel();  
  
    }  
  
    function _DEFAULT()  
  
    {  
  
        JToolBarHelper::title(JText::_('God Showcase'),'generic.png');  
  
        JToolBarHelper::publishList();  
  
        JToolBarHelper::unpublishList();  
  
        JToolBarHelper::editList();  
  
        JToolBarHelper::deleteList();  
  
        JToolBarHelper::addNew();  
  
    }  
  
}
```

## Αρχείο toolbar.pshow.php

```
require_once(JApplicationHelper::getPath( 'toolbar_html' ) );

switch($task)

{

    case 'edit':

    case 'add':

        TOOLBAR_pshow::_NEW();

        break;

    default:

        TOOLBAR_pshow::_DEFAULT();

        break;

}
```

### 4.1.3.2. Ανάλυση κώδικα toolbar

Ο παραπάνω κώδικας είναι χωρισμένος σε δύο αρχεία προκειμένου να διαχωριστεί η λογική με την έξοδο του κώδικα, για να είναι πιο εύκολη η επέκταση και η αλλαγή του κώδικα. Κάθε συνάρτηση μέλος του αρχείου toolbar.pshow.html.php αντιπροσωπεύει και ένα διαφορετικό κουμπί. Η κλάση JToolBarHelper περιλαμβάνει συναρτήσεις όπου δημιουργούν όλη την απαραίτητη HTML προκειμένου να δημιουργηθούν τα κουμπιά. Επίσης με την βοήθεια του JToolBarHelper::title(JText::\_('')) είναι δυνατή η είσοδος τίτλου.

Παρόλο που τα κουμπιά έχουν οριστεί, δεν έχει καθοριστεί ακόμα ποιο σύνολο κουμπιών θα εμφανιστεί, οπότε είναι απαραίτητος ο κώδικας του αρχείου toolbar.pshow.php. Αφού φορτωθεί ο κώδικας του αρχείου toolbar.pshow.html.php, πρέπει να καθοριστεί ποια γραμμή εργαλείων θα εμφανιστεί. Αυτό θα πραγματοποιηθεί με την χρήση της μεταβλητής \$task. Αυτόματα ορίζεται σαν καθολική από το Joomla και χρησιμοποιείται προκειμένου να κατευθύνει την λογική μέσα στο component.

Να παρατηρηθεί ότι με την χρήση της `getpath` πραγματοποιείται η κλήση του αρχείου `toolbar.pshow.html.php`. Αυτό θα συμβεί ακόμα και σε περίπτωση που αλλάξει το όνομα του component και τα ονόματα των αρχείων, χωρίς να χρειαστεί να αλλάξει ο παραπάνω κώδικας.

Επίσης προκειμένου να λειτουργήσουν τα κουμπιά χρειάζονται την φόρμα με όνομα `adminForm`. Η φόρμα ενσωματώνει το `task` και το αποστέλλει αυτόματα προκειμένου να γίνονται οι κατάλληλες ενέργειες.

#### 4.1.3.3. Αποτέλεσμα του κώδικα `toolbar`

`function_NEW()`



`Function_DEFAULT()`



#### 4.1.4. *Pagination*

*Pagination*, ονομάζεται η διαδικασία διαχωρισμού πολλών δεδομένων σε μικρότερα κομμάτια εμφάνισης. Το Joomla περιλαμβάνει την λειτουργικότητα καθώς και την εμφάνιση προκειμένου να ενσωματωθεί στην έξοδο δεδομένων *pagination*. Δηλαδή θα χωριστούν τα δεδομένα σε σελίδες και θα ενσωματωθούν links προκειμένου να μπορεί ο χρήστης να προχωρήσει σε διαφορετική σελίδα.



#### 4.1.4.1. Κώδικας Pagination

##### Αρχείο controller.php (back-end)

Λόγω της ανάμιξης με τον υπόλοιπο κώδικα, ο σχετικός κώδικας που αφορά αποκλειστικά το pagination έχει τονιστεί προκειμένου να ξεχωρίζει. Δεν έγινε αφαίρεση του κώδικα προκειμένου να συνεχίσει να υπάρχει νόημα σε αυτόν.

```
function showPshow()
{
    global $option , $mainframe;

    $limit = JRequest::getVar('limit', $mainframe->getCfg('list_limit'));

    $limitstart = JRequest::getVar('limitstart', 0);

    $db =&JFactory::getDBO();

    $query = "SELECT count(*) FROM #__pshow";

    $db->setQuery( $query );

    $total = $db->loadResult();

    $query = "SELECT * FROM #__pshow";

    $db->setQuery( $query, $limitstart, $limit );

    $rows = $db->loadObjectList();

    if ($db->getErrorNum())
    {
        echo $db->stderr();

        return false;
    }

    jimport('Joomla.html.pagination');

