



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### Μηχανοργάνωση δικηγορικού γραφείου



Των φοιτητών

Βασίλειο Γρηγοριάδη, Χρήστου Μωυσιάδη

Αρ. Μητρώου: 022098, 083405

Επιβλέπων καθηγητής

Γιακουστίδης Κωνσταντίνος

Θεσσαλονίκη 2014

## Πρόλογος

Η πτυχιακή εργασία αποτελεί το τελευταίο κομμάτι των σπουδών μας στο Τμήμα Πληροφορικής του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης και φέρει τον τίτλο “Κατασκευή εφαρμογής μηχανογράφησης δικηγορικού γραφείου με διαφορετικούς ρόλους χρηστών (διαχειριστής, δικηγόρος) με δυνατότητες ενσωμάτωσης της σχετικής πληροφορίας όπως δικηγόροι, πελάτες, υποθέσεις, δικαστήριο, δίκες κ.τ.λ. Η εφαρμογή θα υλοποιηθεί σε PHP, MySQL, HTML5, CSS3, JQuery, Ajax.”

Αποτελεί δύσκολο εγχείρημα η ολοκλήρωση ενός πλήρους και απόλυτα λειτουργικού συστήματος τέτοιας κλίμακας, και γι'αυτό το λόγο σκοπός αυτής της πτυχιακής είναι η έμφαση στη λογική και χρήση ενός PHP framework αντί της σκέτης χρήσης μίας γλώσσας, καθώς και η δημιουργία ενός λειτουργικού πρωτοτύπου που να υλοποιεί την εφαρμογή σε ένα βασικό επίπεδο, και κυρίως να δείχνει τις δυνατότητες του Laravel.

Προσωπικές ευχαριστίες θα θέλαμε να δώσουμε στον κύριο Γιαγκουσιτίδη Κωνσταντίνο για την ευχάριστη υποδοχή και αποδοχή που είχε στην ιδέα μας για την πτυχιακή.

## Περίληψη

Στις μέρες μας γίνεται ολοένα και περισσότερο εμφανής η ανάγκη χρήσης της τεχνολογίας και ιδιαίτερα της πληροφορικής και των τεχνολογιών του Διαδικτύου, για την υλοποίηση και διαχείριση διαδικτυακών εφαρμογών και ιδιαίτερα ιστοσελίδων. Το διαδίκτυο ξεπερνάει τα σύνορα της φυσικής υπόστασης και μας δίνει τη δυνατότητα παρουσίασης της πληροφορίας σε ψηφιακή μορφή μέσω του διαδικτύου.

Αυτό αποτελεί μεγάλο πλεονέκτημα, καθώς η πληθώρα των εργαλείων και δυνατοτήτων που παρέχει το διαδίκτυο δίνει τη δυνατότητα σε όλους τους χρήστες του να προβάλουν σχετικά εύκολα τις απόψεις τους, τα προϊόντα τους, τις επιχειρήσεις τους, κτλ..

Επίσης, αποτελεί αναπόφευκτη συνειδητοποίηση πολλών μεγάλων αλλά και μικρών επιχειρήσεων πως είναι πολύ σημαντική η μηχανοργάνωση τους. Ο όρος μηχανοργάνωση από την ετυμολογία του σημαίνει οργάνωση επιχειρήσεων και οργανισμών με χρησιμοποίηση ηλεκτρονικών υπολογιστών.

Για παράδειγμα, μηχανογράφηση ενός ξενοδοχείου θα μπορούσε να γίνει με τη δημιουργία μίας εφαρμογής η οποία θα διαχειριζόταν τα δωμάτια, τη διαθεσιμότητά τους, τις ιδιότητές τους, την τρέχουσα κατάστασή τους, αλλά και τις ανάγκες τους.

Σε επίπεδο διαδικτυακών εφαρμογών, η PHP αποτελεί μία από τις κυριάρχες γλώσσες προγραμματισμού. Η παρούσα εργασία μελετάει την χρήση του Laravel PHP framework με σκοπό την πρωτυποποίηση ενός συστήματος μηχανοργάνωσης δικηγορικού γραφείου.

## Abstract

Nowadays, the need of technology usage and especially Information Technology and Web technologies, becomes more and more evident, when it comes to the development and administration of web applications, for instance, websites.

This is a great advantage, since most of the tools and possibilities that the Web gives, provide to all users the opportunity to easily promote their beliefs, products, companies etc..

Furthermore, it's an unavoidable realization for many small to large companies, that computerization is crucial and of utmost importance. As far as terminology is concerned, the term computerization means organization of companies and agencies with the use of computers.

For example, computerization of a hotel could be done by creating a web application that would be able to administer and manage all the hotel's rooms, their availability, their attributes, their current status, and also their needs.

When it comes to web application development, PHP is one of the leading languages in programming. This thesis is studying and testing the use of Laravel PHP framework and its goal is both a theoretical approach and the creation of a prototype for the computerization of a law firm.

## Ευχαριστίες

Στους γονείς, στους φίλους  
και σε όσους μας συμπαραστάθηκαν

Χωρίς την πολύτιμη βοήθειά και ανιδιοτελής τους αγάπη και υποστήριξη, δεν θα  
ήταν δυνατή η εκπόνηση αυτής της πτυχιακής

# Κατάλογος περιεχομένων

Πρόλογος.....	2
Περίληψη.....	3
Abstract.....	4
Ευχαριστίες.....	5
ΜΕΡΟΣ Α: .....	11
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	11
1 Υποδομή διαδικτυακών υπηρεσιών.....	11
1.1 Μία σύντομη εισαγωγή στο Internet.....	11
1.2 Βασικά χαρακτηριστικά του Internet.....	12
1.3 Πρωτόκολλα και είδη δικτύων.....	13
1.4 Domain Name System.....	14
1.5 IP και HTTP.....	15
1.6 Μοντέλο πελάτη – εξυπηρετητή.....	16
1.7 Ασφάλεια στο διαδίκτυο μέσω HTTPS.....	17
1.8 Επίλογος.....	17
2 Ιστότοποι και εξυπηρετητές.....	19
2.1 Τεχνολογία και χαρακτηριστικά ενός Ιστότουπου.....	19
2.2 Ονόματα χώρου.....	19
2.3 Web servers: εξυπηρετώντας τα δεδομένα.....	20
2.3.1 Το υλικό των web servers.....	21
2.3.2 Λογισμικό των web servers.....	22
2.4 Εξυπηρετητές βάσεων δεδομένων.....	24
2.5 Nginx, apache και Windows Server.....	25
2.6 Συστήματα διαχείρισης βάσεων δεδομένων.....	25
2.7 Αρχιτεκτονική MVC.....	27

2.8	Μοντέλο REST.....	28
2.9	Αρχιτεκτονική λογισμικού στο μοντέλο REST.....	29
2.9.1	Ιδιότητες της αρχιτεκτονικής του μοντέλου REST.....	30
2.10	Frameworks και CMS.....	31
2.10.1	Διαφορές ανάμεσα στα frameworks και τις κοινές βιβλιοθήκες.....	31
2.10.2	Προσδιορίζοντας ένα CMS.....	32
2.11	Επίλογος.....	33
3	PHP και MySQL.....	34
3.1	Server-side γλώσσες προγραμματισμού: PHP.....	34
3.2	Σύνταξη κώδικα και χαρακτηριστικά της PHP.....	36
3.3	Δυνατότητες της MySQL.....	39
3.4	Επίλογος.....	40
4	Το Διαδίκτυο στην κοινωνία και το εμπόριο.....	41
4.1	E-commerce.....	41
4.2	E-government.....	42
4.3	Ηλεκτρονικό εμπόριο και ασφάλεια ηλεκτρονικών αγορών.....	44
4.4	Κοινωνική δικτύωση και εφαρμογές.....	46
4.4.1	Facebook .....	46
4.4.2	Twitter.....	47
4.4.3	Google Plus.....	47
4.5	Επίλογος.....	47
5	Προγραμματισμός με HTML5, CSS3, JQuery και Ajax.....	49
5.1	Παρέχοντας δομή: HTML5.....	49
5.1.1	Τι είναι και ποιες οι δυνατότητες της HTML5.....	49
5.1.2	Διαφορές της HTML5 με προηγούμενες εκδόσεις.....	50
5.2	Μορφοποίηση σελίδας: CSS3.....	51

5.2.1 Τι είναι η CSS3 και πως λειτουργεί.....	51
5.2.2 Selectors.....	52
5.2.3 Animations.....	54
5.3 Προγραμματισμός με jQuery .....	55
5.3.1 Τι είναι η jQuery και ποια η σχέση της με την Javascript.....	55
5.3.2 Χαρακτηριστικά και δυνατότητες της jQuery.....	55
5.3.3 Χρήση Ajax μέσω jQuery.....	56
5.4 Επίλογος.....	58
6 Το Laravel framework της PHP .....	59
6.1 Δυνατότητες του Laravel.....	59
6.2 Model, View, Controller.....	60
6.2.1 Model.....	60
6.2.2 Views.....	60
6.2.3 Controller.....	61
6.3 Ανάπτυξη templates με το Bootstrap .....	61
6.4 Composer.....	62
6.5 Artisan.....	63
6.6 Επίλογος.....	64
ΜΕΡΟΣ Β: ΥΛΟΠΟΙΗΣΗ.....	65
7 Εξοικείωση, οργάνωση και διαχείριση του web project .....	65
7.1 Απόκτηση ονόματος χώρου.....	65
7.2 Απόκτηση εξυπηρετητή διαδικτύου.....	65
7.3 Δουλεύοντας σε Virtual Machine .....	66
7.4 Κάνοντας χρήση του MySQL Workbench.....	67
7.5 Εγκατάσταση και προετοιμασία του Laravel.....	68
7.6 Επίλογος.....	69



8	Υλοποίηση εφαρμογής μηχανοργάνωσης δικηγορικού γραφείου.....	70
8.1	Μηχανοργανώνοντας ένα δικηγορικό γραφείο.....	70
8.2	Οντότητες και έννοιες που περιλαμβάνει ένα δικηγορικό γραφείο.....	70
8.2.1	Η Υπόθεση.....	70
8.2.2	Ο πελάτης.....	70
8.2.3	Ο δικηγόρος.....	70
8.3	Κάτι για τους ρόλους του δικηγορικού γραφείου και για το μοντέλο MVC.....	71
8.4	Λίγα λόγια για το SEO.....	71
8.4.1	Μεθοδολογίες και μηχανισμοί για καλύτερα αποτελέσματα.....	72
8.5	Επίλογος.....	73
	Αναφορές.....	73
	Βιβλιογραφία.....	74

## Ευρετήριο σχημάτων

Σχήμα 1: «Σουίτα πρωτοκόλλων του Internet».....	14
Σχήμα 2: «Τοπολογίες δικτύου».....	15
Σχήμα 3: «Δενδρική δομή DNS».....	16
Σχήμα 4: «IP Header».....	17
Σχήμα 5: «Αρχιτεκτονική MVC».....	26
Σχήμα 6: «Αρχιτεκτονική MVC στο Joomla!».....	26

# 1 Υποδομή διαδικτυακών υπηρεσιών

## 1.1 Μία σύντομη εισαγωγή στο Internet

Το Internet μπορεί να οριστεί ως ένα δίκτυο δικτύων, το οποίο στις μέρες μας περιλαμβάνει μια πληθώρα συσκευών που δεν περιορίζεται μόνο σε τερματικά και Ηλεκτρονικούς Υπολογιστές, αλλά περιλαμβάνει κινητές συσκευές, εκτυπωτές, κάμερες και γενικά οποιαδήποτε συσκευή μπορεί να υποστηρίξει τα πρωτόκολλα που χρησιμοποιούνται για τη διασύνδεση ενός δικτύου ή μιας συσκευής, στο Internet.

Ξεκινώντας με ένα πολύ μικρό αριθμό δικτύων και με περιορισμό χρήσης σε στρατιωτικούς ή εκπαιδευτικούς χώρους, το Internet σήμερα “φιλοξενεί” και κατά συνέπεια συνδέει δισεκατομμύρια συσκευές και δίκτυα μεταξύ τους.

Ενδεικτικά, η χρήση του Internet στις μέρες μας επεκτείνεται σε ένα ακόμη πιο αφηρημένο επίπεδο διασύνδεσης, το κοινωνικό, συνδέοντας δισεκατομμύρια χρήστες και παρέχοντας ένα μεγάλο αριθμό υπηρεσιών στους τελικούς χρήστες. Ο τελικός χρήστης μπορεί να πάρει διάφορες μορφές στο internet. Τελικός χρήστης μπορεί να είναι ένα κράτος, ένα εκπαιδευτικό ίδρυμα, μια επιχείρηση, κτλ.

Σήμερα, το Internet αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας δισεκατομμυρίων ανθρώπων και πιο συγκεκριμένα:

- Στην εκπαίδευση, μέσω της μορφής online μαθημάτων, online εγχειριδίων κτλ
- Στην κοινωνική δικτύωση, μέσω της ανάπτυξης κοινωνικής ταυτότητας, συγγραφής blog κτλ
- Στην επικοινωνία, είτε μέσω κοινωνικών δικτύων είτε μέσω ηλεκτρονικής αλληλογραφίας
- Στην δημοκρατία, μέσω online ψηφοφοριών
- Και πολλά άλλα.

## **1.2 Βασικά χαρακτηριστικά του Internet**

Θα μπορούσε κάποιος να περιγράψει τις χαρακτηριστικές ιδιότητες του Internet με τον ίδιο τρόπο που θα περιέγραφε μία κοινωνία, ή κοινότητα αποτελούμενη από ανθρώπους, με τη σημαντική διαφορά ότι το Internet απαλείφει το πρόβλημα της απόστασης και της φυσικής παρουσίας.

Άρα, το Internet προσαρμόζεται ανάλογα με τις εκάστοτε πολιτικές διαχείρισης που εφαρμόζονται στο δίκτυο, το οποίο μπορεί να περιορίζεται σε ένα τοπικό δίκτυο ανάμεσα σε μια παρέα 10 ατόμων και να επεκτείνεται μέχρι και σε ολόκληρη ήπειρο. Ορισμένα από τα χαρακτηριστικά του συνοψίζονται παρακάτω:

### **Ταχύτητες μεταφοράς που πλησιάζουν την ταχύτητα του φωτός.**

Η ταχύτητα μεταφοράς δεδομένων πλησιάζει την ταχύτητα του φωτός. Οι καθυστερήσεις που προκύπτουν αποτελούν προϊόν πολλών παραγόντων, όπως για παράδειγμα η εξασθένηση του φυσικού μέσου, το γεγονός ότι τα πακέτα πληροφορίας πρέπει να περάσουν από πολλές δικτυακές συσκευές πριν φτάσουν στον προορισμό τους, τεχνητούς περιορισμούς που θέτονται από τον εκάστοτε παροχέα του Internet ή ακόμη και καθυστερήσεις στην τροποποίηση των δεδομένων από ψηφιακή σε αναλογική μορφή και το αντίθετο. Παρόλα αυτά, η αναλογική πληροφορία ταξιδεύει με 2/3 της ταχύτητας του φωτός, μέσω καλωδίων χαλκού.

### **Ελευθερία λόγου και ισότητα**

Θα μπορούσε κάποιος να επιχειρηματολογήσει λέγοντας πως η ποσότητα ελευθερίας εξαρτάται από τους εκάστοτε κανόνες που θεσπίζει ένα κράτος ή ένας παροχέας, παρόλα αυτά το Internet στις μέρες μας αποτελεί το πιο ελεύθερο μέρος όπου μπορεί κάποιος να εκφραστεί. Ο χρήστης, ως netizen, έχει τα ίδια δικαιώματα έκφρασης και συμπεριφοράς ανεξάρτητα από την καταγωγή του ή την ηλικία του, όσο διατηρεί την ανωνυμία του.