    $pageNav = new JPagination($total, $limitstart, $limit);
}
```

```
HTML_pshow::showPshow( $option, $rows , $pageNav );
```

```
}
```

#### Αρχείο admin.pshow.html.php

```
Function showPshow( $option, &$rows , &$pageNav )
```

```
{
```

```
...
```

```
<tfoot>
```

```
<td colspan="38"><?php echo $pageNav->getListFooter(); ?></td>
```

```
</tfoot>
```

```
...
```

```
}
```





#### 4.1.4.2. Ανάλυση κώδικα *Pagination*

Οι μεταβλητές `$limit` και `$limitstart` στο αρχείο `controller.php` αντιπροσωπεύουν το μέγιστο αριθμό εγγραφών που μπορούν να παρουσιαστούν και από που να ξεκινήσουν αντίστοιχα. Σε περίπτωση που δεν έχει οριστεί κάποιο όριο από το αίτημα, ορίζεται σαν τιμή το όριο που έχει οριστεί στις ρυθμίσεις του Joomla ( το default είναι 20).

Προκειμένου να υπολογιστεί αριθμός των σελίδων που θα δημιουργηθούν χρειάζεται να είναι γνωστή ο συνολικός αριθμός των εγγραφών. Έτσι η μεταβλητή `$total` αναλαμβάνει να κρατήσει σαν τιμή της αυτό τον αριθμό. Προκειμένου να εμφανιστούν μόνο οι εγγραφές που έχουν επιλεγεί βάση του `pagination` όταν ορίζεται το `query` περνούν σαν παράμετροι οι δύο μεταβλητές `$limitstart` και `$limit`. Σαν αποτέλεσμα οι παράμετροι του `setQuery()` χρησιμοποιούνται για να ορίσουν τις εμφανιζόμενες επιθυμητές εγγραφές μόνο. Έπειτα γίνεται εισαγωγή της βιβλιοθήκης που παράγει το HTML κομμάτι του `pagination`. Στην συνέχεια, δημιουργείται μια καινούργια διάσταση της κλάσης `JPagination` με τις παραμέτρους. Οι παράμετροι είναι το σύνολο των τιμών καθώς και το εύρος των τιμών. Το αντικείμενο στην συνέχεια περνιέται στη κλάση `HTML_showpshow()`.

Στο αρχείο `admin.pshow.html.php` η κλήση στην συνάρτηση `getListFooter()` της μεταβλητής `$pageNav` επιστρέφει τον κατάλληλο HTML κώδικα για την εμφάνιση των συνδέσμων για τις υπόλοιπες σελίδες. Επιπλέον επιστρέφεται μία λίστα με τις διαθέσιμες επιλογές για το πλήθος των εμφανιζόμενων εγγραφών. Τέλος, η τρέχον σελίδα επισημαίνεται, αλλά δεν επιστρέφεται σαν σύνδεσμος.

#### 4.1.4.3. Αποτέλεσμα pagination

Display # 5    1 2 3 4 5 6 7 8   Page 6 of 8



## 4.2. Η χρήση του CSS

Σε μεγάλο βαθμό έχει γίνει η χρήση του CSS για οργάνωση της εμφάνισης της σελίδας. Παρόλο που το template αναλαμβάνει αρκετά από τα στοιχεία της εμφάνισης, προκειμένου να εμφανιστεί η επιθυμητή έξοδος, ήταν απαραίτητη η χρήση επιπλέον κώδικα. Για την καλύτερη οργάνωση αυτού χρησιμοποιήθηκε κώδικας CSS ως επί το πλείστον.

Η πιο σύνθετη χρήση CSS έγινε στην σελίδα του Component όπου εμφανίζονται όλοι οι ήρωες.



### 4.2.1. Εφέ λάμψης γύρω από την φωτογραφία

Αρχικά, ο σκοπός ήταν να τονιστούν οι φωτογραφίες όπου δημιουργήθηκε η ανάγκη για κάποιου είδους εφέ προκειμένου να παρουσιαστούν. Έτσι, κάνοντας χρήση γνωρισμάτων του CSS 3 επιτυγχάνεται η λάμψη γύρω από την φωτογραφία. Παράλληλα η φωτογραφία είναι διαφανής. Καλύπτεται με μαύρο χρώμα προκειμένου να τονιστεί ακόμα περισσότερο όταν το ποντίκι εστιάσει πάνω σε αυτήν.

#### 4.2.2. *Κρυμμένο κείμενο*

Όμως, ήταν αναγκαίο να παρουσιαστεί και ο τίτλος της φωτογραφίας, δηλαδή το όνομα του ήρωα. Έτσι με την κατάλληλη χρήση κώδικα το κείμενο παραμένει κρυμμένο έως ότου ο χρήστης εστιάσει σε μία φωτογραφία όπου και εμφανίζεται. Η παραπάνω διαδικασία πραγματοποιείται με το να υπάρχει παράμετρος που κρατά την παράγραφο του κειμένου κρυμμένη. Για να μπορέσει όμως το κείμενο να εμφανιστεί, έπρεπε να χρησιμοποιηθεί ένας διακόπτης, ο οποίος σε αυτή την περίπτωση είναι η εικόνα. Σαν αποτέλεσμα, η εικόνα αλλάζει την παράμετρο που κρατάει κρυμμένο το κείμενο κάθε φορά που εστιάζεται από το ποντίκι.

#### 4.2.3. *Διάφανο φόντο για το κείμενο πάνω από την εικόνα*

Με τον έλεγχο της τιμής position για την εικόνα και το κείμενο έγινε η τοποθέτηση του κειμένου πάνω από την εικόνα. Προκειμένου να δημιουργηθούν όμως οι κατάλληλες συνθήκες, για την εμφάνιση του κειμένου, απαιτείται σταθερό φόντο. Η ποικιλία των χρωμάτων που έχουν όλες αυτές οι εικόνες δεν το καθιστούσε δυνατό, και για το λόγο αυτό δημιουργήθηκε ημιδιάφανο φόντο. Αυτό καλύπτει την φωτογραφία ως ένα βαθμό προσφέροντας αναγνωρίσιμο κείμενο πάνω από τις εικόνες. Ταυτόχρονα όμως δεν το καλύπτει πλήρως, χάρη στο διάφανο χαρακτήρα επιτρέποντας την συνοχή της φωτογραφίας.



#### 4.2.4. Χρήση CSS αντί για Javascript

Αρχικά, ο βασικότερος λόγος που έγινε χρήση CSS αντί για Javascript είναι η μη υποστήριξη στην έκδοση Joomla 1.5 με φυσικό τρόπο. Σαν αποτέλεσμα και προκειμένου να χρησιμοποιηθεί η επιθυμητή βιβλιοθήκη Javascript, JQuery ήταν απαραίτητη η χρήση συναρτήσεων noconflict(). Αυτό συμβαίνει λόγω της βιβλιοθήκης javascript που υπάρχει εγκατεστημένη στο Joomla, Mootools. Παρόλο που στις επόμενες εκδόσεις περιλαμβάνονται λειτουργίες της βιβλιοθήκης JQuery, η βιβλιοθήκη mootools συνεχίζει να υποστηρίζεται. Ωστόσο, είναι αρκετά πιθανόν στις επόμενες εκδόσεις του Joomla να αφαιρεθεί.

Επίσης, σημαντικός λόγος είναι ότι η χρήση του CSS είναι φτιαγμένη προκειμένου να επηρεάζει την έξοδο. Το οποίο τον καθιστά ως τον ιδανικό υποψήφιο για την διαχείριση της εξόδου. Όμως, η Javascript είναι κατάλληλη σε περιπτώσεις που εμπλέκονται γεγονότα και παρόλο που μπορούσε να προσφέρει εντυπωσιακά animations επιλέχθηκε να μην χρησιμοποιηθεί.

Με αυτή την απόφαση ακόμα και χρήστες που δεν επιτρέπουν την λειτουργία javascript κώδικα στο φυλλομετρητή τους θα βλέπουν τον ιστότοπο χωρίς κανένα πρόβλημα. Αυτό συνάπτει με την θεματολογία του ιστότοπου, αφού οι πιθανοί χρήστες είναι παίκτες ηλεκτρονικών παιχνιδιών. Οι παίκτες συνήθως είναι εξοικειωμένοι με την τεχνολογία με αποτέλεσμα να επιλέγουν να κλείσουν την λειτουργία του Javascript για λόγους ασφαλείας.

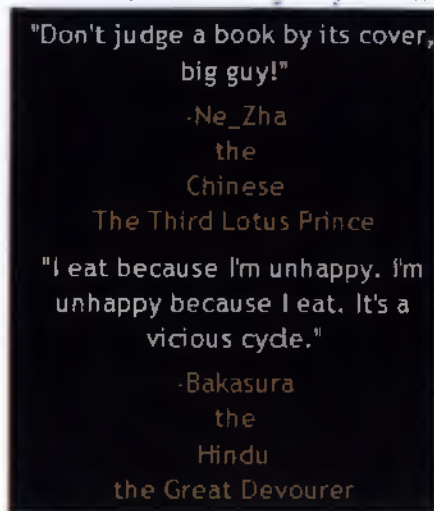
#### 4.2.5. Χρήση Javascript

Ωστόσο, λόγω της απαραίτητης χρήσης javascript για να εισέλθει κάποιος στο back-end θεωρήθηκε διαφορετική περίπτωση. Άρα, όπου χρειαζόταν έγινε χρήση javascript κώδικα προκειμένου να προσφερθούν λειτουργίες που διαφορετικά δεν θα ήταν διαθέσιμες στο διαχειριστή.

### 4.3. Module

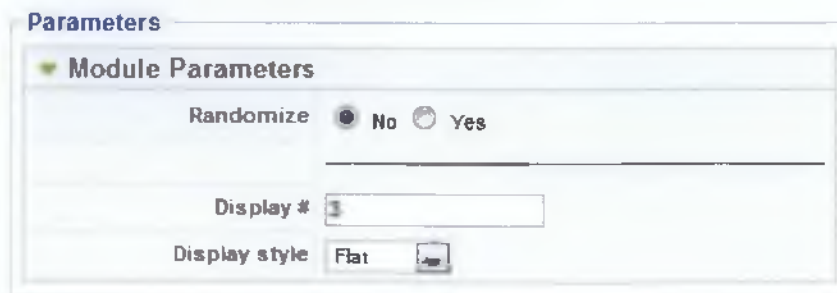
Τα module χαρακτηρίζονται από την απλότητα αλλά και ευελιξία τους. Παρόλο που μπορούν να δημιουργηθούν αρκετά πολύπλοκα module, τους αρμόζει καλύτερα η δουλειά της εμφάνισης πληροφορίας και απλές φόρμες.

#### 4.3.1. Έξοδος του module



Στην εικόνα φαίνεται η έξοδος δύο διαφορετικών quote από δύο διαφορετικούς ήρωες.

#### 4.3.2. Back-End του module



Στην εικόνα φαίνονται οι διαθέσιμες επιλογές για την τροποποίηση του module.

#### 4.3.3. Παρουσίαση Module, pshow

Ο σκοπός του συγκεκριμένου module είναι να εμφανίσει πληροφορίες από την βάση δεδομένων. Πιο συγκεκριμένα είναι να εμφανίσει το στοιχείο quote από την βάση δεδομένων που δημιουργήθηκε για το component. Πέρα από την εμφάνιση του quoteπεδίου, δημιουργείται σύνδεσμος όπου οδηγεί στις ολοκληρωμένες πληροφορίες του αντίστοιχου ήρωα. Επίσης, έχει την δυνατότητα να εμφανίσει τις εγγραφές με τυχαία σειρά ή να επιλεγθεί ο αριθμός των εγγραφών που θα εμφανιστούν.

#### 4.3.4. Ανάλυση κώδικα module, pshow.

Ο κώδικας ξεκινάει να εκτελείται από το mod\_pshow.php όπου καλείται το αρχείο helper.php. Όταν το module φορτώνεται, το αντικείμενο \$params έχει την δυνατότητα να πάρει αλλά και να αλλάξει τις παραμέτρους. Προκειμένου να επιτευχθεί η εμφάνιση συγκεκριμένου αριθμού εγγραφών από την βάση δεδομένων, γίνεται χρήση των παραμέτρων της setquery(). Οι παράμετροι της setquery() που χρησιμοποιούνται είναι τρεις. Η πρώτη είναι το ίδιο το query, στην συνέχεια η αρχική εγγραφή η οποία είναι το 0 και τέλος η τιμή της μεταβλητής \$items. Η μεταβλητή \$items είναι στην ουσία η μεταβλητή που εκφράζει το πλήθος των εγγραφών που επιθυμεί ο διαχειριστής να εμφανιστούν. Για να διαχωριστεί η λειτουργία της βάσης με την εμφάνιση γίνεται χρήση του αρχείου helper.php. Το αρχείο αυτό αναλαμβάνει την σύνδεση με την βάση δεδομένων και την επιστροφή των κατάλληλων εγγραφών, καθώς και την δημιουργία των συνδέσμων προς τους ήρωες.

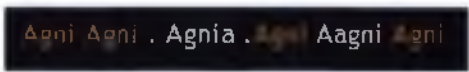
Προκειμένου να δημιουργηθεί η έξοδος γίνεται χρήση του αρχείου \_pshow.php. Το αρχείο αυτό αναλαμβάνει την έξοδο κάθε εγγραφής. Στην συνέχεια δημιουργήθηκαν δύο επιπλέον αρχεία όπου αναλαμβάνουν να τροποποιήσουν την έξοδο ώστε να προσφέρουν επιλογές στον διαχειριστή. Το default.php, είναι υπεύθυνο για την έξοδο σε περίπτωση που δεν επιλεγεί κάποια άλλη επιλογή. Για να επιτευχθεί η εμφάνιση τυχαίων quote από διαφορετικό ήρωα κάθε φορά γίνεται χρήση της rand() όπου επιλέγει έναν αριθμό ανάμεσα στο 0 και στον αριθμό των συνολικών εγγραφών. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να ικανοποιηθεί για τον προεπιλεγμένο αριθμό αντικειμένων που επιθυμεί ο διαχειριστής να εμφανιστούν.

#### 4.4. Plugin Pshow

Το plugin με όνομα pshow έχει σκοπό να μετατρέπει τα ονόματα ηρώων σε συνδέσμους προς τις πληροφορίες των αντίστοιχων ηρώων. Συχνά τα άρθρα σχετικά με τέτοιου είδους παιχνίδια είναι συζητήσεις σχετικά με τα διάφορα σενάρια ανάμεσα σε μία πληθώρα από χαρακτήρες. Με την μετατροπή σε υπερσυνδέσμους των ονομάτων των ηρώων δίνεται άμεση και χωρίς κόπο πρόσβαση στην απαιτούμενη πληροφορία.

Η πληροφορία αυτή είναι χρήσιμη τόσο στους έμπειρους παίκτες για την εμβάθυνση στη συζήτηση, όσο και για τους καινούργιους παίκτες για την κατανόηση αυτής της συζήτησης. Προκειμένου να μην δημιουργεί προβλήματα στην χρήση του, έπρεπε να αναγνωρίζει μόνο τα ονόματα των χαρακτήρων ως λέξεις. Δηλαδή σε περίπτωση που το όνομα ενός χαρακτήρα περιέχεται σε άλλες λέξεις δεν θα έπρεπε να ενεργοποιείται.

Τέλος, αυτό το plugin θα μπορούσε να φανεί χρήσιμο σε μία πληθώρα από σελίδες όπου υπάρχουν μοναδικά ονόματα όπως προϊόντων, μαθημάτων κλπ.



Agni Agni . Agnia . Agni Aagni Agni

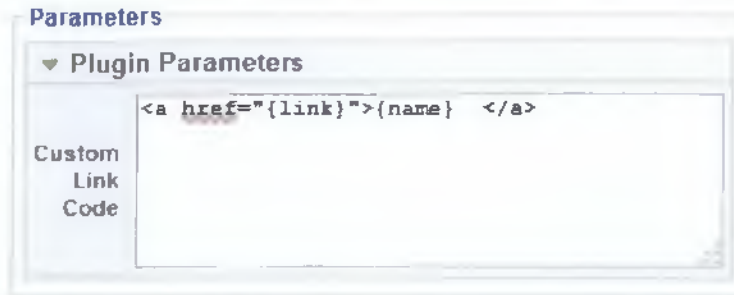
Επίδειξη χρήσης του plugin. Χρήση του ήρωα Agni ως παράδειγμα αναγνώρισης ακόμα και με την χρήση ειδικών χαρακτήρων. Επιτυχής μη αναγνώριση σε περίπτωση που περιλαμβάνεται σαν κομμάτι μεγαλύτερης λέξης.

##### 4.4.1. Ανάλυση κώδικα Plugin Pshow

Προκειμένου ένα plugin να ενεργοποιηθεί χρειάζεται ένα γεγονός. Με την χρήση του onPrepareContent γεγονότος ενεργοποιείται το plugin. Δηλαδή, όταν το Joomla φορτώνει κάποιο αντικείμενο από την βάση δεδομένων θα ενεργοποιήσει όλα τα plugin που κάνουν χρήση του συγκεκριμένου γεγονότος. Όταν ενεργοποιηθεί το plugin, θα γεμίσει ένα πίνακα με τα ονόματα όλων των ηρώων καθώς και με το ID αυτών.

Στην συνέχεια θα συγκρίνει την είσοδο του plugin με τον πίνακα αυτό προκειμένου να βρει στοιχεία που ταιριάζουν. Σε περίπτωση που αυτό γίνει με επιτυχία θα γίνει η αντικατάσταση του κειμένου με τον κατάλληλο HTML κώδικα για την δημιουργία του αντίστοιχου συνδέσμου.

#### 4.4.2. Παράμετροι Plugin Pshow



Όπως φαίνεται και στην εικόνα παραπάνω η παράμετρος που έχει προστεθεί για πιθανές αλλαγές είναι η σύνταξη του υπερσυνδέσμου. Εκεί ο διαχειριστής μπορεί κάνοντας χρήση κειμένου ή HTML κώδικα να μετατρέψει χωρίς κόπο την έξοδο του plugin σε αυτό που επιθυμεί.



#### 4.5. Plugin Pshow Information

Σκοπός του συγκεκριμένου plugin είναι να εμφανίζει πληροφορίες σχετικά με ένα ήρωα στην σελίδα με την χρήση ειδικών χαρακτήρων. Λόγω της φύσης της ιστοσελίδας οι πληροφορίες που υπάρχουν είναι αρκετές. Η δυνατότητα απομνημόνευσης όλων αυτών των πληροφοριών είναι πέρα από τις ανθρώπινες δυνατότητες. Άρα η χρήση ενός εύκολου χαρακτηριστικού για την μνήμη, όπως το όνομα, για την εμφάνιση μίας δύσκολης πληροφορίας για την μνήμη, όπως η ζωή ενός χαρακτήρα, έδωσε μορφή στο plugin.

Η εμφάνιση μορφοποιημένης πληροφορίας, καθιστά το συγκεκριμένο plugin ιδιαίτερα χρήσιμο. Όπως και το προηγούμενο plugin δεν έχει χρήση μόνο στην συγκεκριμένη σελίδα, αφού ο σκοπός είναι η χρησιμότητα του για μεγάλο αριθμό ιστοσελίδων.

Για παράδειγμα το παρακάτω άρθρο κάνει χρήση του plugin.



The Clash of the Brothers

Zeus

Information about Zeus

Health:	380
Health regeneration:	6
Mana:	245
Mana regeneration:	4.6
Attack speed:	0.87
Attack speed per level:	0.01
Physical protection:	0
Physical protection per level:	2.7
Magical protection:	30
Magical protection per level:	0
Movement:	355
Range:	55

Poseidon

Information about Poseidon

Health:	380
Health regeneration:	6
Mana:	245
Mana regeneration:	4.6
Attack speed:	0.87
Attack speed per level:	0.01
Physical protection:	8
Physical protection per level:	2.2
Magical protection:	30
Magical protection per level:	0.6
Movement:	360
Range:	55

Όπως φαίνεται από τα παραπάνω χαρακτηριστικά σε μεγάλο βαθμό οι δύο Θεοί έχουν παρόμοια χαρακτηριστικά. Ωστόσο, ο Δίας υπερτερεί στην απόκτηση προστασιών ανά επίπεδο σε επιθέσεις φυσικού είδους το οποίο σημαίνει ότι με την πάροδο του παιχνιδιού είναι πιο ανθεκτικός σε σύγκριση με τον Ποσειδώνα. Το οποίο τον κάνει πιο κατάλληλη επιλογή ενάντια σε χαρακτήρες φυσικού τύπου. Σε αντίθεση το χαρακτηριστικό στο οποίο υπερτερεί ο Ποσειδώνας αλλά σε πολύ μικρό βαθμό είναι η ταχύτητα. Επίσης λόγω της μεγαλύτερης προστασίας ανά επίπεδο ενάντια σε μαγικές επιθέσεις είναι καταλληλότερος ενάντια σε ήρωες με επιθέσεις μαγικού τύπου. Άρα κανένας από τους δύο χαρακτήρες δεν είναι καταλληλότερος για όλα τα σενάρια, αλλά εξαρτάται την περίπτωση.

Ενώ το μόνο το οποίο χρειάστηκε να γραφτεί στο άρθρο για να εμφανιστούν οι παραπάνω πληροφορίες ήταν το εξής..

Zeus

{ρι Zeus}

Poseidon

{ρι Poseidon}

Όπως φαίνεται από τα παραπάνω χαρακτηριστικά σε μεγάλο βαθμό οι δύο Θεοί έχουν παρόμοια χαρακτηριστικά. Ωστόσο, ο Δίας υπερτερεί στην απόκτηση προστασιών ανά επίπεδο σε επιθέσεις φυσικού είδους το οποίο σημαίνει ότι με την πάροδο του παιχνιδιού είναι πιο ανθεκτικός σε σύγκριση με τον Ποσειδώνα. Το οποίο τον κάνει πιο κατάλληλη επιλογή ενάντια σε χαρακτήρες φυσικού τύπου. Σε αντίθεση το χαρακτηριστικό στο οποίο υπερτερεί ο Ποσειδώνας αλλά σε πολύ μικρό βαθμό είναι η ταχύτητα. Επίσης λόγω της μεγαλύτερης προστασίας ανά επίπεδο ενάντια σε μαγικές επιθέσεις είναι καταλληλότερος ενάντια σε ήρωες με επιθέσεις μαγικού τύπου. Άρα κανένας από τους δύο χαρακτήρες δεν είναι καταλληλότερος για όλα τα σενάρια , αλλά εξαρτάται την περίπτωση.

#### 4.5.1. Ανάλυση κώδικα plugin Pshow Information

Γίνεται χρήση του γεγονότος onPrepareContent για την ενεργοποίηση του plugin. Όταν αυτό ενεργοποιείται η πρώτη του πράξη θα είναι να μαζέψει όλες τις περιπτώσεις όπου γίνεται χρήση του προκαθορισμένου κώδικα. Δηλαδή οποιοδήποτε σημείο του άρθρου περιέχει το {ρι...} θα αποθηκευτεί σε ένα πίνακα. Στην συνέχεια και με την χρήση του ονόματος ανακτώνται οι πληροφορίες για τον κάθε ήρωα. Εν συνέχεια διαμορφώνονται οι πληροφορίες με τον κατάλληλο HTML κώδικα πριν γίνει η αντικατάσταση της ετικέτας {ρι ...} με την επιθυμητή έξοδο.

#### 4.5.2. Παράμετροι plugin Pshow Information

Οι παράμετροι που προσφέρονται στο plugin είναι όλες οι πληροφορίες που περιέχονται στο πίνακα pshow προκειμένου να μπορεί να το προσαρμόσει ο διαχειριστής για να αποκτήσει την επιθυμητή έξοδο. Άρα κάθε πεδίο του πίνακα προσφέρεται σαν επιλογή για ενεργοποίηση ή απενεργοποίηση.

#### 4.6. Η εφαρμογή της ασφάλειας στον ιστότοπο

Η εφαρμογή ασφαλείας που έγινε σε αυτή την εργασία δεν περιλαμβάνει τον ιστότοπο παρά μόνο την δημιουργία ασφαλών επεκτάσεων. Η παραπάνω απόφαση πάρθηκε λόγω της παραβίασης των κανόνων ασφαλείας εξαιτίας της επιλογής της έκδοσης Joomla που επιλέχθηκε για ανάπτυξη. Η σελίδα είναι εκ φύσεως ανασφαλής αφού έχει μία μη ενημερωμένη έκδοση του Joomla, άρα η ανάπτυξη ασφαλείας πάνω σε αυτή θα ήταν ανούσια. Λόγω της χρονικής περιόδου που ξεκίνησε η ανάπτυξη της εργασίας η έκδοση που επιλέχθηκε βρισκόταν στο τέλος του κύκλου ζωής της με αποτέλεσμα κατά την διάρκεια ανάπτυξης της να σταματήσει να υποστηρίζεται. Για το λόγο αυτό αποφασίστηκε να μην αναπτυχθεί ασφάλεια σχετική με τον ιστότοπο, αφού η κάθε έκδοση έχει διαφορετικά κενά ασφαλείας, τα οποία συχνά καλύπτονται από επόμενες εκδόσεις.. Ωστόσο η ασφάλεια του ιστότοπου μελετήθηκε εκτενώς όπως και παρουσιάστηκε στο κομμάτι της θεωρίας. Για τις επεκτάσεις επιλέχθηκε να αναπτυχθεί ασφάλεια , αφού οι διαφορές είναι μικρότερες. Οι κανόνες ανάπτυξης ασφαλής επέκτασης θα παρουσιαστούν παρακάτω αναλυτικότερα.

##### 4.6.1. Ασφάλεια στην δημιουργία επέκτασης του Joomla

Τσως ο σημαντικότερος κανόνας στην ανάπτυξη επεκτάσεων είναι να γίνει η χρήση των θωρακίσεων που προσφέρονται από τον Joomla.

Για αρχή, προκειμένου να μην προσπερνιούνται οι άμυνες του Joomla πρέπει να βεβαιωθούμε ότι δεν είναι δυνατόν να εκτελεστεί ο κώδικας των αρχείων των επεκτάσεων άμεσα. Με το άμεσα εννοείται η δομή του Joomla όπως αυτή είναι στους φακέλους στους οποίους περιέχεται η επέκταση. Το πρόβλημα που θα προέκυπτε από την πρόσβαση με άμεσο τρόπο, θα ήταν ότι θα εμφανιζόταν μία πληθώρα από μηνύματα λάθους. Επιπλέον, θα μπορούσαν να βρεθούν πληροφορίες που θα εξέθεταν την ασφάλεια του συστήματος μέσω των μηνυμάτων λάθους. Προκειμένου να θωρακιστεί η επέκταση από την άμεση πρόσβαση, είναι απαραίτητη η χρήση στην αρχή κάθε PHP αρχείου η χρήση του εξής κώδικα.