### **Κατανεμημένη διαχείριση**

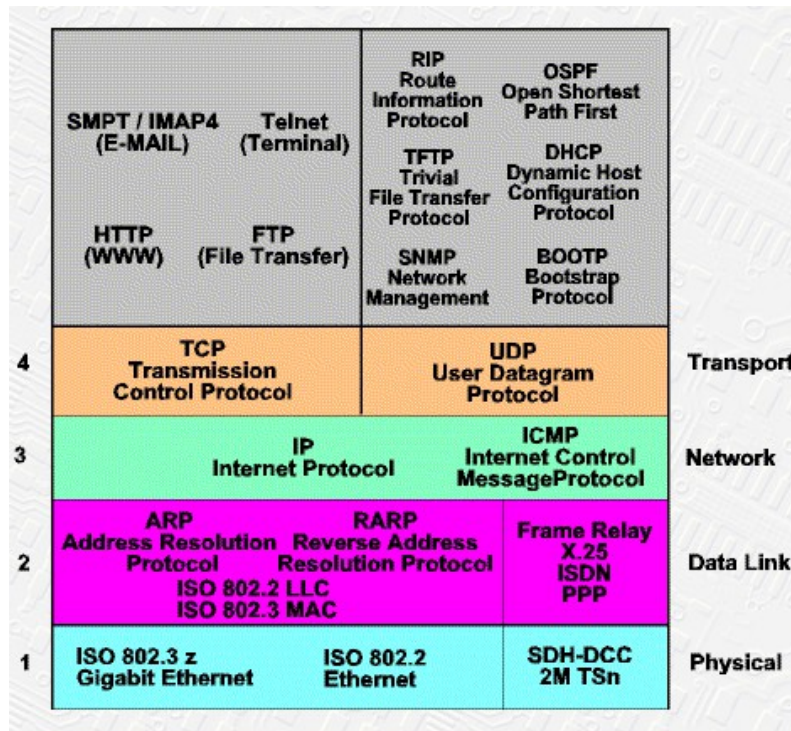
Το Internet δεν διαθέτει κεντρική διαχείριση, ούτε διακόπτη on / off όπου κάποιος θα μπορούσε να χρησιμοποιήσει για να τερματίσει τη λειτουργία του. Αυτό αποτελεί απόρροια του γεγονότος ότι οι κόμβοι, οι διασυνδέσεις και οι τρόποι διασύνδεσης είναι πλέον τόσοι πολλοί, που ακόμη και εάν κάποιο μέρος του τεθεί εκτός λειτουργίας, να υπάρχει πάντα πληθώρα τρόπων διασύνδεσης.

### **Διατήρηση δεδομένων**

Η ψηφιακή μορφή της πληροφορίας καθώς και η κατανεμημένη φύση της (αποθήκευση της ίδιας πληροφορίας σε πολλαπλά σημεία και μέσα) αποτελεί ένα από τα βασικά στοιχεία που χαρακτηρίζουν το Internet.

### 1.3 Πρωτόκολλα και είδη δικτύων

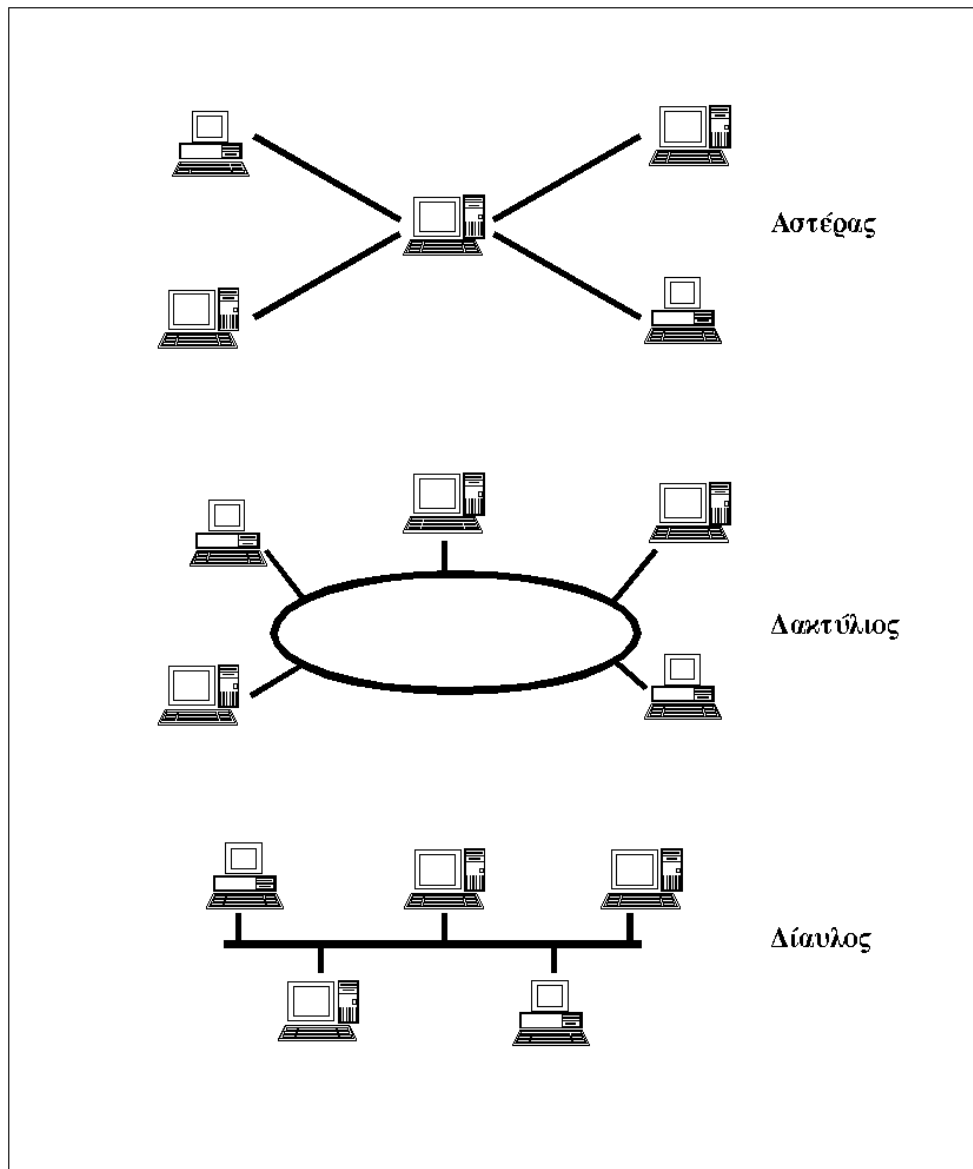
Αρχικά να σημειωθεί ότι το Internet είναι ανεξάρτητο του Hardware και στηρίζεται σε ένα πλαίσιο που περιγράφει μια σουίτα πρωτοκόλλων που χρησιμοποιούνται σε ένα ιεραρχικό μοντέλο, το οποίο “χωρίζει” την τελική επικοινωνία ανάμεσα σε συσκευές ή δίκτυα σε διάφορα επίπεδα. Η σουίτα αυτών των πρωτοκόλλων ονομάζεται TCP / IP και παίρνει το όνομά της από διασημότητα και χρησιμότητα δύο πολύ γνωστών πρωτοκόλλων της, του TCP και του IP. Στο σχήμα 1 βλέπουμε μια σύντομη οπτική παρουσίαση των σημαντικότερων πρωτοκόλλων ανά στρώμα.



Σχήμα 1: «Σουίτα πρωτοκόλλων του Internet»

Από την άλλη, στο φυσικό επίπεδο, ένα δίκτυο μπορεί να αποκτήσει διάφορες τοπολογικές μορφές. Οι πιο συνηθισμένες τοπολογίες είναι του αστέρα, του δακτυλίου και του δίαυλου. Στην τοπολογία αστέρα, η διαχείριση και η δρομολόγηση των πακέτων γίνεται από μια κεντρική συσκευή πάνω στην οποία συνδέονται οι υπόλοιπες συσκευές του δικτύου. Αξίζει να σημειωθεί ότι η δρομολόγηση μπορεί να επιτευχθεί και μέσω υλικού και μέσω λογισμικού. Στην τοπολογία δακτυλίου η δρομολόγηση γίνεται από συσκευή σε συσκευή, οι οποίες είναι συνδεδεμένες σε ένα κλειστό δακτύλιο. Τέλος, στην τοπολογία διαύλου, η μεταφορά των πακέτων γίνεται μέσω ενός κοινού μέσου επικοινωνίας πάνω στο οποίο οι συσκευές τοποθετούν και αποδέχονται τα πακέτα πληροφορίας που τους

αφορούν. Στο σχήμα 2 μπορείτε να δείτε την οπτική απεικόνιση των πιο συχνών τοπολογιών ενός δικτύου συσκευών.



Σχήμα 2: «Τοπολογίες δικτύου»

#### **1.4 Domain Name System**

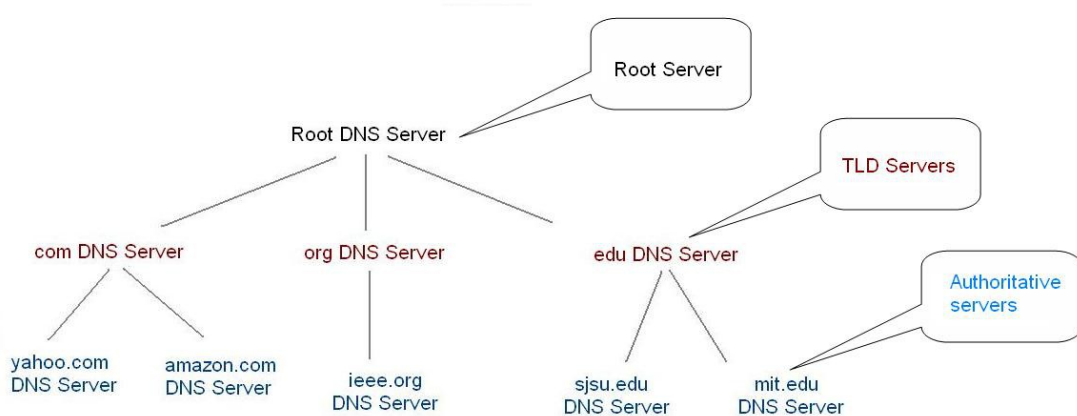
Όπως ένα διαμέρισμα χαρακτηρίζεται από μία διεύθυνση, έτσι και στο Internet σε κάθε διασυνδεδεμένη συσκευή αποδίδεται μία διεύθυνση αποτελούμενη από αριθμούς και ονομάζεται διεύθυνση IP (Κατά το πρωτόκολλο IP).

Όταν ένας χρήστης του Διαδικτύου θέλει να προσπελάσει ένα αρχείο το οποίο βρίσκεται σε ένα εξυπηρετητή ο οποίος είναι διασυνδεδεμένος στο Internet, πρέπει

να γνωρίζει τη διεύθυνσή του για να το εντοπίσει. Αυτό όμως αποτελεί δύσκολο εγχείρημα για τον ανθρώπινο εγκέφαλο, ο οποίος δυσκολεύεται να απομνημονεύει μεγάλες σειρές από αριθμούς.

Γιαυτό το λόγο, χρησιμοποιούμε το DNS (Domain Name System) το οποίο αποτελεί ένα σύστημα ονοματοδοσίας. Ρόλος του είναι η αντιστοίχιση διευθύνσεων IP σε ονόματα. Η δομή του είναι δενδρική και η διανομή του γίνεται μέσω βάσεων δεδομένων οι οποίες διατηρούν εγγραφές για τις αντιστοιχήσεις διευθύνσεων σε ονόματα, και οι βάσεις με τη σειρά τους διατηρούνται στους εξυπηρετητές.

Η διανεμημένη φύση του DNS του παρέχει μεγάλη ευκαμψία, καθώς σε περίπτωση μη λειτουργίας κάποιας βάσης, μπορεί να γίνει ενημέρωση από άλλες βάσεις που διατηρούν τις ίδιες εγγραφές.



Σχήμα 3: «Δενδρική δομή DNS»

## 1.5 IP και HTTP

Για τη μεταφορά δεδομένων από έναν εξυπηρετητή (Server) και ένα τελικό χρήστη (client), γίνεται η χρήση του πρωτοκόλλου HTTP, το οποίο υλοποιείται από τους φυλλομετρητές οι οποίοι χρησιμοποιούνται για την προσπέλαση του Παγκόσμιου Ιστού.

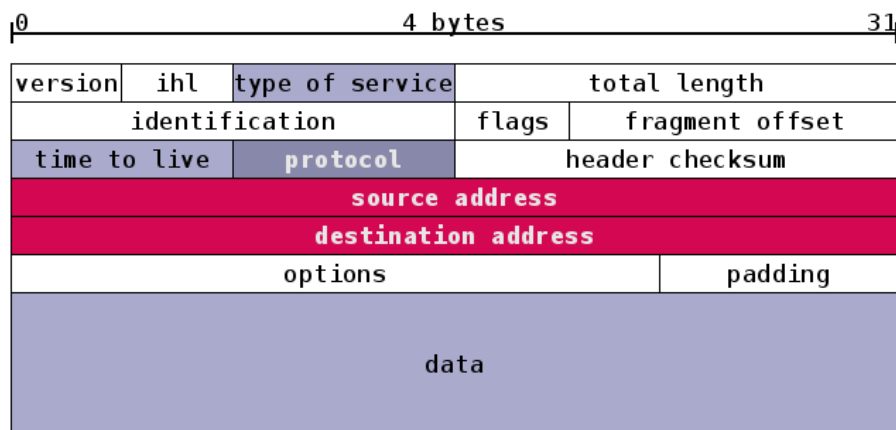
Σε τεχνικό επίπεδο, η διαδικασία που χρησιμοποιεί το πρωτόκολλο HTTP είναι η εξής:

- Σύνδεση στον εξυπηρετητή
- Ερώτημα προς τον εξυπηρετητή / Αποστολή δεδομένων προς τον εξυπηρετητή

- Απάντηση από τον εξυπηρετητή
- Αποσύνδεση

Καθώς το HTTP δεν παρέχει καμία ασφάλεια, γίνεται χρήση του πρωτοκόλλου SSL (Secure Socket Layer), το οποίο διασφαλίζει την ασφαλή μετάδοση δεδομένων από και προς τον εξυπηρετητή. Ο συνδυασμός των δύο αυτών πρωτοκόλλων συμβολίζεται με τους χαρακτήρες HTTPS.

Το πρωτόκολλο IP αποτελεί το βασικότερο και πιο διάσημο πρωτόκολλο επικοινωνίας στο Διαδίκτυο. Ανήκει στο επίπεδο του δικτύου και μεταφέρει την πληροφορία από τα κατώτερα επίπεδα υπό την μορφή πακέτων. Το κάθε IP πακέτο φθάνει στον παραλήπτη διασχίζοντας ένα ή περισσότερα διασυνδεδεμένα δίκτυα IP, χωρίς να βασίζεται στη μεταφορά και λειτουργία άλλων χρονικά μεταγενέστερων πακέτων, διατηρώντας έτσι την αυτονομία του στο δίκτυο. Το IP δεν ασχολείται με τις λειτουργίες των κατώτερων επιπέδων, αλλά με την διευθυνσιοδότηση (IP Addressing), τον κατακερματισμό (fragmentation) των μεγάλων σε όγκο πακέτων και την αναδόμησή τους στον προορισμό. Να σημειωθεί ότι το πρωτόκολλο IP δεν θεωρείται αξιόπιστο, δεδομένου ότι δεν εξασφαλίζει την απ'άκρη σ'άκρη ακεραιότητα των δεδομένων με τεχνικές επανεκπομπής ή ελέγχου ροής κλπ, σε αντίθεση με το πρωτόκολλο TCP. Στην εικόνα βλέπουμε τη δομή ενός πακέτου IP.



Σχήμα 4: «IP Header»

## 1.6 Μοντέλο πελάτη – εξυπηρετητή

Το μοντέλο πελάτη-εξυπηρετητή (client-server model) αποτελεί μια κυρίαρχη προσέγγιση στα δίκτυα υπολογιστών και εφαρμόζεται σε πάρα πολλές πτυχές του διαδικτύου όπως είναι η ηλεκτρονική αλληλογραφία, ο Παγκόσμιος Ιστός κτλ. Η



αρχιτεκτονική του μοντέλου περιλαμβάνει δύο μέρη. Το μέρος του πελάτη και το μέρος του εξυπηρετητή.