```
❖ <?php
defined( '_JEXEC' ) or die( 'Restricted access' );
echo 'Restaurant Reviews';
?>
```

Ο κώδικας ελέγχει εάν «καλέστηκε» κάνοντας χρήση της εσωτερικής δομής του Joomla, δηλαδή από κάποιο άλλο σημείο της σελίδας ή άμεσα με χρήση του μονοπατιού των αρχείων. Σε περίπτωση που ο κώδικας καλέστηκε από το Joomla η πρόσβαση επιτρέπεται, διαφορετικά η σύνδεση απορρίπτεται με ένα μήνυμα Restrictedaccess.

Αυτό συμβαίνει χάρη στην σταθερά `_JEXEC` που ορίζεται στο `index.php` αρχείο στην ρίζα `root` του ιστοτόπου. Άρα, σε περίπτωση άμεσης πρόσβασης ενός αρχείου όπου περιλαμβάνεται αυτός ο κώδικας, η σταθερά `_JEXEC` δεν θα έχει οριστεί με αποτέλεσμα την απόρριψη της σύνδεσης.

- ❖ Η χρήση του παραπάνω κώδικα έγινε σε όλα τα αρχεία επιτυχάνοντας έτσι την διατήρηση της ασφάλειας που προσφέρει ο Joomla, με την απαγόρευση της άμεσης πρόσβασης.

Συνήθως όταν αναπτύσσονται PHP scripts, γίνεται χρήση σε μεγάλο βαθμό των `$_GET`, `$_POST`, `$_FILES`, `$_COOKIE`, `$_REQUEST`. Οι πιο συχνές χρήσεις είναι των `$_GET`, `$_POST`, για την αποστολή και λήψη δεδομένων. Στο Joomla παρόλο που συνεχίζουν να είναι διαθέσιμες οι παραπάνω μεταβλητές δεν συνιστάται η χρήση τους. Προκειμένου να χειριστεί τις καταστάσεις όπου χρησιμοποιούνται οι παραπάνω προκαθορισμένες μεταβλητές υπάρχει στην διάθεση μας η στατική κλάση `JRequest`. Οι παραπάνω μεταβλητές της PHP περιέχονται στις μεθόδους της κλάσης.

Με την χρήση της παραπάνω κλάσης επιτυγχάνεται ασφάλεια αφού επιτρέπει με την χρήση των μεθόδων `JRequest::get()` και `JRequest::set()` το φιλτράρισμα των τιμών. Αυτή η διεργασία επιτρέπει την εφαρμογή φίλτρων που θα έχουν ως αποτέλεσμα την απομάκρυνση χαρακτήρων που θα μπορούσαν να χρησιμοποιηθούν για τον έλεγχο του ερωτήματος προς την βάση δεδομένων. Έτσι με την χρήση των μεθόδων της κλάσεως `JRequest` επιτυγχάνεται διατήρηση της θωράκισης του Joomla.

- ❖ Ένα παράδειγμα χρήσης του παραπάνω κώδικα :

```
$id = JRequest::getVar('id', 0);
```

Κάνοντας χρήση της μεθόδου `getVar` της κλάσης `JRequest` αποκτείται η τιμή της μεταβλητής `ID` όπου σε περίπτωση που δεν έχει οριστεί θα έχει τιμή ίση με το μηδέν και αποθηκεύεται στη μεταβλητή `$id`. Λόγω έλλειψης τρίτης παραμέτρου η προεπιλεγμένη



είναι με την χρήση της `$ _REQUEST`. Διαφορετικές επιλογές που θα μπορούσαν να υπάρχουν είναι η `GET`, `POST`, `FILES`, `COOKIE`, `DEFAULT`.

Η επόμενη παράμετρος που μπορεί να χρησιμοποιηθεί είναι ότι η μεταβλητή θα είναι ενός συγκεκριμένου τύπου. Μερικές από τις διαθέσιμες επιλογές είναι `INT`, `STRING`, `PATH` όπου η καθεμία προσδιορίζει τον τύπο της μεταβλητής κατάλληλα.

Τέλος, η πέμπτη και τελευταία παράμετρος που υπάρχει στην διάθεση της `JRequest` είναι ένα φίλτρο που προσδιορίζει το επιτρεπτό περιεχόμενο του `string` του αιτήματος. Η προκαθορισμένη τιμή του πέμπτου πεδίου είναι ότι τα κενά θα εξαφανιστούν και ότι η `HTML` θα αφαιρεθεί. Στην ουσία οι διαθέσιμες καθορισμένες σταθερές που προσφέρει το Joomla είναι μόλις τρεις, οι οποίες είναι `JREQUEST_NOTRIM`, `JREQUEST_ALLOWRAW` και τέλος η `JREQUEST_ALLOWHTML`. Ωστόσο με συνδυασμό αυτών επιτυγχάνεται 8 δυνατοί συνδυασμοί.

❖ Τελικό παράδειγμα με χρήση πλήρων παραμέτρων είναι το εξής:

```
$name = JRequest::getVar('name', null, 'POST', 'STRING',JREQUEST_NOTRIM);
```

Το οποίο σημαίνει το εξής: Αίτημα για την τιμή της μεταβλητής `name`, η οποία δεν έχει κάποια προκαθορισμένη τιμή. Σε περίπτωση που δεν έχει τιμή, η επιλογή της προέλευσης των δεδομένων είναι η `POST`. Η μεταβλητή είναι τύπου `STRING`, δηλαδή κείμενο και όποια τυχόν κενά μπροστά ή πίσω από την μεταβλητή θα χαθούν.

Στην ανάπτυξη των επεκτάσεων που δημιουργήθηκαν στα πλαίσια της πτυχιακής εργασίας, έγινε ανελλιπώς χρήση της κλάσης `JRequest` και όχι των συνηθισμένων πρακτικών της `PHP`. Με την χρήση των παραμέτρων όπου κρίθηκε απαραίτητο, πραγματοποιήθηκε διατήρηση της θωράκισης του Joomla και ασφαλής χειρισμός των δεδομένων καθώς και προστασίας της βάσης δεδομένων.

Οι παράμετροι προσφέρουν έναν καλύτερα κατανοητό κώδικα αφού δίνεται καλύτερη περιγραφή του προσδιορισμού ενώ ταυτόχρονα μειώνονται οι δυνατές επιλογές επίθεσης, αφού τα δεκτά δεδομένα είναι κατά μεγάλο βαθμό λιγότερα.

#### 4.7. SEO – Η ανάπτυξη Search Engine Friendly (SEF) συνδέσμων

Προκειμένου να επιτευχθεί η δημιουργία SEF συνδέσμων μία σειρά από βήματα είναι απαραίτητο να γίνουν. Το πρώτο πράγμα που πρέπει να γίνει είναι να ενεργοποιηθεί η επιλογή για την δημιουργία συνδέσμων φιλικών προς τις μηχανές αναζήτησης. Η επιλογή βρίσκεται στο back-end του Joomla. Πηγαίνοντας στο Site -> Configuration και επιλέγοντας το ναι στην επιλογή σχετικά με Search Engine Friendly URLs ενεργοποιούνται. Σε περίπτωση που γίνεται χρήση του Apache και είναι ενεργοποιημένο το mod\_rewrite , μπορεί επίσης να χρησιμοποιηθεί προκειμένου να αφαιρεθεί το index.php από τα URLs της σελίδας.



Παραπάνω εικονίζονται οι επιλογές του Joomla που επιτρέπουν την λειτουργία των SEF συνδέσμων.

Παραδείγματα :

- Σε περίπτωση που το Usemod\_rewrite είναι στο No οι σύνδεσμοι θα φαίνονται ως εξής: <http://www.onomaselidas.gr/index.php/pshow/1>
- Σε περίπτωση που το Usemod\_rewrite είναι στο Yes οι σύνδεσμοι θα φαίνονται ως εξής: <http://www.onomaselidas.gr/pshow/1>

##### 4.7.1. Χτίζοντας τα κομμάτια του URL

Όμως τα παραπάνω δεν αρκούν για την δημιουργία SEF συνδέσμων. Κατά την δημιουργία εσωτερικών συνδέσμων, στο φτιάξιμο μίας σελίδας Joomla, το component και το module θα καλέσουν την συνάρτηση JRoute::\_(). Αυτή η συνάρτηση παίρνει ένα σύνδεσμο σαν παράμετρο και επιστρέφει ένα φιλικό προς τις μηχανές αναζήτησης σύνδεσμο. Αυτό πραγματοποιείται με το να περαστεί σε ένα πίνακα ο αρχικός σύνδεσμος, όπου αφαιρείται το στοιχείο option και προστίθεται η αξία του στο πρώτο

κομμάτι του συνδέσμου. Η συνάρτηση τότε θα κοιτάξει για το router.php στον φάκελο του component με το ίδιο όνομα με το option. Εάν το router.php βρεθεί, θα συμπεριληφθεί. Όταν γίνει αυτό η συνάρτηση που ξεκινάει με το όνομα του component και που τελειώνει σε BuildRoute() θα καλεστεί, δηλαδή PshowBuildRoute() στην περίπτωση αυτή.

4.7.2. *Κώδικας ανάπτυξης SEF συνδέσμων*  
function PshowBuildRoute(&\$query)