Η φύση του πελάτη μπορεί να είναι είτε λογισμική είτε υλική και αποκτά πρόσβαση σε μια υπηρεσία που μπορεί να παρέχει ένας εξυπηρετητής. Ο πελάτης στέλνει αιτήματα προς τον εξυπηρετητή και ο εξυπηρετητής ανταποκρίνεται. Πολύ συνοπτικά, υπάρχουν δύο ειδών μοντέλα πελάτη, το μοντέλο fat και το μοντέλο thin. Στο μοντέλο fat η επεξεργασία των δεδομένων και η διεκπεραίωση των διεργασιών γίνεται κυρίως στον πελάτη. Στο Thin μοντέλο οι προηγούμενες ενέργειες και ο φόρτος είναι υπευθυνότητα του εξυπηρετητή. Φυσικά, υπάρχουν και υβριδικές υλοποιήσεις που αξιοποιούν πλεονεκτήματα και από τα δύο μοντέλα.

Ο εξυπηρετητής από την άλλη αποτελεί και αυτός υλικό ή λογισμικό που έχει ως βασική λειτουργία την εξυπηρέτηση των αιτήσεων από διάφορους πελάτες. Η αρχιτεκτονική του υποστηρίζει τη δυνατότητα εξυπηρέτησης πολλών πελατών ταυτόχρονα και οι δυνατότητές του εξαρτώνται από τον τύπο του εξυπηρετητή και τις ρυθμίσεις του.

### **1.7 Ασφάλεια στο διαδίκτυο μέσω HTTPS**

Το HTTPS (Hypertext Transfer Protocol Secure) αποτελεί τον συνδυασμό των πρωτοκόλλων HTTP και SSL/ TLS. Ουσιαστικά παρέχει στο πρωτόκολλο HTTP την ασφάλεια και λειτουργικότητα του επιπέδου ασφαλείας που διαθέτει το πρωτόκολλο SSL/TLS.

Το HTTPS παρέχει αυθεντικοποίηση του Ιστότοπου και κατά συνέπεια του εξυπηρετητή, δίνοντας περαιτέρω τη δυνατότητα αμφίδρομης κρυπτογράφησης δεδομένων ανάμεσα στο χρήστη και τον εξυπηρετητή. Αυτό το επίπεδο ασφαλείας έχει ως αποτέλεσμα την πολύ δυσκολότερη υποκλοπή ή και τροποποίηση των δεδομένων από τρίτους, λόγω της κρυπτογράφησης των δεδομένων. Αρχικά το HTTPS χρησιμοποιήθηκε κυρίως για τις online αγορές στον Παγκόσμιο Ιστό καθώς και την ασφαλή ανταλλαγή ηλεκτρονικής αλληλογραφίας σε επιχειρησιακά περιβάλλοντα, αλλά αργότερα εξαπλώθηκε σε καθημερινή χρήση από τον μέσο χρήστη, για προβολή ιστοσελίδων και ανταλλαγή δεδομένων μέσω του πρωτοκόλλου HTTP.

### **1.8 Επίλογος**

Καθώς ένας Ιστότοπος “φιλοξενείται” στο διαδίκτυο, έγινε μια σύντομη αναφορά στην τεχνολογία που κρύβεται πίσω από το διαδίκτυο καθώς και σε γενικά χαρακτηριστικά της δομής του. Είδαμε αρχιτεκτονικές όπως το μοντέλο του πελάτη-εξυπηρετητή καθώς και τη χρήση του πρωτοκόλλου SSL για την ασφαλή

περιήγηση μέσω του πρωτοκόλλου HTTP. Στο επόμενο κεφάλαιο θα παραθετούν περισσότερες λεπτομέρειες σχετικά με την τεχνική φύση ενός Ιστότοπου, αλλά θα γίνει και αναφορά πάνω στα διάφορα είδη εξυπηρετητών.

## 2 Ιστότοποι και εξυπηρετητές

### 2.1 Τεχνολογία και χαρακτηριστικά ενός Ιστότοπου

Για την ολοκληρωμένη λειτουργία ενός Ιστότοπου απαιτούνται τουλάχιστον τα εξής:

- Χώρος σε κάποιον εξυπηρετητή. Καθώς ένας Ιστότοπος αποτελείται από αρχεία τα οποία μπορεί να προσπελάσει ο χρήστης, τα αρχεία αυτά πρέπει να βρίσκονται σε κάποια συσκευή η οποία να μπορεί να λειτουργήσει ως Web Server και να υποστηρίξει το πρωτόκολλο HTTP.
- Όνομα Χώρου (Domain Name). Όπως προαναφέρθηκε στο πρώτο κεφάλαιο, ο ανθρώπινος εγκέφαλος δεν είναι ικανός να θυμάται εύκολα μια διεύθυνση IP. Γι' αυτό το λόγο, χρησιμοποιώντας το DNS, γίνεται αντιστοίχιση μιας διεύθυνσης IP σε ένα όνομα χώρου. Αντίστοιχα, ένα όνομα χώρου “βλέπει” προς έναν εξυπηρετητή, και κατά συνέπεια τα αρχεία του.
- Υποστήριξη βάσεων δεδομένων από τον εξυπηρετητή. Στις περισσότερες περιπτώσεις υπάρχει ξεχωριστός εξυπηρετητής για τις βάσεις δεδομένων. Βάση δεδομένων είναι μια συλλογή από εγγραφές / δεδομένα, η οποία διαχειρίζεται από κάποιο Σύστημα Διαχείρισης Βάσεων δεδομένων και μπορεί να προσπελαθεί χρησιμοποιώντας μια γλώσσα ερωταπαντήσεων, όπως για παράδειγμα την SQL.
- Υποστήριξη γλωσσών προγραμματισμού εξυπηρετητή. Χρησιμοποιώντας μια τέτοια γλώσσα, η εκτέλεση των εντολών γίνεται στον εξυπηρετητή και στον χρήστη επιστρέφονται μόνο τα αποτελέσματα. Η PHP αποτελεί μια διάσημη server-side γλώσσα προγραμματισμού.

### 2.2 Ονόματα χώρου

Το όνομα χώρου ή τομέα ή περιοχής (αγγλ. domain name) είναι η αναπαράσταση με συμβολοσειρές μιας διεύθυνσης IP και οριοθετείται γύρω από τους κανόνες που ορίζει το DNS. Να σημειωθεί ότι το όνομα χώρου δεν χρησιμοποιείται αποκλειστικά για την αντιστοίχιση σε ιστοσελίδες, αλλά μπορεί να αντιστοιχηθεί με οποιοδήποτε πόρο που μπορεί να διαθέτει διεύθυνση IP, όπως είναι μία υπηρεσία, ένας server, κάποιο socket κτλ.

Είναι ένας περιορισμένος τομέας των διεθνών πόρων του Συστήματος Ονομάτων Χώρου (DNS) ο οποίος εκχωρείται για αποκλειστική χρήση σε ένα φυσικό ή νομικό πρόσωπο. Το όνομα τομέα / χώρου δεν ανήκει στο πρόσωπο που του έχει

εκχωρηθεί αλλά έχει μόνο την αποκλειστική δυνατότητα χρήσης του για όσο διάστημα έχει καταβάλει τα τέλη κατοχύρωσης. Ένα όνομα χώρου μπορεί να έχει διάφορες καταλήξεις όπως .com, .eu, .gr, .net, .org, .info, .biz, .de, .it, .es κ.λ.π., ανάλογα με τη χρήση και τη χώρα προέλευσής του.

Στα ονόματα χώρου επιτρέπεται μόνο η χρήση αλφαριθμητικών στοιχείων και παυλών. Για τα ονόματα χώρου με κατάληξη .gr υπάρχουν απαγορευμένες κατηγορίες. Αν ένα όνομα χώρου θεωρείται κοινόχρηστο ή γεωγραφικός όρος εκχωρείται μόνο στους αντίστοιχους οργανισμούς τοπικής αυτοδιοίκησης ανεξάρτητα από τον τρόπο γραφής του με λατινικά στοιχεία. Επίσης δεν επιτρέπεται η εκχώρηση ονομάτων χώρου με κατάληξη .gr που αποτελούν λέξεις κλειδιά στο Διαδίκτυο.

Τα κατοχυρωμένα ονόματα χώρου είναι συνήθως τα ονόματα των τριών ή τεσσάρων πρώτων επιπέδων. Τα υπόλοιπα ονόματα χώρου δεν χρειάζονται κατοχύρωση. Στα ονόματα χώρου κάθε τελεία δείχνει την αλλαγή επιπέδου ή αρχή ενός υποσυνόλου - υποτομέα και το σύνολο - χώρος που περιλαμβάνει όλα τα σύνολα είναι η πιο δεξιά τελεία που συνήθως παραλείπεται. Οι λύτες είναι το λογισμικό που μας βοηθά να χρησιμοποιήσουμε τα ονόματα χώρου. Οι λύτες διαβάζουν τα ονόματα του DNS από δεξιά προς τα αριστερά.

Για παράδειγμα όταν γράφουμε το όνομα «DNS.example.wikipedia.www.el.ipduh.com» εννοούμε «DNS.example.wikipedia.www.el.ipduh.com.». Η τελική τελεία είναι το σύνολο που περιλαμβάνει όλο το σύστημα και το υποσύνολο που ονομάζεται «com.». Το σύνολο «com.» περιλαμβάνει το σύνολο «ipduh.com.», το σύνολο «ipduh.com.» περιλαμβάνει το «el.ipduh.com.», το σύνολο «el.ipduh.com.» περιλαμβάνει το σύνολο «www.el.ipduh.com.» κ.ο.κ.

### **2.3 Web servers: εξυπηρετώντας τα δεδομένα**

Η εξυπηρέτηση δεδομένων γίνεται συνήθως μέσω εξυπηρετητών Ιστού. Επειδή η εγκατάσταση και διαχείριση ενός εξυπηρετητή είναι για τον μέσο χρήστη δύσκολη και χρονοβόρα, η πιο συνηθισμένη λύση είναι η ενοικίαση εξυπηρετητών από εταιρίες.

Εξυπηρετητής ή διακομιστής (αγγλ.: server) είναι υλικό ή / και λογισμικό που αναλαμβάνει την παροχή διάφορων υπηρεσιών, «εξυπηρετώντας» αιτήσεις άλλων προγραμμάτων, γνωστούς ως πελάτες (clients) που μπορούν να τρέχουν στον ίδιο υπολογιστή ή σε σύνδεση μέσω δικτύου. Όταν ένας υπολογιστής εκτελεί κυρίως τέτοια προγράμματα εξυπηρετητές συνεχόμενα, 24 ώρες την ημέρα, τότε μπορούμε να αναφερθούμε σε όλον τον υπολογιστή ως εξυπηρετητή, αφού αυτή

είναι η κύρια λειτουργία του. Παρομοίως, ως πελάτη μπορούμε να θεωρήσουμε είτε κάποιο λογισμικό που επικοινωνεί και υποβάλει αιτήματα στον εξυπηρετητή, είτε σε όλο τον υπολογιστή όταν ο εξυπηρετητής είναι άλλος υπολογιστής και οι 2 υπολογιστές είναι συνδεδεμένοι σε ένα δίκτυο.

Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή γίνεται μέσω ενός τοπικού δικτύου, ή ακόμα και μέσω του Διαδικτύου. σε μεγάλα δίκτυα όπου ο εξυπηρετητής αναλαμβάνει πολλές εξυπηρετήσεις είναι συνήθως υπολογιστής που διαφέρει ως προς τη σύνθεσή του από άλλους κοινούς υπολογιστές, μιας και οι δυνατότητες του είναι σαφώς αναβαθμισμένες. Κύρια χαρακτηριστικά ενός εξυπηρετητή είναι οι επεξεργαστές που υποστηρίζει και χρησιμοποιεί για την επεξεργασία των δεδομένων που δέχεται, οι γρήγοροι και μεγάλης χωρητικότητας σκληροί δίσκοι αλλά και οι ταχύτατες μνήμες που υποστηρίζει. Συνήθως συνοδεύεται από σύστημα διπλής τροφοδοσίας (dual power supply) και από συσκευή αδιάλειπτης παροχής ενέργειας (UPS), για μεγαλύτερη αξιοπιστία και σιγουριά στις παρεχόμενες υπηρεσίες του.

Υπάρχουν εταιρίες που παρέχουν στους χρήστες της δυνατότητες hosting δωρεάν, και άλλες επί πληρωμής. Η επιλογή ενός host provider δεν είναι εύκολη υπόθεση και συνήθως χρειάζεται αρκετή έρευνα για την κατάλληλη επιλογή. Οι παράμετροι που μπορούν να επηρεάσουν την απόφαση του χρήστη είναι αρκετές, οι πιο σημαντικοί εκ των οποίων είναι:

- Πόσο αποθηκευτικό χώρο χρειάζεται;
- Πόσες βάσεις δεδομένων θα χρειαστεί;
- Προβλέπεται να έχει μεγάλη ποσότητα διακίνησης πληροφορίας; (περιορισμός στο bandwidth)
- Η ασφάλεια είναι σημαντικό κομμάτι;
- Πόσα ονόματα χώρου θα χρειαστεί;
- Χρειάζεται πρόσβαση σε SSH?
- Τι είδους λειτουργικό σύστημα επιθυμεί;

κτλ.

### **2.3.1 Το υλικό των web servers**

Οι υπολογιστές που χρησιμοποιούνται ως εξυπηρετητές δικτύου πρέπει να μπορούν να δουλεύουν όλη την ημέρα και συνεχώς χωρίς διακοπές. Για το λόγο

αυτό οι εταιρείες κατασκευής υπολογιστών και τμημάτων υπολογιστών προσφέρουν ξεχωριστές κατασκευές για εξυπηρετητές. Φυσικά αυτές οι κατασκευές μπορούν να χρησιμοποιηθούν και για προσωπικό υπολογιστή όμως ανεβάζουν το κόστος.

Συνηθισμένα χαρακτηριστικά για το υλικό είναι:

#### **μητρικές πλακέτες με**

- μεγάλη αντοχή στις θερμοκρασίες
- ενσωματωμένα συστήματα RAID
- ενσωματωμένη κάρτα οθόνης μικρής μνήμης και ταχύτητας
- πιο άνετα κατανεμημένα τα στοιχεία της μνήμης και του επεξεργαστή ώστε να είναι δυνατή η ψύξη και των δύο από ξεχωριστά δυνατά ανεμιστηράκια
- λιγότερες θύρες επέκτασης
- περισσότερα ενδεικτικά στοιχεία λειτουργίας
- δυνατότητα χρήσης μνήμης ECC

#### **κουτιά**

- βαριά
- με ειδικό κλείδωμα
- με θέσεις για περισσότερα ανεμιστηράκια
- με ειδικά διαμορφωμένους εσωτερικούς χώρους ώστε να γίνεται πιο καλή κυκλοφορία του αέρα
- ειδικά κουτιά που μπορούν να ενσωματωθούν σε rack

#### **δίσκοι**

- με προδιαγραφές για μεγαλύτερο συνεχόμενο χρόνο χρήσης
- που μπορούν να αποσυνδέονται την ώρα που είναι σε λειτουργία

### **2.3.2 Λογισμικό των web servers**

#### **Λειτουργικό σύστημα**

Τα περισσότερα λειτουργικά συστήματα έχουν ειδική έκδοση για χρήση ως εξυπηρετητή ή μπορούν να διαμορφωθούν έτσι εκ των υστέρων εκτελώντας λογισμικό που κάνει την εξυπηρέτηση.

Οι περισσότεροι εξυπηρετητές του διαδικτύου αναλαμβάνουν πολύ δουλειά και είναι κατάλληλα οργανωμένοι με χαρακτηριστικά που τους διαφοροποιούν από απλούς εξυπηρετητές από τα οποία είναι:

#### **Δυνατότητα λειτουργίας χωρίς την ύπαρξη:**

- πληκτρολογίου
- οθόνης
- γραφικού περιβάλλοντος
- κάρτας ήχου
- δυνατότητες επιλογής χρόνου επεξεργασίας των διάφορων προγραμμάτων
- πρόγραμμα επικοινωνίας με το UPS
- δυνατότητα χρήσης περισσότερων του ενός επεξεργαστή
- δυνατότητα συνεπεξεργασίας με άλλους υπολογιστές- εξυπηρετητές
- περισσότερες δυνατότητες αλλαγής του υλικού και αναβάθμισης του λειτουργικού χωρίς την ανάγκη επανεκκίνησης του συστήματος
- περισσότερα συστήματα ασφαλείας

#### **Προγράμματα**

Πολλά σύγχρονα προγράμματα δουλεύουν με τη λογική πελάτη - εξυπηρετητή. Ακόμα και το ίδιο το λειτουργικό σύστημα δουλεύει με αυτήν τη λογική. Τα προγράμματα ζητάνε κάποια ενέργεια και το λειτουργικό σύστημα αναλαμβάνει να τα εξυπηρετήσει εκτελώντας τις λειτουργίες που του ζητήθηκαν.