```
{  
  
    $segments = array();  
  
    if (isset($query['view']))  
    {  
        $segments[] = $query['view'];  
        unset($query['view']);  
    }  
  
    if(isset($query['id']))  
    {  
        $segments[] = $query['id'];  
        unset($query['id']);  
    }  
  
    return $segments;  
}
```

#### 4.7.3. Ανάλυση του κώδικα ανάπτυξης SEF συνδέσμων

Όταν η συνάρτηση `JRoute::_()` καταλαβαίνει ότι ο σύνδεσμος τον οποίο επεξεργάζεται αποτελεί σύνδεσμο για έναν ήρωα θα καλέσει την συνάρτηση `PshowBuildRoute()` με ένα πίνακα της αναλυτικής URL θα περαστεί. Προκειμένου να ολοκληρωθεί η κατασκευή ενός SEF συνδέσμου, πρέπει να επιστραφεί ένας ταξινομημένος πίνακας με τα υπόλοιπα κομμάτια του URL. Αρχικά ορίζεται η μεταβλητή `$segments` σαν ένας άδειος πίνακας. Στην συνέχεια, ελέγχεται ο πίνακας `$query` για την ύπαρξη view στοιχείου. Εάν υπάρχει, προστίθεται η τιμή του view σαν το πρώτο στοιχείο της μεταβλητής `$segments` και τότε αφαιρείται το view από το `$query`. Επαναλαμβάνεται η ίδια διαδικασία για το `id`. Τέλος, επιστρέφεται η μεταβλητή `$segments` προκειμένου να μπορέσει η συνάρτηση `JRoute::_()` να ολοκληρώσει το URL.

Υπάρχουν δύο κρίσιμες μέθοδοι που εμπλέκονται στον τρόπο που η συνάρτηση αυτή είναι γραμμένη προκειμένου να υπάρξουν σωστά SEF URL. Σημαντικό είναι ότι ο πίνακας `$query` πρέπει να περαστεί με αναφορά (να προηγείται το σύμβολο `&` στον ορισμό της συνάρτησης). Ενώ κατασκευάζονται τα κομμάτια του URL, απομακρύνονται τα επεξεργασμένα στοιχεία από τον πίνακα `$query`. Οποιοδήποτε στοιχείο έμεινε στον πίνακα, μετά την συνάρτηση θα σταλεί πίσω στο URL και θα εμφανιστεί σαν GET στοιχείο. Σε περίπτωση που ο πίνακας `$query` δεν είχε περαστεί σαν αναφορά, όταν θα καλούταν η συνάρτηση `unset()` θα επηρέαζαν μόνο τοπικά το αντίγραφο. Το αποτέλεσμα θα ήταν να εμφανιστούν όλα τα στοιχεία του URL, μετά από τα κομμάτια του SEF.

Εκτός όμως από τον σωστό έλεγχο στον πίνακα, η σειρά των στοιχείων στον πίνακα `$segments` έχει μεγάλη σημασία. Λόγω του τρόπου που παρουσιάζεται το SEF URL δεν υπάρχει τρόπος να αναγνωρίσει τα στοιχεία που ορίζει. Ο μόνος τρόπος να γίνει είναι με την προκαθορισμένη σειρά. Όταν επιστρέφεται η μεταβλητή `$segments`, η συνάρτηση `JRoute::_()` θα προσθέσει κάθε στοιχείο από τον πίνακα στο URL, χωρίζοντας τα με καθέτους. Εάν υπάρχει κάποια μεταβλητή που έχει περισσέψει στην `$query`, αυτή θα προστεθεί στο τέλος του URL σαν `GetString`.

#### 4.7.4. Αποκωδικοποιώντας τα URL

Μέχρι τώρα έχει γίνει ανάλυση της διαδικασίας δημιουργίας των SEF URL για το component. Ωστόσο είναι απαραίτητη μία συνάρτηση όπου να αποκωδικοποιεί τα URL αυτά. Για το λόγο αυτό υπάρχει η συνάρτηση PshowParseRoute().

##### 4.7.4.1. Κώδικας αποκωδικοποίησης URL