Συνήθως τα περισσότερα προγράμματα εξυπηρετητών απαιτούν και ξεχωριστό υπολογιστή-εξυπηρετητή χωρίς αυτό να είναι πάντα απαραίτητο.

Συνηθισμένοι εξυπηρετητές-προγράμματα σε περιβάλλον γραφείου που μπορεί να εκτελούνται στον ίδιο ή σε ξεχωριστούς υπολογιστές είναι:

- Εξυπηρετητής αρχείων (file server)
- Εξυπηρετητής εκτυπωτών (printer server)
- Εξυπηρετητής αντιγράφων ασφαλείας (backup server)
- Εξυπηρετητής βάσεων δεδομένων (database server)
- Εξυπηρετητής φαξ (fax server)
- Εξυπηρετητής διαμεσολαβητή (proxy server)
- Εξυπηρετητής ηλεκτρονικού ταχυδρομείου (mail server)

- Εξυπηρετητής ήχου (sound server)
- Εξυπηρετητής γραφικής απεικόνισης

Μερικοί εξυπηρετητές όπως εκτυπώσεων, ήχου, γραφικής διεπαφής θεωρούνται αυτονόητο ότι εξυπηρετούν τον ίδιο υπολογιστή, όμως αυτό δεν είναι απαραίτητο, για παράδειγμα ένας υπολογιστής μπορεί να μην εκτυπώνει στον δικό του εκτυπωτή αλλά να τις εκτυπώνει σε άλλο υπολογιστή του δικτύου, το ίδιο μπορεί να συμβαίνει και με τον εξυπηρετητή που απεικονίζει γραφικές διεπαφές, θα ήταν δυνατό να απεικονίζει τα προγράμματα σε μία τελείως διαφορετική οθόνη ενός άλλου υπολογιστή του ίδιου δικτύου.

Συνηθισμένοι εξυπηρετητές-προγράμματα στο Ίντερνετ είναι:

- Παγκόσμιου Ιστού με το πρωτόκολλο http (http server)
- Domain Name System (DNS server)
- Ηλεκτρονικού ταχυδρομείου (mail server)
- Μεταφοράς αρχείων με το πρωτόκολλο FTP (ftp server)
- IRC (Internet Relay Chat) και instant messaging (IRC server)
- Επικοινωνίες φωνής
- streaming audio και video (streaming server)
- Online παιχνίδια

Οι υπηρεσίες hosting είναι πλέον στην εποχή μας τόσο απαραίτητες, που πολλές φορές προσφέρονται και ως δώρο σε αγορά άλλων υπηρεσιών, όπως για παράδειγμα ενός ονόματος χώρου. Πολλοί Registrars δίνουν τη δυνατότητα hosting στους πελάτες τους για τα ονόματα χώρου που αγόρασαν.

Να σημειωθεί πως τα πακέτα hosting που συνήθως προσφέρονται, μπορεί να ξεκινάνε από δωρεάν και να καταλήγουν σε διάφορα ποσά ανά μήνα, ή έτος, ανάλογα με την λειτουργικότητα που προσφέρει το κάθε πακέτο.

## **2.4 Εξυπηρετητές βάσεων δεδομένων**

Στην πράξη, η διαχείριση κάποιου συστήματος διαχείρισης βάσης δεδομένων μπορεί να γίνει και από τον εξυπηρετητή ιστού, με το κατάλληλο λογισμικό, παρόλα αυτά, σε μεγάλα production συστήματα που έχουν μεγάλες απαιτήσεις σε πόρους και αποθηκεύουν τεράστιους όγκους πληροφορίας σε βάσεις δεδομένων, η χρήση ανεξάρτητων εξυπηρετητών βάσεων δεδομένων είναι απαραίτητη.



Συνήθως τα συστήματα διαχείρισης βάσεων δεδομένων (DBMS - Database Management Systems) παρέχουν λειτουργικότητα εξυπηρετητών από μόνα τους. Ορισμένα παραδείγματα από υλοποιημένους εξυπηρετητές βάσεων δεδομένων είναι η Oracle, η DB2 και ο Microsoft SQL Server.

## **2.5 Nginx, apache και Windows Server**

Ο Apache HTTP Server είναι ένα πακέτο λογισμικού με λειτουργικότητα εξυπηρετητή Ιστού, ο οποίος έπαιξε πολύ σημαντικό ρόλο στην ανάπτυξη του Παγκόσμιου Ιστού καθώς αποτέλεσε μία πολύ σταθερή και πολυχρησιμοποιημένη λύση. Η γλώσσα πάνω στην οποία είναι γραμμένος είναι η C, τρέχει σε συστήματα τύπου Unix και και σήμερα συνεχίζει να αναπτύσσεται και να υποστηρίζεται από το Apache Software Foundation. Διατίθεται με μηδαμινό κόστος και υποστηρίζεται από την κοινότητα ανοικτού λογισμικού. Η λειτουργικότητα του Apache παρέχεται μέσω των δομημένων μονάδων που διαθέτει, οι οποίες επεκτείνουν την βασική του λειτουργικότητα παρέχοντάς του εξτρά λειτουργίες.

Η Microsoft από την πλευρά της διαθέτει τα Windows Server, το οποίο περιλαμβάνει ένα μεγάλο πακέτο από εφαρμογές - εξυπηρετητές όπως είναι για παράδειγμα τα Windows 2000 Server και τα Windows Server 2008.

Ο Nginx, ο οποίος διαβάζεται ως “engine-x” είναι ένας server που ανήκει στην κατηγορία ανοικτού λογισμικού, reverse proxy (ο server αποκτά τα δεδομένα για τον πελάτη από πολλές πηγές / servers) για τα πρωτόκολλα HTTP, HTTPS, SMTP, POP3 και IMAP, αλλά λειτουργεί επίσης και ως ισορροπιστής φόρτου, και HTTP cache. Το nginx project ξεκίνησε με βασική έμφαση στον υψηλό συγχρονισμό (high concurrency), υψηλή απόδοση και χαμηλή χρήση της μνήμης. Η άδεια χρήσης του είναι η BSD και μπορεί να τρέξει σε Linux, διάφορους κλώνους του BSD, Mac OS X, Solaris, AIX, HP-UX καθώς και σε διάφορα άλλα.

## **2.6 Συστήματα διαχείρισης βάσεων δεδομένων**

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι ένα σύνολο προγραμμάτων που επιτρέπει λειτουργίες όπως αποθήκευση, τροποποίηση και απόσπαση πληροφοριών από μία βάση δεδομένων (Οι βάσεις δεδομένων είναι κοινά αρχεία σε υπολογιστή, δομημένα σε εγγραφές). Παρέχει επίσης μεγάλη πληθώρα λειτουργικότητας στους χρήστες του συστήματος για την τροποποίηση και προσθήκη πληροφοριών σε οποιοδήποτε σημείο της βάσης. Οι σύγχρονες βάσεις δεδομένων παρέχουν τα κατάλληλα εργαλεία ώστε να επιτυγχάνεται η εύκολη πρόσβαση στα δεδομένα, ακεραιότητα των δεδομένων, διαχείριση ταυτοχρονισμού (διαχείριση περιπτώσεων ταυτόχρονης πρόσβασης των χρηστών στα δεδομένα)

και επανάκτησης πληροφοριών από τη βάση, σε περίπτωση σφαλμάτων, μέσω αντιγράφων ασφαλείας.

Ο πιο συνηθισμένος τύπος Συστημάτων Διαχείρισης Βάσεων Δεδομένων είναι ο σχεσιακός, ο οποίος έχει επικρατήσει και στηρίζεται στη σχεσιακή άλγεβρα.

Οι στόχοι μιας βάσης δεδομένων(και κατ' επέκταση του συστήματος διαχείρισης) είναι οι εξής :

- Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
- Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
- Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.
- Η επιβολή κανόνων ασφαλείας (security).
- Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
- Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται (οργανώνονται) στις εξής στοιχειώδεις μορφές :

- Πεδίο (Field), είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.
- Εγγραφή (Record), είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.
- Αρχείο (File), είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.
- Πρωτεύον Κλειδί (Primary Key), είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.
- Κλειδί (Key), είναι ένα πεδίο που δεν έχει κατ' ανάγκη μοναδική τιμή και που μπορούμε να το χρησιμοποιήσουμε για να κάνουμε αναζήτηση σ' ένα αρχείο.

- Ξένο Κλειδί (Foreign Key), είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου αρχείου.

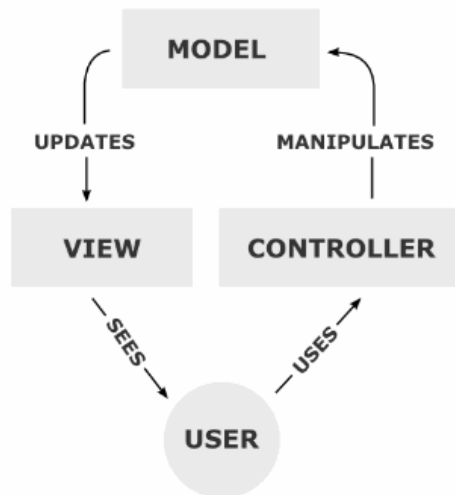
Τα εργαλεία χειρισμού πληροφοριών μιας βάσης δεδομένων είναι γνωστά και σαν "Γλώσσες Εντολών" και με τη βοήθειά τους μπορούμε να δώσουμε εντολές χειρισμού των δεδομένων. Η πιο γνωστή και ευρέως διαδεδομένη γλώσσα εντολών για τις σύγχρονες βάσεις δεδομένων είναι η Δομημένη Γλώσσα Ερωτήσεων SQL (Structured Query Language), η οποία αποτελείται από τα εξής μέρη:

- DDL (Data Definition Language, Γλώσσα Ορισμού Δεδομένων), με την οποία καθορίζουμε τις δομές και τα τμήματα μιας βάσης δεδομένων.
- DML (Data Manipulation Language, Γλώσσα Χειρισμού Δεδομένων), με την οποία επεξεργαζόμαστε τα δεδομένα μιας βάσης δεδομένων.
- DCL (Data Control Language, Γλώσσα Ελέγχου Δεδομένων), με την οποία εξασφαλίζουμε την ασφάλεια και την ακεραιότητα των δεδομένων μιας βάσης δεδομένων.

## **2.7 Αρχιτεκτονική MVC**

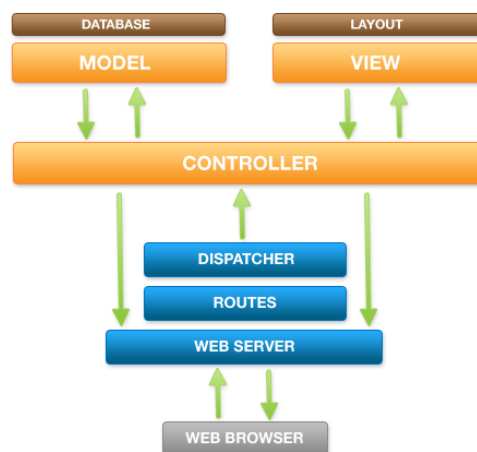
Η MVC (Model-View-Controller) αποτελεί μια αρχιτεκτονική λογισμικού για εφαρμογές (αρχικά υλοποιήθηκε για γενικές εφαρμογές και αργότερα υιοθετήθηκε και για στις διαδικτυακές εφαρμογές) που χωρίζει την αναπαράσταση (Viewing) της πληροφορίας από την αλληλεπίδραση του χρήστη με αυτή. Πιο συγκεκριμένα, το μοντέλο (model) περιλαμβάνει τα δεδομένα της εφαρμογής και τους τυχόν κανόνες διαχείρισής τους και ο ελεγκτής (controller) είναι ο διαμεσολαβητής, λαμβάνοντας τα δεδομένα και τροποποιώντας τα ώστε να μπορούν να χρησιμοποιηθούν είτε από άλλα μοντέλα ή από κάποια προβολή. Η προβολή μπορεί να είναι η παραμετροποιημένη έξοδος των δεδομένων που επεξεργάστηκαν από τον ελεγκτή. Γίνεται φανερό ότι μπορούμε να έχουμε πολλαπλές προβολές των ίδιων δεδομένων.

Με την αρχιτεκτονική MVC επιτυγχάνεται αυξημένη δυνατότητα επαναχρησιμοποίησης κώδικα και διαχωρισμό του προβλήματος σε μικρότερα και πιο εύκολα διαχειρίσιμα υποπροβλήματα.



Σχήμα 5: «Αρχιτεκτονική MVC»

#### JOOMLA MODEL VIEW CONTROLLER (MVC)



Σχήμα 6: «Αρχιτεκτονική MVC στο Joomla»

## 2.8 Μοντέλο REST

Το μοντέλο Representational state transfer (REST) είναι μία αφαίρεση της αρχιτεκτονικής που περιγράφει τον Παγκόσμιο Ιστό (WWW, World Wide Web). Πιο λεπτομερειακά, το μοντέλο REST είναι ένα αρχιτεκτονικό στυλ το οποίο αποτελείται από ένα καλώς συντονισμό σετ από αρχιτεκτονικούς περιορισμούς που εφαρμόζονται στα πακέτα λογισμικού, στους connectors και στα στοιχεία που απαρτίζουν τα δεδομένα, εντός ενός κατανενημένου συστήματος πολυμέσων. Το μοντέλο REST αγνοεί τις λεπτομέρειες υλοποίησης των πακέτων λογισμικού και της αντίστοιχης σύνταξης των πρωτοκόλλων έτσι ώστε να επικεντρωθεί στους

ρόλους των πακέτων / εξαρτημάτων, των περιορισμών τους σε σχέση με την αλληλεπίδραση που έχουν με τα υπόλοιπα πακέτα καθώς και στην ερμηνεία σημαντικών στοιχείων δεδομένων.

Ο όρος *representational state transfer* πρωτάθηκε και ορίστηκε για πρώτη φορά το 2000 από τον Roy Fielding στην πτυχιακή διατριβή του διδακτορικού του στο πανεπιστήμιο UC Irvine. Το μοντέλο REST έχει εφαρμοστεί για να περιγράψει επιθυμητή αρχιτεκτονική του Διαδικτύου, να αναγνωρίσει προβλήματα που υπάρχουν ήδη στις υπάρχων αρχιτεκτονικές, να συγκρίνει εναλλακτικές λύσεις και να εξασφαλίσει ότι διάφορες επεκτάσεις πρωτοκόλλων δεν θα παραβιάζουν τους βασικότερους περιορισμούς που καθιστούν το Παγκόσμιο Ιστό επιτυχή. Ο Fielding χρησιμοποίησε το μοντέλο REST για να σχεδιάσει το HTTP 1.1 και την έννοια URI (Uniform Resource Identifiers)

Το αρχιτεκτονικό στυλ του μοντέλου REST εφαρμόζεται επίσης στην ανάπτυξη και προγραμματισμό διαδικτυακών υπηρεσιών ως μία εναλλακτική λύση σε άλλες προδιαγραφές κατανεμημένων συστημάτων, όπως είναι για παράδειγμα το SOAP (Simple Object Access protocol). Κάποιος θα μπορούσε να χαρακτηρίσει τις διαδικτυακές υπηρεσίες ως “RESTful”, εάν συμμορφώνονται στους περιορισμούς που περιγράφονται στον αντίστοιχο τομέα στο REST που περιγράφει τους αρχιτεκτονικούς περιορισμούς.

Το αρχιτεκτονικό στυλ REST αναπτύχθηκε από το W3C Technical Architecture Group (TAG) παράλληλα με το HTTP 1.1, το οποίο στηρίχθηκε στο ήδη υπάρχον σχέδιο του HTTP 1.0. Ο Παγκόσμιος Ιστός αντιπροσωπεύει μία μεγαλύτερη υλοποίηση ενός συστήματος το οποίο συμμορφώνεται στο αρχιτεκτονικό στυλ που ορίζει το μοντέλο REST.

## **2.9 Αρχιτεκτονική λογισμικού στο μοντέλο REST**

Μία τυπική αρχιτεκτονική λογισμικού περιγράφεται και ορίζεται από το σύνολο των ρυθμίσεων ανάμεσα στα πακέτα / εξαρτήματα λογισμικού, τους connectors και τα δεδομένα που περιορίζονται ανάλογα τις σχέσεις που υπάρχουν μεταξύ τους ώστε να επιτευχθεί ένα επιθυμητό σετ από αρχιτεκτονικές ιδιότητες.

### **Πακέτα / εξαρτήματα λογισμικού (Components)**

Ένα εξάρτημα / πακέτο λογισμικού είναι μία αφηρημένη μονάδα από οδηγίες λογισμικού καθώς και εσωτερικής κατάστασης του συγκεκριμένου πακέτου, και η οποία μονάδα παρέχει μετασχηματισμούς των δεδομένων μέσω της διεπαφής του.

## Connectors

Ένας connector είναι ένας αφηρημένος μηχανισμός ο οποίος ρυθμίζει την επικοινωνία, τον συντονισμό και την συνεργασία ανάμεσα στα πακέτα λογισμικού τα οποία αναφέρθηκαν παραπάνω.

## Data

Τα δεδομένα είναι στοιχεία επικοινωνίας τα οποία μεταφέρονται από ένα πακέτο λογισμικού, ή λαμβάνονται από ένα πακέτο λογισμικού, μέσω ενός connector.

### 2.9.1 Ιδιότητες της αρχιτεκτονικής του μοντέλου REST

Οι αρχιτεκτονικές ιδιότητες που απορρέουν από τους αρχιτεκτονικούς περιορισμούς που προκύπτουν από το αρχιτεκτονικό στυλ του REST, είναι:

- Απόδοση - Οι αλληλεπιδράσεις ανάμεσα στα πακέτα λογισμικού παίζουν πολύ σημαντικό και πρωταρχικό ρόλο στην απόδοση και αποτελεσματικότητα δικτύου που αντιλαμβάνεται ο τελικός χρήστης.
- Επεκτασιμότητα, ώστε να υποστηριχθεί ένας μεγάλος αριθμός από πακέτα λογισμικού καθώς και οι αλληλεπιδράσεις ανάμεσα σε αυτά τα πακέτα.

Ο Fielding περιγράφει τις ιδιότητες του REST μοντέλου σε σχέση με την επεκτασιμότητα:

Ο διαχωρισμός του πελάτη με τον εξυπηρετητή στο μοντέλο REST απλοποιεί την υλοποίηση των εξαρτημάτων, μειώνει την πολυπλοκότητα των σημασιολογικών συνδέσεων, βελτιώνει την αποτελεσματικότητα της ρύθμισης της απόδοσης και βελτιώνει την επεκτασιμότητα των εξαρτημάτων που αφορούν αποκλειστικά τον server. Η λογική και ο διαχωρισμός της εφαρμογής σε επίπεδα, επιτρέπει την εύκολη είσοδο και εγκατάσταση διαφόρων proxies, gateways και firewalls, σε διάφορα σημεία της επικοινωνίας χωρίς να αλλάζουν οι διεπαφές ανάμεσα στα πακέτα λογισμικού, με αποτέλεσμα να είναι εύκολη η μετάφραση επικοινωνίας που αφορά το υλικό καθώς και η βελτίωση της απόδοσης του συστήματος μέσω μεγάλης κλίμακας κοινόχρηστου caching. Το μοντέλο REST επιτρέπει την ενδιάμεση επεξεργασία καθώς περιορίζει τα μηνύματα ώστε να περιγράφονται από μόνα τους: η επικοινωνία είναι stateless ανάμεσα στα αιτήματα, καθώς επίσης χρησιμοποιούνται βασικές μέθοδοι και τύποι πολυμέσων για να προσδιορίζεται η σημασιολογία και η πληροφορία που ανταλλάσσεται. Οι απαντήσεις στα αιτήματα ορίζονται και διαμορφώνονται με τέτοιο τρόπο ώστε να είναι κατάλληλες για caching.

- Απλότητα των διεπαφών

- Ύπαρξη της δυνατότητας επεξεργασίας των πακέτων λογισμικού, ώστε να προσαρμόζονται στις αλλαγές που προκύπτουν στις διάφορες ανάγκες (ακόμη και εάν η εφαρμογή εκτελείται)
- Ορατότητα στην επικοινωνία ανάμεσα στα πακέτα λογισμικού που χρησιμοποιούν οι διάφοροι πράκτορες υπηρεσιών
- Μεταφερσιμότητα των πακέτων λογισμικού μεταφέροντας κώδικα του προγράμματος που διαθέτει δεδομένα
- Αξιοπιστία ως προς την ανθεκτικότητα σε αποτυχίες στο επίπεδο συστήματος, ενώ υπάρχουν αποτυχίες μέσα στα πακέτα λογισμικού, στους connectors αλλά ακόμη και στα ίδια τα δεδομένα.

## **2.10 Frameworks και CMS**

Στον προγραμματισμό ηλεκτρονικών υπολογιστών, ένα framework λογισμικού είναι μία αφαίρεση στην οποία λογισμικό το οποίο παρέχει γενικευμένη λειτουργικότητα, μπορεί επιλεκτικά να αλλάξει από επιπρόσθετο κώδικα ο οποίος έχει γραφτεί από τον χρήστη / προγραμματιστή, άρα και να παρέχεται λογισμικό το οποίο είναι ειδικό ως προς την εφαρμογή. Ένα framework λογισμικού μπορεί επίσης να οριστεί ως μία καθολική, επαναχρησιμοποιήσιμη πλατφόρμα λογισμικού η οποία χρησιμοποιείται για την ανάπτυξη εφαρμογών λογισμικού, προϊόντων και λύσεων. Τα frameworks λογισμικού περιλαμβάνουν διάφορα προγράμματα υποστήριξης, μεταγλωττιστές, βιβλιοθήκες κώδικα, σετ από εργαλεία καθώς και διάφορα APIs (Application Programming Interfaces) τα οποία ενωποιούν ουσιαστικά όλα τα διαφορετικά πακέτα λογισμικού ώστε να επιτρέψουν τον προγραμματισμό μίας λύσης ή ενός project.

### **2.10.1 Διαφορές ανάμεσα στα frameworks και τις κοινές βιβλιοθήκες**

Τα frameworks λογισμικού διαθέτουν ένα σετ από ορισμένα χαρακτηριστικά τα οποία τα διαφοροποιούν από τις κοινές βιβλιοθήκες λογισμικού:

- Αντιστροφή του ελέγχου (Inversion of control): Σε ένα framework λογισμικού, αντίθετα με τις βιβλιοθήκες λογισμικού ή τις κοινότυπες εφαρμογές χρηστών, η συνολογική ροή ελέγχου του προγράμματος δεν ελέγχεται από τον caller, αλλά από το framework λογισμικού.
- Τυπική συμπεριφορά: Ένα framework λογισμικού έχει μία τυπική συμπεριφορά. Αυτή η τυπική συμπεριφορά πρέπει να είναι χρήσιμη συμπεριφορά και όχι ένα σύνολο από NOPs. Τα NOPs (No operation) είναι

ένα σύνολο εντολών είτε σε επίπεδο assembly ή σε προγραμματιστικό κώδικα, που δεν προσφέρουν καμμία απολύτως λειτουργικότητα.

- **Επεκτασιμότητα:** Ένα framework λογισμικού μπορεί να επεκταθεί από τον χρήστη, συνήθως γράφοντας κώδικα “πάνω” από ήδη υπάρχουσα λειτουργικότητα / κώδικα, έτσι ώστε να εξειδικευτεί η λειτουργία με τον κώδικα που έγραψε ο προγραμματιστής, και να παρέχεται έτσι συγκεκριμένη λειτουργικότητα.
- **Μη τροποποιήσιμο κώδικα του framework:** Σε γενικές γραμμές, ο κώδικας που παρέχεται στο framework λογισμικού δεν πρέπει να τροποποιείται, ενώ όμως παράλληλα δέχεται αλλαγές μέσω επεκτάσεων που κατασκευάζουν οι προγραμματιστές. Με λίγα λόγια, οι χρήστες / προγραμματιστές μπορούν να επεκτείνουν το framework λογισμικού, αλλά όχι να τροποποιήσουν τον κώδικά του.

### **2.10.2 Προσδιορίζοντας ένα CMS**

Τα Συστήματα Διαχείρισης Περιεχομένου (ΣΔΠ, Content Management Systems, CMS) είναι διαδικτυακές εφαρμογές που επιτρέπουν την online τροποποίηση του περιεχομένου ενός δικτυακού τόπου.

Οι διαχειριστές μέσω του διαδικτύου ενημερώνουν το περιεχόμενο στο ΣΔΠ, το οποίο είναι εγκατεστημένο σ' ένα διακομιστή. Οι αλλαγές αυτές γίνονται αυτόματα διαθέσιμες πάλι μέσω του διαδικτύου, σε όλους τους επισκέπτες και χρήστες του δικτυακού τόπου.

#### **Επιχειρησιακά συστήματα διαχείρισης περιεχομένου (ECMS - Enterprise Content Management System)**

Ένα επιχειρησιακό σύστημα διαχείρισης περιεχομένου (ECMS - Enterprise Content Management System), ασχολείται με το περιεχόμενο, τα έγγραφα, τα στοιχεία και τα παραστατικά που έχουν να κάνουν με τις οργανωτικές διαδικασίες μιας επιχείρησης. Η όλη λογική και ο σκοπός του ECMS είναι να διαχειρίζεται το περιεχόμενο και τις αδόμητες πληροφορίες του οργανισμού, με όλες του τις ποικιλίες μορφής και τοποθεσίας. Ουσιαστικά αποτελεί εξειδικευμένη μορφή ενός Content Management System.

#### **Συστήματα διαχείρισης περιεχομένου ιστού (WCMS – Web Content Management System)**

Ένα σύστημα διαχείρισης περιεχομένου (WCMS) είναι ένα σύστημα CMS σχεδιασμένο να απλοποιήσει τη δημοσίευση του δικτυακού περιεχομένου των web sites και των κινητών συσκευών, μεταξύ άλλων, επιτρέπει στους δημιουργούς περιεχομένου να υποβάλουν περιεχόμενο χωρίς να απαιτεί τεχνική γνώση HTML ή



μεταφόρτωση των αρχείων. Γνωστά παραδείγματα τέτοιων CMS είναι τα Wordpress, Joomla, κτλ. Πολλά web-based συστήματα διαχείρισης περιεχομένου μπορούν να βρεθούν τόσο στην κοινότητα ανοιχτού κώδικα όσο και σε διάφορες εμπορικές εκδόσεις οι οποίες έχουν κατοχυρωμένα πνευματικά δικαιώματα. Ωστόσο, αυτός είναι ένας τομέας στον οποίο το OSS (Open Source Software) έχει αποκτήσει δεσπόμενη θέση έναντι των ιδιοκτησιακής μορφής ομολόγων του.

### **Σύστημα διαχείρισης περιεχομένου εξαρτημάτων λογισμικού (Component content management system)**

Χρησιμοποιώντας ένα στοιχείο του συστήματος διαχείρισης περιεχομένου, το περιεχόμενο αποθηκεύεται και αποτελεί αντικείμενο διαχείρισης σε επίπεδο Υποεγγράφου (ή εξαρτήματος), για μεγαλύτερη επαναχρησιμοποίηση του περιεχομένου.

### **2.11 Επίλογος**

Σε αυτό το κεφάλαιο έγινε μία σύντομη εισαγωγή σε τεχνικά χαρακτηριστικά που απαρτίζουν έναν ιστότοπο, όπως για παράδειγμα το όνομα χώρου που διαθέτει, καθώς και τις τεχνικές λεπτομέρειες του χώρου στον οποίο βρίσκεται, από το υλικό των servers μέχρι και το λογισμικό που απαιτείται για την υλοποίησή τους. Περιγράφηκε επίσης το σύστημα διαχείρισης βάσεων δεδομένων, το πολύ σημαντικό μοντέλο REST, το MVC καθώς και τα διασημότερα πακέτα λογισμικού του Διαδικτύου, τα CMS. Το επόμενο κεφάλαιο συζητάει θέματα που έχουν να κάνουν με την PHP και την MySQL.

### 3.1 Server-side γλώσσες προγραμματισμού: PHP

Η php είναι μια server-side γλώσσα προγραμματισμού που έχει δημιουργηθεί με γνώμονα το Διαδίκτυο και την δημιουργία δυναμικών σελίδων. Ο όρος "δυναμική σελίδα" προσδιορίζει τον τρόπο με τον οποίο ο εξυπηρετητής επιστρέφει τα δεδομένα. Όταν αναφερόμαστε σε στατικό περιεχόμενο, εννοούμε περιεχόμενο που διαμένει στον server και δεν αλλάζει.

Αντίθετα, το δυναμικό περιεχόμενο τροποποιείται με βάση διάφορες παραμέτρους που μπορεί να παρέχει είτε ο χρήστης είτε ένα πρόγραμμα. Για παράδειγμα, μία ιστοσελίδα η οποία επιστρέφει εγγραφές από μία βάση, χαρακτηρίζεται δυναμική, καθώς το περιεχόμενό της εξαρτάται από το περιεχόμενο της βάσης.

Η php παρέχει την δυνατότητα στον εξυπηρετητή να ερμηνεύσει τα δεδομένα και να επιστρέφει αποτελέσματα τα οποία με τη σειρά τους επιστρέφονται στον χρήστη, με τη μορφή html εγγράφου. Ακολουθεί παράδειγμα του κλασσικού προγράμματος "Hello world" σε PHP, καθώς και σύντομη ανάλυσή του.

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

Το παραπάνω αρχείο PHP περιλαμβάνει κώδικα HTML, στον οποίο βρίσκεται ενσωματωμένος κώδικας PHP. Για να αντιλαμβάνεται ο εξυπηρετητής ότι αναφερόμαστε σε PHP κώδικα, χρησιμοποιούνται οι ετικέτες "<?php" για το άνοιγμα και την αρχή του κώδικα PHP και κλείνουμε με τα σύμβολα ">". Η PHP εντολή "echo" δίνει τη δυνατότητα εκτύπωσης HTML κειμένου, και στη συγκεκριμένη περίπτωση μία παράγραφο με περιεχόμενου κειμένου "Hello World".

Εναλλακτικά, μπορεί κάποιος να χρησιμοποιήσει και άλλες scriptογλώσσες αντί της PHP, ορισμένες εκ των οποίων είναι οι Python, Ruby και Perl.

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

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.

### **Απόδοση**

Η PHP είναι πολύ αποτελεσματική. Με ένα φθινό διακομιστή μπορεί κανείς 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.

### **Ολοκλήρωση με Βάσεις δεδομένων**

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, ο προγραμματιστής μπορεί να συνδεθεί απευθείας με τις βάσεις δεδομένων PostgreSQL, mSQL, Oracle, dbm, filePro, Informix, InterBase, Sybase, μεταξύ πολλών άλλων.

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

### **Ενσωματωμένες Βιβλιοθήκες**

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες , που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές

με το Web. Για παράδειγμα, η PHP παρέχει τη δυνατότητα δυναμικής δημιουργίας GIF εικόνων, σύνδεση με άλλες υπηρεσίες δικτύων, αποστολή ηλεκτρονικού ταχυδρομείου, διαχείριση και πρόσβαση στα cookies, καθώς και δημιουργία PDF εγγράφων, και όλα αυτά με λίγες, περιεκτικές και απλές γραμμές κώδικα.

### 3.2 Σύνταξη κώδικα και χαρακτηριστικά της PHP

Ξεκινώντας μπορεί κανείς να δημιουργήσει ένα κομμάτι κώδικα, εισάγοντας αρχικά τα open και close tags.