```
function PshowParseRoute($segments)
```

```
{  
  
    $vars = array();  
  
    $vars['view'] = $segments[0];  
  
    if (count($segments) > 1)  
    {  
        $vars['id'] = $segments[1];  
    }  
  
    return $vars;  
}
```

##### 4.7.4.2. Ανάλυση του κώδικα αποκωδικοποίησης URL

Μόλις το Joomla καθορίσει ότι το αίτημα της σελίδας πρόκειται για το component pshow, θα καλεστεί η συνάρτηση PshowParseRoute() όπου θα περαστεί ένας πίνακας με τα σχετικά κομμάτια του URL. Αυτά τα κομμάτια είναι ταξινομημένα με τον ίδιο τρόπο όπως στην συνάρτηση PshowBuildRoute(). Γίνεται προετοιμασία του πίνακα \$vars όπου κρατούνται οι τιμές των μεταβλητών που επιστρέφονται στην συνάρτηση.

Στην συνέχεια, ορίζεται το στοιχείο view στον πίνακα σαν τα πρώτο του πίνακα \$segments. Σε περίπτωση που υπάρχει παραπάνω από ένα στοιχείο στον πίνακα \$segments γίνεται η υπόθεση ότι είναι id και ορίζεται η στήλη \$vars['id'] με αυτό. Μετά επιστρέφεται ένας πίνακας στο Joomla όπου με την σειρά του θέτει σαν μεταβλητές



αιτήματος. Με αυτό τον τρόπο η όλη διαδικασία της δρομολόγησης είναι προσιτή στον υπόλοιπο κώδικα.

#### 4.7.4.3. Αποτέλεσμα της δημιουργίας SEF συνδέσμων

Το αποτέλεσμα της παραπάνω διαδικασίας είναι η αλλαγή των συνδέσμων προκειμένου να είναι οι σύνδεσμοι πιο φιλικοί προς τις μηχανές αναζήτησης (καθώς και πιο μικροί και κατανοητοί).

- Άρα ο σύνδεσμος που οδηγεί σε όλους τους διαθέσιμους ήρωες θα είναι ως εξής :

:

<http://www.onomaselidas.gr/pshow/all>

- Ενώ ο σύνδεσμος που οδηγεί στο κάθε μεμονωμένο θεό είναι ως εξής :

<http://www.onomaselidas.gr/pshow/single/1>

- ❖ Σε περίπτωση που δεν είχε γίνει η παραπάνω διαδικασία ο σύνδεσμος θα ήταν ως εξής:

[http://www.onomaselidas.gr/pshow/index.php?option=com\\_pshow&view=single  
&id=1](http://www.onomaselidas.gr/pshow/index.php?option=com_pshow&view=single&id=1)

## 5. ΕΠΙΛΟΓΟΣ

### 5.1. Συμπερασματα

Με την ολοκλήρωση της πτυχιακής και έχοντας πλέον μία σφαιρική εικόνα σχετικά με το έργο υπήρξε κατανόηση κάποιων θεμάτων.

Η αναγκαιότητα σχετικά με την χρήση CMS προκειμένου να υπάρξει οργάνωση του περιεχομένου. Οι επιλογές που προσφέρονται σχετικά με την οργάνωση , προσφέρουν μία οργάνωση που διαφορετικά θα ήταν πολύ δύσκολο να επιτευχθεί. Επίσης η ευκολία που προσφέρει το CMS προκειμένου να κρατηθεί ενημερωμένη η σελίδα είναι ασύγκριτη. Αναμφισβητήτα το CMS αποτελεί το σωστό δρόμο για την ανάπτυξη των σελίδων του μέλλοντος.

Σχετικά με την δομή ωστόσο, τόσο του CMS αλλά και του MVC ήταν δυσνόητη στην αρχή. Αλλά με την εμπειρία , τα οφέλη από την δομή αυτή ήταν εμφανές. Υποστήριζαν τα θετικά χαρακτηριστικά που προσφέρει τόσο το MVC όσο και το CMS με αποτέλεσμα να τονίζουν ακόμα περισσότερο τα χαρακτηριστικά αυτά. Το καθένα με το τρόπο του επιβάλλει την οργάνωση του κώδικα, έτσι οι παραπάνω δομές επιτρέπουν εύκολη ενημέρωση ή αλλαγή του κώδικα.

Τέλος , ότι δεν προσφέρεται από το σύστημα μπορεί να προστεθεί σαν λειτουργία με τις επεκτάσεις προσφέροντας απεριόριστες δυνατότητες.

## 5.2.Μελλοντικές Επεκτασεις

- Αναβάθμιση στην τρέχουσα έκδοση του Joomla! όλων των επεκτάσεων.

Η αναβάθμιση στην τρέχον έκδοση είναι το σημαντικότερο, αλλά λόγω κακού συγχρονισμού δεν ήταν έτοιμο για την κατάθεση της παρούσας πτυχιακής. Ωστόσο, σίγουρα θα γίνει προκειμένου να μπορέσουν να δημοσιευθούν στο διαδίκτυο με ασφαλές τρόπο και να αποκτήσουν εφαρμογή όλες οι επεκτάσεις. Επίσης, με την χρήση των αναβαθμισμένων εκδόσεων υπάρχουν νέες δυνατότητες προκειμένου να αναπτυχθούν επεκτάσεις που δεν ήταν πρωτίστως πραγματοποιήσιμες.

- Εισαγωγή των αντικειμένων αλλά και των ικανοτήτων στον ιστότοπο.

Εισαγωγή νέων πληροφοριών προκειμένου να υπάρξει όλη η πληροφορία σχετικά με το παιχνίδι. Στην συνέχεια θα καταστούν δυνατόν να δημιουργηθούν εργαλεία όπου θα κάνουν χρήση όλων των δεδομένων , προκειμένου να προσφέρουν νέες πληροφορίες.

- Component

Δημιουργία λειτουργίας patch στο back-end προκειμένου να είναι δυνατή η εισαγωγή των μαζικών δεδομένων πιο εύκολη στην περίπτωση νέων ενημερώσεων.

- Module

Δημιουργία νέου module που θα επιτρέπει την δημιουργία τυχαίων γεγονότων σχετικά με ακραίες.

- Plugin

Εφαρμογή των plugin στα component. Ταυτόχρονη λειτουργία.