<? Σύμβολο ανοίγματος

?> Σύμβολο κλεισίματος

Εναλλακτικά, χρησιμοποιείται και το <?php για το άνοιγμα του προγράμματος.

Ότι βρίσκεται ανάμεσα στα tags ανοίγματος και κλεισίματος, ερμηνεύεται ως PHP, ενώ αντίθετα, ότι βρίσκεται εκτός των tags, ερμηνεύεται ως HTML. Είναι δυνατή η ύπαρξη πολλών σημείων στα οποία να ανοίγει και να κλείνει ο κώδικας PHP.

```
<?  
//php code  
?>
```

Η χρήση σχολίων είναι πολύ σημαντική ώστε να μπορεί ο προγραμματιστής να θυμάται τι εργασίες έχει να κάνει, αλλά και για να παρέχει την κατάλληλη τεκμηρίωση στον κώδικά του. Όπως και σε άλλες γλώσσες προγραμματισμού, τα σχόλια μπορεί να είναι μίας γραμμής, ή και περισσότερων:

```
<?  
// όπως σε πολλές γλώσσες προγραμματισμού, έτσι θέτουμε σχόλια μίας γραμμής  
?>
```

Η δεύτερη επιλογή είναι το σχόλιο πολλαπλών γραμμών. Χρησιμοποιεί κανείς τα σύμβολα /\* και \*/ για σχόλια πολλών γραμμών, όπως στο ακόλουθο παράδειγμα:

```
<?  
/* είναι δυνατή η χρήση πολλών  
   γραμμών για την προσθήκη σχολίων */  
?>
```

## Μεταβλητές στην PHP

Μια μεταβλητή είναι ένα μέγεθος ή μια ποσότητα την οποία, μέσω της χαρακτηριστικής ονομασίας της, μπορεί να διαχειριστή ο προγραμματιστής ώστε να εκτελεί ή να εκτελείται σε εντολές.

Υπάρχουν δύο μεγάλες κατηγορίες μεταβλητών, οι προκαθορισμένες (predefined) και οι οριζόμενες από τον χρήστη (user defined).

Τα ονόματα των μεταβλητών αρχίζουν πάντα με το σύμβολο \$. Τα ονόματα αποτελούνται από το αρχικό \$ και μερικά αλφαριθμητικά (a-zA-Z0-9) και είναι δυνατό επίσης να περιέχουν την κάτω παύλα \_. Ο πρώτος χαρακτήρας μετά το \$ πρέπει να είναι το σύμβολο \_ ή ένα γράμμα (a-zA-Z).

Κάθε εντολή της PHP εκτός από τα σχόλια τελειώνει με το ερωτηματικό ;. Για να αναθέσει ο προγραμματιστής τιμή σε μια μεταβλητή, γράφει όνομα ίσον τιμή και το ερωτηματικό. Αν η τιμή πρόκειται για συμβολοσειρά (string), την περικλείει σε εισαγωγικά. Ακολουθεί ένα παράδειγμα:

```
<?
$amount = "100";
?>
```

Ένα χαρακτηριστικό της PHP το οποίο βρίσκεται και στην Javascript, είναι ότι δεν χρειάζεται να οριστεί η μεταβλητή πριν από την πρώτη της χρήση.

Με την PHP ένας προγραμματιστής μπορεί να διεκπεραιώσει και να υλοποιήσει ένα τεράστιο σύνολο λειτουργικότητας. Μερικά παραδείγματα είναι:

### Αποστολή Email μέσω κάποιας HTML φόρμας

Ο προγραμματιστής μπορεί να χρησιμοποιήσει την PHP για να προγραμματίσει τη δυνατότητα αποστολής e-mail μέσω μίας φόρμας σε κάποια ιστοσελίδα. Δίνοντας στους χρήστες αυτή τη μέθοδο επικοινωνίας, ο προγραμματιστής μπορεί να μειώσει σημαντικά το spam e-mail που θα μπορούσε να λάβει με το να τοποθετεί απευθείας ένα link προς το e-mail στην ιστοσελίδα του. Επίσης δίνει τη δυνατότητα επιλογής των πεδίων που θέλει ο χρήστης να γεμίσει, έτσι ώστε να μπορεί ο τελικός παραλήπτης να λάβει όλες τις πληροφορίες που χρειάζεται.

### Είσοδος χρήστη (Member login)

Η PHP μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία μίας ξεχωριστής περιοχής στην ιστοσελίδα, στην οποία έχουν πρόσβαση αποκλειστικά κάποιοι συγκεκριμένοι χρήστες. Μέσω της PHP δίνεται η δυνατότητα εγγραφής των χρηστών στην ιστοσελίδα, και μετέπειτα να γίνεται η χρήση των πληροφοριών αυτών για την είσοδό τους στην ιστοσελίδα. Όλες οι πληροφορίες του χρήστη μπορούν να αποθηκευτούν σε κάποια βάση δεδομένων, όπως για παράδειγμα μία βάση του συστήματος MySQL, με κρυπτογραφημένα συνθηματικά κτλ.

## **Καταμέτρηση επισκεπτών**

Μέσω της PHP προγραμματίζονται επίσης και μετρητές οι οποίοι μετράνε τις επισκέψεις χρηστών σε μία ιστοσελίδα. Ως επιπρόσθετο, ο προγραμματιστής θα μπορούσε να ελέγχει και τα session μέσω PHP, ώστε να μην καταγράφεται ο ίδιος χρήστης δύο φορές. Η τιμή του μετρητή θα μπορούσε να τοποθετηθεί σε ένα απλό αρχείο και όχι σε μία βάση δεδομένων, κάτι που θα το έκανε περιττά πιο περίπλοκο.

## **Αλλαγή / περιστροφή των διαφημιστικών banner**

Εάν ο κάτοχος της ιστοσελίδας διαθέτει banners, θα μπορούσε να χρησιμοποιήσει PHP ώστε, για παράδειγμα, να αλλάζουν / επιλέγονται τυχαία σε κάθε επίσκεψη του χρήστη. Η ίδια λογική μπορεί να εφαρμοστεί στην κυκλική αλλαγή και άλλων πραγμάτων, όπως για παράδειγμα εικόνες από τις κεφαλίδες της ιστοσελίδας, ή τυχαίες εκφράσεις μεγάλων φιλόσοφων, μέσα σε εισαγωγικά.

## **Δημιουργία Ημερολογίου**

Ακόμη, θα μπορούσε να χρησιμοποιηθεί η PHP για την εύρεση της σημερινής ημερομηνίας και μετέπειτα την κατασκευή ενός ημερολογίου για ολόκληρο το μήνα.

## **Τελευταία επίσκεψη χρήστη**

Ενημέρωση των χρηστών σχετικά με την τελευταία τους επίσκεψη, ή ακόμη και με τη λογισμικό έγινε η τελευταία τους επίσκεψη. Η PHP το καταφέρνει αυτό με τη χρήση των cookies, αποθηκεύοντας για παράδειγμα ένα στον φυλλομετρητή του χρήστη. Όταν ο χρήστης επιστρέφει, εάν δεν έχει φυσικά καθαρίσει τα cookies, μπορεί να ενημερωθεί τότε είχε επισκεπτεί την ιστοσελίδα τελευταία φορά.

## **Ανακατεύθυνση χρήστη**

Μπορεί να γραφτεί κώδικας στην PHP που να επιτρέπει την ανακατεύθυνση του χρήστη από την ιστοσελίδα που μόλις επισκέφθηκε, σε κάποια άλλη. Αυτό θα μπορούσε να γίνει είτε επειδή η συγκεκριμένη ιστοσελίδα δεν υπάρχει πια, ή επειδή ο ιδιοκτήτης μπορεί να επιθυμεί την χρήση κάποιου πιο μικρού και σύντομου URL, το οποίο να είναι εύκολο στο να το θυμάται ο χρήστης. Όλες οι πληροφορίες ανακατεύθυνσεις γίνονται μέσω του server, οπότε είναι πιο “ελαφρύ” για τον χρήστη στην περίπτωση που θα γινόταν μέσω HTML.

## **Δημιουργία forum**

Η PHP είναι μία από τις πιο διάσημες γλώσσες για τη δημιουργία διαδικτυακών forum. Μπορεί κάποιος να χρησιμοποιήσει ένα ήδη υπάρχων forum και να το τροποποιήσει σύμφωνα με τις δικές του ανάγκες, ή να γράψει ένα καινούργιο από την αρχή. Η αποθήκευση των δεδομένων που λαμβάνει και επεξεργάζεται η PHP, μπορεί να γίνει με κάποια βάση δεδομένων, όπως για παράδειγμα μία MySQL

βάση. Αυτό επιτρέπει την αποθήκευση τεράστιων ποσοτήτων δεδομένων, καθώς και της γρήγορης πρόσβασης σε αυτά.

### **Δημιουργία δημοσκοπήσεων**

Τέλος, ένα τελευταίο παράδειγμα αποτελεί η δημιουργία δημοσκοπήσεων. Μία δημοσκόπηση μπορεί να διαθέτει μία ερώτηση καθώς και ένα σύνολο επιλογών, όπου κάθε επιλογή μαζεύει έναν αριθμό από ψήφους.

### **3.3 Δυνατότητες της MySQL**

Η MySQL είναι ένα πολύ γρήγορο και δυνατό, σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων δίνει τη δυνατότητα αποθήκευσης, αναζήτησης και ταξινόμησης, καθώς και ανακάλεσης δεδομένων, αποτελεσματικά και γρήγορα. Ο MySQL server ελέγχει την πρόσβαση στα δεδομένα της βάσης δεδομένων, με τέτοιο τρόπο ώστε να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα σε αυτή. Με λίγα λόγια, υποστηρίζεται πολυχρηστικότητα. Η πολυχρηστικότητα επεκτείνεται και στο γεγονός ότι υποστηρίζονται μεγάλες ταχύτητες και διασφάλιση του γεγονότος 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.

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 ανοικτός και διαθέσιμος.

#### **Απόδοση**

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Διατηρείται ένα benchmark από την εταιρία και το οποίο βρίσκεται στην ιστοσελίδα <http://web.mysql.com/benchmark.html>, όπου βρίσκονται όλες οι δοκιμές απόδοσης. Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από αντίστοιχα συστήματα διαχείρισης βάσεων δεδομένων.

#### **Χαμηλό κόστος**

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

#### **Ευκολία Χρήσης**

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν ο χρήστης έχει ήδη χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων, τότε δεν θα έχει πρόβλημα να προσαρμοστεί στη MySQL, λόγω των αρκετών κοινών γνωρισμάτων ανάμεσα στα διάφορα συστήματα.

### **Μεταφερσιμότητα**

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

### **Κώδικας Προέλευσης**

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

## **3.4 Επίλογος**

Σε αυτό το κεφάλαιο δώθηκαν περιγραφές και λεπτομέρειες για δύο σημαντικά τεχνολογικά πακέτα στην τεχνολογία λογισμικού. Αρχικά, περιγράφηκε η γλώσσα προγραμματισμού PHP, δίνοντας έμφαση στα χαρακτηριστικά και τις δυνατότητές της, ενώ στη συνέχεια έγινε μία σύντομη εισαγωγή στην MySQL και στα χαρακτηριστικά της. Το επόμενο κεφάλαιο επιχειρεί να προσεγγίσει ορισμένες βασικές έννοιες του Διαδικτύου σε σχέση με την κοινωνία και το εμπόριο. Γίνεται αναφορά στο e-government, καθώς και στην έννοια του ηλεκτρονικού επιχειρείν.



## 4 Το Διαδίκτυο στην κοινωνία και το εμπόριο

### 4.1 E-commerce

Το E-commerce αποτελεί την ηλεκτρονική έκδοση του εμπορίου και στηρίζεται σε τεχνολογίες όπως είναι το mobile commerce, η μεταφορές ηλεκτρονικών χρημάτων, διαδικτυακό marketing, κτλ. Το πιο διάσημο και τυπικό παράδειγμα υλοποίησης τεχνολογιών e-commerce είναι το ηλεκτρονικό κατάστημα.

Είναι σημαντικό να δούμε μέσω ομοιοτήτων και διαφορών, πως ορίζεται ένα ηλεκτρονικό κατάστημα και πως προσδιορίζεται στον χώρο του Διαδικτύου.

#### Τοποθεσία

Το πιο εμφανές χαρακτηριστικό ενός ηλεκτρονικού καταστήματος είναι το γεγονός ότι δεν έχει φυσική παρουσία, αλλά ηλεκτρονική, σε αντίθεση με ένα τυπικό κατάστημα το οποίο χωροταξικά καταλαμβάνει φυσικό χώρο.

#### Ρόλοι εργαζομένων

Συνέπεια της προηγούμενης διαφοράς είναι το γεγονός ότι οι εργαζόμενοι σε ένα ηλεκτρονικό κατάστημα έχουν διαφορετικούς ρόλους και οριοθετούνται γύρω από τη διαχείριση του ηλεκτρονικού καταστήματος, περιλαμβάνοντας την εξυπηρέτηση πελατών, τη διαχείριση των προϊόντων κτλ χωρίς φυσική παρουσία.

#### Ωράριο λειτουργίας

Ένα ηλεκτρονικό κατάστημα διέπεται από διαφορετικούς κανονισμούς όσον αφορά τις διαδικασίες διαχείρισης και πώλησης. Για παράδειγμα, ένα ηλεκτρονικό κατάστημα λειτουργεί 24ώρες το 24ωρο.

#### Ταχύτητα εξυπηρέτησης

Η εξυπηρέτηση σε ένα ηλεκτρονικό κατάστημα γίνεται με πολύ μεγαλύτερη ταχύτητα, καθώς εξαλείφεται ο ανθρώπινος παράγοντας στο κομμάτι που περιλαμβάνει αριθμητικούς υπολογισμούς, έκδοση αποδείξεων / τιμολογίων κτλ.

#### Κόστος προϊόντων

Στα περισσότερα ηλεκτρονικά καταστήματα, το κόστος των προϊόντων κυμαίνεται σε χαμηλότερες τιμές σε σχέση με ένα τυπικό κατάστημα.

Ένα ηλεκτρονικό κατάστημα λοιπόν δεν έχει φυσική παρουσία. Εάν λάβουμε υπόψη μας και τη φύση του διαδικτύου, μπορούμε να πούμε με ασφάλεια ότι όλα τα επιμέρους “κομμάτια” ενός ηλεκτρονικού καταστήματος ενδέχεται να είναι καταναμημένα σε πολλά μέρη του πλανήτη. Για παράδειγμα, ένα Ιταλικό ηλεκτρονικό κατάστημα θα μπορούσε να διατηρεί τα αρχεία του σε έναν εξυπηρετητή στην Ιαπωνία, τη βάση δεδομένων του σε 5 διαφορετικά κράτη ως

αντίγραφα ασφαλείας, Ιταλικό όνομα Χώρου και χρήση υπηρεσιών από τρίτους παροχείς, οι οποίοι με τη σειρά τους είναι κατανεμημένοι σε άλλα μέρη του πλανήτη.

Προσθέτοντας και τον ανθρώπινο παράδειγμα, ένα άλλο παράδειγμα θα μπορούσε να είναι ένα ηλεκτρονικό κατάστημα του οποίου ο εξυπηρετητής όλων των αρχείων βρίσκεται στην Γαλλία, αλλά οι διαχειριστές και “υπάλληλοι” του καταστήματος, υπεύθυνοι για τον έλεγχο των παραγγελιών ή την προσθήκη των προϊόντων, βρίσκονται στην Αγγλία.

Από πλευράς διαχείρισης, ένα ηλεκτρονικό κατάστημα δίνει τη δυνατότητα στους διαχειριστές του να επεξεργάζονται μια πληθώρα δεδομένων της βάσης, όπως για παράδειγμα τα προϊόντα και τους χρήστες της.

Ο διαχειριστής μπορεί να τοποθετεί, αφαιρεί και να επεξεργάζεται προϊόντα, να διαχειρίζεται τα προσωπικά στοιχεία των πελατών, να εκτυπώνει τιμολόγια και γενικά να διαθέτει όλη τη λειτουργικότητα που θα διέθεται με ένα τυπικό κατάστημα.

Ένα ηλεκτρονικό κατάστημα λοιπόν δεν έχει φυσική παρουσία. Εάν λάβουμε υπόψη μας και τη φύση του διαδικτύου, μπορούμε να πούμε με ασφάλεια ότι όλα τα επιμέρους “κομμάτια” ενός ηλεκτρονικού καταστήματος ενδέχεται να είναι κατανεμημένα σε πολλά μέρη του πλανήτη. Για παράδειγμα, ένα Ιταλικό ηλεκτρονικό κατάστημα θα μπορούσε να διατηρεί τα αρχεία του σε έναν εξυπηρετητή στην Ιαπωνία, τη βάση δεδομένων του σε 5 διαφορετικά κράτη ως αντίγραφα ασφαλείας, Ιταλικό όνομα Χώρου και χρήση υπηρεσιών από τρίτους παροχείς, οι οποίοι με τη σειρά τους είναι κατανεμημένοι σε άλλα μέρη του πλανήτη.

Προσθέτοντας και τον ανθρώπινο παράδειγμα, ένα άλλο παράδειγμα θα μπορούσε να είναι ένα ηλεκτρονικό κατάστημα του οποίου ο εξυπηρετητής όλων των αρχείων βρίσκεται στην Γαλλία, αλλά οι διαχειριστές και “υπάλληλοι” του καταστήματος, υπεύθυνοι για τον έλεγχο των παραγγελιών ή την προσθήκη των προϊόντων, βρίσκονται στην Αγγλία.

## **4.2 E-government**

Το e-government αποτελεί ένα ηλεκτρονικό κοινωνικό σύστημα το οποίο περιλαμβάνει το κράτος, την ηλεκτρονική διεπαφή, τους πολίτες και τις αλληλεπιδράσεις ανάμεσα στους πολίτες και το κράτος μέσω του ηλεκτρονικού πληροφοριακού συστήματος. Ουσιαστικά αποτελεί ένα θεωρητικό μοντέλο το οποίο περιλαμβάνει την ηλεκτρονική διεπαφή των πολιτών με το κράτος, το οποίο

παρέχει διάφορους μηχανισμούς απλοποίησης γραφειοκρατικών διαδικασιών μέσω ηλεκτρονικού συστήματος, ενημέρωση, ψηφοφορίες κτλ.

Η αξιοποίηση της πληροφορικής και επικοινωνιών είναι βασικός παράγοντας ανάπτυξης και βελτίωσης της ποιότητας της ζωής των πολιτών, κυρίως όσον αφορά στις συναλλαγές τους με το Δημόσιο Τομέα - όπως έχει αναγνωρισθεί τόσο σε Ευρωπαϊκό όσο και σε Εθνικό επίπεδο - με την πρωτοβουλία e Europe και το Επιχειρησιακό Πρόγραμμα για την «Κοινωνία της Πληροφορίας» αντίστοιχα.

Καλύτερες δημόσιες υπηρεσίες προσαρμοσμένες στις ανάγκες του πολίτη και του ιδιωτικού τομέα, όπως οραματίζονται στη Λευκή Βίβλο Εκσυγχρονισμού της Κυβέρνησης, απαιτούν την απρόσκοπτη ροή πληροφοριών σε όλα τα κρατικά όργανα. Στην κατεύθυνση αυτή, το Πλαίσιο Διαλειτουργικότητας θέτει τις τεχνικές πολιτικές και τα πρότυπα του κρατικού συστήματος προκειμένου να επιτευχθεί η διαλειτουργικότητα και η διασύνδεση των συστημάτων πληροφορικής στο δημόσιο τομέα. Το Πλαίσιο Διαλειτουργικότητας προσδιορίζει τα απαραίτητα προαπαιτούμενα για διασυνδεδεμένα και με δυνατότητα WEB διοίκησης και αποτελεί θεμελιώδη πολιτική στη συνολική στρατηγική του e- government.

Το μοντέλο που πανευρωπαϊκά προτείνεται για την εξειδίκευση των υπηρεσιών «Ηλεκτρονικής Διακυβέρνησης» αναπτύσσεται σε τέσσερα επίπεδα, με βάση τους τέσσερις κύριους τύπους υπηρεσιών που καλύπτουν:

### **1ο επίπεδο e-government: Υπηρεσίες Πληροφόρησης**

Υπηρεσίες πληροφόρησης μέσω διαδικτύου για την ανάκληση επιλεγμένων & ταξινομημένων πληροφοριών όταν ζητηθούν από τους πολίτες & επιχειρήσεις.

### **2ο επίπεδο e-government: Υπηρεσίες επικοινωνιών και αλληλεπίδρασης**

Υπηρεσίες επικοινωνιών & αλληλεπίδρασης μεταξύ δημοσίων υπηρεσιών και φυσικών, νομικών ή ομάδας προσώπων. Διάθεση ψηφιοποιημένων δικαιολογητικών (π.χ. εντύπων διοικητικών διαδικασιών) μέσω διαδικτύου.

### **3ο επίπεδο e-government: Υπηρεσίες αμφίδρομης επικοινωνίας**

Υπηρεσίες αμφίδρομης επικοινωνίας για την ηλεκτρονική υποβολή δεδομένων και δικαιολογητικών εκ μέρους των πολιτών και των επιχειρήσεων σε απευθείας ηλεκτρονική σύνδεση (χρήση ηλεκτρονικής υπογραφής)

### **4ο επίπεδο e-government: Υπηρεσίες διοικητικής διεκπεραίωσης και συναλλαγών**

Υπηρεσίες διοικητικής διεκπεραίωσης, αιτημάτων πολιτών, παροχή των συγκεκριμένων υπηρεσιών και ολοκλήρωση των συναλλαγών περιλαμβανομένων και ηλεκτρονικών πληρωμών όπου είναι απαραίτητο.

## **5ο επίπεδο e-government: Προσωποποίηση: Προληπτική, στοχευμένη παροχή υπηρεσιών.**

Προβαίνει προληπτικά σε δράσεις με στόχο να προάγει την ποιότητα παροχής της υπηρεσίας και το βαθμό φιλικότητας της προς το χρήστη, ενώ γίνεται και αυτόματη εκτέλεση ορισμένων υπηρεσιών, απαλλάσσοντας από τις αντίστοιχες ενέργειες τον πολίτη είτε την επιχείρηση.

### **4.3 Ηλεκτρονικό εμπόριο και ασφάλεια ηλεκτρονικών αγορών**

Το ηλεκτρονικό εμπόριο σε γενικές γραμμές έχει να κάνει με όλες τις μορφές των συναλλαγών που έχουν σχέση με εμπορικές δραστηριότητες, εμπλέκουν δύο ή και παραπάνω οργανισμούς ή / και ιδιώτες, που στηρίζονται σε επεξεργασία και μετάδοση δεδομένων ψηφιακής φύσης, όπως για παράδειγμα κειμένου, ήχου και οπτικές εικόνες. Το E-commerce θέτει τα θεμέλια για πολλά άμεσα και πρακτικά πλεονεκτήματα για τις επιχειρήσεις, οι οποίες αντιμετωπίζουν μια όλο και πιο παγκοσμιοποιημένη αγορά. Το ηλεκτρονικό εμπόριο αποτελεί τη διεξαγωγή των επιχειρήσεων χρησιμοποιώντας διάφορα σύγχρονα εργαλεία επικοινωνίας, όπως για παράδειγμα: ηλεκτρονικές πληρωμές, ηλεκτρονικές επενδύσεις, ηλεκτρονικές ανταλλαγές δεδομένων, ηλεκτρονική χρηματοδότηση κλπ. Η πλήρως ορισμένη έννοια του Ηλεκτρονικού Εμπορίου έκανε την εμφάνισή της το 1996. Μέχρι το τέλος του 1980, υπήρχε η εφαρμογή του EDI (Electronic Data Interchange), και η οποία είχε γίνει δημοφιλής στις ανεπτυγμένες χώρες, καθώς προκάλεσε παγκόσμια έκρηξη του "εμπορίου χωρίς χαρτιά". Το 95% από τις 1000 τότε μεγαλύτερες εταιρείες παγκοσμίως είχε αρχίσει να χρησιμοποιεί σε κάποιο μέρος τους το EDI. Σε παράλληλο χρόνο, η χρήση του EDI προωθούσε την εφαρμογή όλων των ειδών της τεχνολογίας των πληροφοριών, η οποία είναι σχετική με τη διαδικασία της επιχείρησης σε πάρα πολλούς τομείς της οικονομίας, όπως: η εμπορία, η κατασκευή, η ανάπτυξη των υποδομών και η παροχή υπηρεσιών. Η ίδια η εφαρμογή EDI αναπτύχθηκε από ένα σύνολο σχετικών τεχνολογιών, ενοποιώντας την όλη διαδικασία της λειτουργίας των επιχειρήσεων, έχοντας ως αποτέλεσμα τη γέννηση του ηλεκτρονικού εμπορίου.

Αν και το ηλεκτρονικό εμπόριο είναι δυνατό να οριστεί μέσα από διάφορες οπτικές γωνίες, η βασική του ιδέα παραμένει η ίδια, η οποία είναι ότι, το ηλεκτρονικό εμπόριο αποτελεί ένα είδος επιχειρηματικής συμπεριφοράς, και η οποία διαθέτει όλους τους απαραίτητους τομείς που χρειάζονται για την εκπλήρωση των αναγκών των επιχειρηματικών δραστηριοτήτων. Η επιχειρηματικότητα, πάντα, θα αποτελεί το κυρίαρχο θέμα του ηλεκτρονικού εμπορίου. Η μόνη αλλαγή στη λογική και στον κύκλο των επιχειρηματικών δραστηριοτήτων είναι στον τρόπο των συναλλαγών, από αυτόν που γινόταν πριν την εμφάνιση του ηλεκτρονικού εμπορίου, τον παραδοσιακό, στον σύγχρονο, όπου γίνεται με τη χρήση του Διαδικτύου.

Πέρα από την ασφάλεια που παρέχουν οι εκάστοτε τρίτες υπηρεσίες που χρησιμοποιούνται κατά τις αγορές, η κρυπτογράφηση αποτελεί πλέον αναπόσπαστο κομμάτι κατά την διεξαγωγή αγορών στο διαδίκτυο. Το πρωτόκολλο SSL υλοποιεί ασφαλής συναλλαγές ανάμεσα στον εμπλεκόμενο χρήστη και το ηλεκτρονικό κατάστημα, κρυπτογραφώντας τα δεδομένα που στέλνει ο χρήστης στο πληροφοριακό σύστημα του καταστήματος και μπορεί να περιλαμβάνουν:

- Ονοματεπώνυμο
- Διεύθυνση κατοικίας
- Password
- Αριθμό πιστωτικής κάρτας

Παρόλα αυτά, το κομμάτι της ασφάλειας δεν υλοποιείται μόνο στην επικοινωνία μεταξύ χρήστη και καταστήματος, αλλά και αυτόνομα στο κατάστημα, καθώς ο κακόβουλος χρήστης μπορεί να κλέψει προσωπικά στοιχεία απευθείας από την βάση δεδομένων του ηλεκτρονικού καταστήματος.

Σε αυτές τις περιπτώσεις, το ηλεκτρονικό κατάστημα φροντίζει να παίρνει διάφορα μέτρα ασφαλείας, όπως για παράδειγμα κρυπτογράφηση και hashing των password των χρηστών, πριν αποθηκευτούν. Έτσι, όσο πιο περίπλοκο password διαθέτει ένας χρήστης, τόσο πιο δύσκολο γίνεται για τον κακόβουλο χρήστη να εντοπίσει το password, ακόμη και αν καταφέρει να μπει στη βάση δεδομένων.

Τα προβλήματα που αρχικά ανέκυψαν από την παρεμβολή κακόβουλων χρηστών και τις υποκλοπές στοιχείων πιστωτικών καρτών έχουν, στο μεγαλύτερο βαθμό, επιλυθεί και έτσι η συναλλαγή μέσω Διαδικτύου έχει γίνει κατά πολύ ασφαλέστερη σε σχέση με το παρελθόν. Σήμερα οι δύο τύποι αξιόπιστων τεχνολογιών ασφαλείας που είναι διαθέσιμα για ηλεκτρονικές αγορές είναι το SSL (Secure Socket Layer) και το SET (Secure Electronic Transaction).

Όταν πραγματοποιούνται συναλλαγές με ηλεκτρονικά καταστήματα μπορεί να χρησιμοποιηθεί το SSL για να προστατευτεί η μυστικότητα των συναλλαγών. Παρέχεται μυστικότητα κρυπτογράφησης του «καναλιού» της συναλλαγής μεταξύ πελάτη και εμπόρου. Όταν η συναλλαγή είναι πράγματι εμπιστευτική θα πρέπει να εμφανιστεί το ανάλογο λογότυπο προστασίας πχ κλειδαριά στο πλαίσιο του φυλλομετρητή και η διεύθυνση (URL) να έχει αλλάξει από <http://διαδρομή...> σε <https://διαδρομή...> Το HTTPS είναι μια ασφαλής μορφή του γνωστού πρωτοκόλλου μεταφοράς υπερκειμένου HTTP (Hyper Text Transfer Protocol), ώστε να είναι εξασφαλισμένη η ανταλλαγή πληροφοριών ανάμεσα στον φυλλομετρητή και τον Web server.

Η τεχνολογία SET αναπτύχθηκε για την εξακρίβωση και γνησιότητας ταυτότητας μεταξύ εμπόρων και καταναλωτών πριν από μία ηλεκτρονική συναλλαγή. Συγκεκριμένα παρέχει εμπιστευτικότητα και ακεραιότητα των κρίσιμων μεταδιδόμενων πληροφοριών αλλά και πιστοποίηση ότι ο έμπορος μπορεί να δέχεται συναλλαγές με πιστωτική κάρτα μέσω συνεργασίας από κάποιο οικονομικό οργανισμό αλλά και πιστοποίηση ότι ο κάτοχος της κάρτας είναι πραγματικά ο νόμιμος και γνήσιος χρήστης του λογαριασμού. Το SET δημιουργήθηκε από τη Visa και τη MasterCard.

#### **4.4 Κοινωνική δικτύωση και εφαρμογές**

Ο όρος κοινωνική δικτύωση έχει να κάνει με τη χρήση κοινωνικών δικτύων για την οργάνωση και την παροχή υπηρεσιών στους χρήστες του. Ο άνθρωπος, ως κοινωνικό ον, χρησιμοποιεί το διαδίκτυο ως επέκταση της φυσικής του κοινωνικότητας. Διαμοιράζεται περιεχόμενο, διατηρεί προφίλ, ανταλλάσσει μηνύματα, εκφράζει τις απόψεις του, κτλ. Παρακάτω θα δούμε εν συντομία τα πιο διάσημα κοινωνικά δίκτυα.

##### **4.4.1 Facebook**

Το Facebook ξεκίνησε το Φεβρουάριο του 2004 και μέχρι σήμερα φιλοξενεί σελίδες πάνω από ενός δισεκατομμυρίου χρηστών. Το Facebook παρέχει τη δυνατότητα ανταλλαγής μηνυμάτων, ανάρτηση περιεχομένων στην προσωπική σελίδα του καθενός, δυνατότητα δήλωσης αρέσκειας περιεχομένου μέσω του γνωστού "Like", Βιντεοκλήσεις, κτλ.

Κάθε προφίλ στο Facebook χαρακτηρίζεται από μία πληθώρα ιδιοτήτων. Κάθε χρήστης μπορεί να ανεβάσει φωτογραφίες, βίντεο, να προσθέσει τις αρέσκειές του σε πολυμεσικές εφαρμογές όπως για παράδειγμα μουσική, ταινίες, κτλ.

Ο Μαρκ Ζάκερμπεργκ ίδρυσε το Facebook ως μέλος του Πανεπιστημίου Χάρβαρντ. Αρχικά δικαίωμα συμμετοχής είχαν μόνο οι φοιτητές του Χάρβαρντ ενώ αργότερα επεκτάθηκε για την Ivy League. Το όνομα της ιστοσελίδας προέρχεται από τα έγγραφα παρουσίασης των μελών πανεπιστημιακών κοινοτήτων μερικών Αμερικάνικων κολεγίων και προπαρασκευαστικών σχολείων που χρησιμοποιούσαν οι νεοεισερχόμενοι σπουδαστές για να γνωριστούν μεταξύ τους. Το 2005 το δικαίωμα πρόσβασης επεκτάθηκε σε μαθητές συγκεκριμένων λυκείων και μέλη ορισμένων μαθητικών κοινοτήτων, ενώ το 2006 η υπηρεσία έγινε προσβάσιμη σε κάθε άνθρωπο του πλανήτη που η ηλικία του ξεπερνούσε τα 13 χρόνια.

Το Facebook σήμερα έχει πάνω από 1 δισεκατομμύριο ενεργούς χρήστες, κατατάσσοντάς το έτσι στη λίστα ταξινόμησης του Alexa ως ένα από τα

δημοφιλέστερα web sites του πλανήτη (2ο μετά το Google). Επίσης, το Facebook είναι ένα από τα δημοφιλέστερα sites για ανέβασμα φωτογραφιών με πάνω από 14 εκατομμύρια φωτογραφίες καθημερινά. Με αφορμή τη δημοτικότητά του, το Facebook έχει υποστεί κριτική και κατηγορηθεί σε θέματα που αφορούν τα προσωπικά δεδομένα και τις πολιτικές απόψεις των ιδρυτών του. Ωστόσο η συγκεκριμένη ιστοσελίδα παραμένει η πιο διάσημη κοινωνική περιοχή δικτύωσης σε πολλές αγγλόφωνες χώρες.

Το Facebook είναι ένας καλός τρόπος δικτύωσης με φίλους και γνωστούς. Παρά το ότι ενέχει κινδύνους (κυρίως για παραβίαση προσωπικών δεδομένων), ο προσεκτικός χρήστης δεν έχει πρόβλημα. Το Facebook ακόμα παρέχει παιχνίδια και υπάρχει η δυνατότητα ανεβάσματος φωτογραφιών και βίντεο.

#### **4.4.2 Twitter**

Το twitter κυμαίνεται σε διαφορετική φιλοσοφία από αυτή του Facebook, καθώς στηρίζεται κυρίως στην γρήγορη ενημέρωση, ανάρτηση νέων και δυνατότητα παρακολούθησης των υπολοίπων χρηστών μέσω της λειτουργίας "Follow".

Registered users can read and post tweets, but unregistered users can only read them. Users access Twitter through the website interface, SMS, or mobile device app. Twitter Inc. is based in San Francisco and has offices in New York City, Boston, Austin and Detroit.

Twitter was created in March 2006 by Jack Dorsey, Evan Williams, Biz Stone and Noah Glass and by July 2006 the site was launched. The service rapidly gained worldwide popularity, with 500 million registered users in 2012, who posted 340 million tweets per day. The service also handled 1.6 billion search queries per day. In 2013 Twitter was one of the ten most-visited websites, and has been described as "the SMS of the Internet."

#### **4.4.3 Google Plus**

Το Google plus αποτελεί μία σχετικά νέα προσπάθεια της Google για την εισχώρηση στον κόσμο των κοινωνικών δικτύων. Αν και δεν έχει μεγάλες διαφορές από τα υπόλοιπα κοινωνικά δίκτυα, το Google Plus δίνει τη δυνατότητα σε όσους διαθέτουν ήδη λογαριασμό στην Google, να διαθέτουν και μία προσωπική σελίδα.

### **4.5 Επίλογος**

Το τέταρτο κεφάλαιο εισήγαγε τον αναγνώστη σε θέματα που αφορούν το ηλεκτρονικό εμπόριο και την ασφάλειά του, καθώς και σε κοινωνικοπολιτικά

ζητήματα που μεταμορφώνονται χάρη στην τεχνολογία, όπως για παράδειγμα η χρήση τεχνολογιών δικτύων σε κυβερνητικές διαδικασίες, γνωστό και ως e-government. Η φύση του 5ου κεφαλαίου είναι πιο τεχνική, καθώς ο αναγνώστης εισχωρεί σε θέματα που σχετίζονται με τον προγραμματισμό, και συγκεκριμένα HTML5, CSS3, JQuery και Ajax.



## 5 Προγραμματισμός με HTML5, CSS3, JQuery και Ajax

### 5.1 Παρέχοντας δομή: HTML5

#### 5.1.1 Τι είναι και ποιες οι δυνατότητες της HTML5

Η HTML (HyperText Markup Language) είναι η βασικότερη και πιο χρησιμοποιημένη γλώσσα σήμανσης για την διαμόρφωση της δομής και προβολή ιστοσελίδων στον Παγκόσμιο Ιστό. Το βασικότερο συστατικό στοιχείο της γλώσσας είναι η ετικέτα (tag) εντός της οποίας εμπεριέχεται το κείμενο και διαμορφώνεται ανάλογα με τον τίτλο της ετικέτας. Για παράδειγμα, εάν θέλουμε να δώσουμε έντονη υφή σε μία λέξη, μία από τις ετικέτες που μπορούμε να χρησιμοποιήσουμε είναι η <strong>. Το κλείσιμο των ετικετών, η παραμετροποίηση καθώς και η ορθή σύνταξη τους, εξαρτάται από την εκάστοτε έκδοση της γλώσσας ή την πολιτική συμμόρφωσης που επιλέγεται.

Η τρέχουσα έκδοση της γλώσσας είναι η 5 και διαθέτει τα παρακάτω χαρακτηριστικά και λειτουργίες, μέσω των καινούργιων διεπαφών που υλοποιεί:

- Στοιχείο "canvas", για το σχηματισμό δισδιάστατων γραφικών
- Δυνατότητα αναπαραγωγής πολυμέσων (Βίντεο και ήχου)
- Δυνατότητα ύπαρξης εφαρμογών που λειτουργούν ακόμη και όταν ο χρήστης βρίσκεται εκτός του Διαδικτύου
- Δυνατότητες Drag and drop
- Δυνατότητα υποστήριξης γεωγραφικής τοποθεσίας
- Μικροδεδομένα
- Διαχείριση ιστορικού φυλλομετρητή
- Υποστήριξη MIME τύπων
- Διαχείριση και υποστήριξη αποθηκευτικού χώρου παρόμοια με τα cookies αλλά με υποστήριξη μεγαλύτερου αποθηκευτικού χώρου
- και άλλα

Ακολουθεί ένα πλήρες παράδειγμα εγγράφου το οποίο εκτυπώνει στην οθόνη "Hello World".

```
<html>
<head>
  <title>Tutorial: HelloWorld</title>
</head>
```

```
<body>
HelloWorld Tutorial
</body>
</html>
```

### 5.1.2 Διαφορές της HTML5 με προηγούμενες εκδόσεις

Η HTML5 έχει διάφορους στόχους οι οποίοι διαφέρουν από την HTML4.

Η βασική διαφορά-στόχος έγκειται στην σταθερή και καλώς ορισμένη μεθοδολογία ελέγχου των σφαλμάτων. Όπως είναι γνωστό, η HTML υποστηρίζει 'σούπα των tags', ή πιο ειδικά, την δυνατότητα να γράφει κανείς λανθασμένης μορφής κώδικα ο οποίος να διορθώνεται μετέπειτα σε ένα εγκυρου τύπου έγγραφο-αρχείο. Το πρόβλημα με αυτό είναι ότι οι κανόνες για αυτή τη διεργασία δεν είναι κάπου γραμμένοι. Όταν ένας νέος κατασκευαστής φυλλομετρητών επιθυμεί να μπει στην αγορά των φυλλομετρητών, απλώς ελέγχουν λανθασμένα αρχεία σε διάφορους φυλλομετρητές (ειδικά των Internet Explorer) και εφαρμόζουν τη λογική της αντίστροφης μηχανικής τους μηχανισμούς ελέγχου των σφαλμάτων. Εάν δεν κάνουν κάτι τέτοιο, τότε πολλές σελίδες δεν θα εμφανίζονται σωστά (οι εκτιμήσεις θέλουν το 90% των σελίδων στο Διαδίκτυο να περιέχουν, έστω και κατά ένα μικρό μέρος, λάθη).

Άρα, η HTML5 προσπαθεί να ανακαλύψει και να κωδικοποιήσει τη μεθοδολογία διαχείρισης των σφαλμάτων, έτσι ώστε οι προγραμματιστές των φυλλομετρητών να μπορούν να τυποποιήσουν και να ελατώσουν κατά πολύ τον χρόνο και το κόστος που απαιτούνται για την συνεπή προβολή των στοιχείων. Ακόμη και για ιστορικούς λόγους, όταν μετά από πολλά χρόνια στο μέλλον, όπου θα έχει πεθάνει η HTML σαν ένα format για τα αρχεία, οι ιστορικοί ίσως επιθυμούν να διαβάσουν αυτά τα αρχεία και να διαθέτουν έναν απόλυτα καλά ορισμένο αλγόριθμο ώστε να μπορούν να πετύχουν αυτό το σκοπό. Χρήσιμα επιπρόσθετα, όπως για παράδειγμα το Local Storage (μία sql βάση δεδομένων που υλοποιείται τοπικά στους browsers και μπορεί να ανακτηθεί μέσω Javascript, για αποθήκευση δεδομένων πέρα από τη χρήση των cookies), νέες μεθόδους εισόδου όπως είναι η ημερομηνία για την οποία ο φυλλομετρητής μπορεί να προβάλει εύκολη στη χρήση διεπαφή χρήστη (έτσι ώστε να μη χρειάζεται ο χρήστης να χρησιμοποιήσει-προγραμματίσει μόνος του κάποιο επιλογέα ημερομηνίας στη Javascript), καθώς και επικύρωση φορμών που υποστηρίζεται από τους φυλλομετρητές. Αυτά τα εργαλεία επιτρέπουν στους προγραμματιστές ευκολότερη ανάπτυξη διαδικτυακών εφαρμογών, αλλά και η υποστήριξη χρηστών με νέα επιπρόσθετα και αναβαθμίσεις θα είναι πιο εύκολη, καθώς πλέον τα περισσότερα εργαλεία θα υπάρχουν τοπικά στην HTML / φυλλομετρητή, και όχι μέσω Javascript.

Ο δευτερεύων στόχος της HTML5 είναι να μπορεί ο φυλλομετρητής να διαθέτει τη δυνατότητα να είναι πλατφόρμα εφαρμογής, μέσω HTML, CSS και Javascript. Έχουν προστεθεί στη γλώσσα πολλά στοιχεία τα οποία μέχρι αυτή τη στιγμή ήταν κυρίως κομμάτια κώδικα από τρίτους με σκοπό τη διευκόλυνση, και ήταν γραμμένα σε flash ή Javascript, όπως για παράδειγμα τα <canvas>, <video>, και το <audio>.

Υπάρχουν διάφορες ακόμη μικρές προσπάθειες που λαμβάνουν χώρα στην HTML5, όπως για παράδειγμα καλύτερα ορισμένοι σημασιολογικοί κανόνες για ήδη υπάρχοντα στοιχεία. Για παράδειγμα, τα στοιχεία <strong> και <em> πλέον στην HTML έχουν διαφορετική σημασία, ενώ ο ασαφής σημασιολογικός προσδιορισμός των στοιχείων <b> και <i> δεν τα εμποδίζουν από το να διαβαστούν σωστά σε παλαιότερου τύπου έγγραφα. Επίσης, προσθήκη νέων στοιχείων με χρήσιμη σημασιολογία, όπως για παράδειγμα τα στοιχεία <article>, <section>, <header>, <aside> και <nav>, τα οποία θα έπρεπε να αντικαταστήσουν την πλειοψηφία των <div>s που χρησιμοποιούνται σε μία ιστοσελίδα, έτσι ώστε να μετατραπούν σε σελίδες σημασιολογικά καλύτερα ορισμένες, αλλά κυρίως, πιο εύκολες στην ανάγνωση. Αυτό θα επέτρεπε, για παράδειγμα, την αποφυγή της ανάγνωσης ενός ολόκληρου εγγράφου (του κώδικά του) ώστε να βρεθεί το κλείσιμο κάποιου <div> στοιχείου. Αντίθετα, στην HTML5 υπάρχει το <header> και το <article> έτσι ώστε η δομή του εγγράφου να είναι πιο σαφής.

## **5.2 Μορφοποίηση σελίδας: CSS3**

### **5.2.1 Τι είναι η CSS3 και πως λειτουργεί**

Η CSS (Cascading Style Sheets) είναι γλώσσα χρησιμοποιείται για τη διαμόρφωση του περιεχομένου των HTML αρχείων ή αρχεία που έχουν υλοποιηθεί με άλλες γλώσσες σήμανσης και ανήκει στην κατηγορία των γλωσσών φύλλων στυλ.

Ο βασικός λόγος σχεδιασμού και υλοποίησης της CSS είναι για τον διαχωρισμό δύο πολύ σημαντικών ρόλων, της δομής ενός HTML εγγράφου και της διαμόρφωσης του. Παρά το γεγονός ότι η διαμόρφωση του εγγράφου μπορεί να γίνει χρησιμοποιώντας CSS κώδικα μέσα στο HTML έγγραφο, ή χρησιμοποιώντας παραμέτρους των ετικετών της HTML, αποτελεί οργανωτικά, δομικά και προγραμματιστικά πολύ καλύτερη λύση ο διαχωρισμός των ρόλων και η αυτόνομη χρήση της CSS για την διαμόρφωση του HTML εγγράφου.

Ο κώδικας CSS, στην περίπτωση που χρησιμοποιείται αυτόνομα, μπορεί να αποθηκευτεί σε ξεχωριστά αρχεία τα οποία διατηρούν την κατάληξη .css. Ακολουθεί ένα παράδειγμα χρήσης CSS. Έστω για παράδειγμα ότι ο προγραμματιστής θέλει το κείμενο το οποίο εμπεριέχεται στην ετικέτα span της

HTML, να είναι είναι έντονα επισημασμένο. Μπορεί να το πετύχει αυτό με τον παρακάτω κανόνα CSS:

```
span{
  font-weight: bold;
}
```

### 5.2.2 Selectors

Τα CSS selectors επιτρέπουν στον προγραμματιστή να επιλέξει και να χειριστεί τα στοιχεία της HTML. Χρησιμοποιούνται ουσιαστικά για να “βρουν” (ή να επιλέξουν) στοιχεία HTML βασισμένα στο id, κλάσεις, τύπους, ιδιότητες, τιμές ιδιοτήτων τους και πολλά άλλα.

#### Ο selector στοιχείου

Ο selector στοιχείου επιλέγει στοιχεία βασισμένα στο όνομα του στοιχείου. Ο προγραμματιστής μπορεί να επιλέξει, για παράδειγμα, όλα τα <p> στοιχεία μίας σελίδας, με τον εξής τρόπο:

(Όλα τα <p> στοιχεία θα είναι κεντραρισμένα και με χρώμα μπλε)

```
p {
  text-align: center;
  color: red;
}
```

#### Ο id Selector

Ο id selector χρησιμοποιεί την ιδιότητα id μίας HTML ετικέτας για να βρει το συγκεκριμένο στοιχείο. Ένα id πρέπει να είναι μοναδικό μέσα σε μία ιστοσελίδα, άρα ο προγραμματιστής πρέπει να χρησιμοποιεί τον id selector όταν επιθυμεί να βρει ένα μόνο, μοναδικό στοιχείο.

Για να βρεθεί ένα στοιχείο με ένα συγκεκριμένο id, ο προγραμματιστής μπορεί να γράψει τον hash χαρακτήρα, ακολουθούμενος από το id του στοιχείου.

Το παρακάτω παράδειγμα εφαρμόζεται στο HTML στοιχείο το οποίο διαθέτει το id με το όνομα para1 :Αξίζει να σημειωθεί ότι πρέπει να αποφεύγεται η εκκίνηση του ονόματος με αριθμό:

```
#para1 {
  text-align: center;
  color: red;
}
```

#### Ο selector κλάσης

Όπως μπορεί κανείς να καταλάβει από το όνομα, ο selector κλάσης (class selector) μπορεί να εντοπίσει στοιχεία στα οποία έχει αποδοθεί μία συγκεκριμένη κλάση. Ο class selector χρησιμοποιεί την ιδιότητα “class” της HTML. Για να βρει κανείς στοιχεία με μία συγκεκριμένη κλάση, πρέπει να γράψει μία τελεία, η οποία ακολουθείται από το όνομα της τάξης. Στο παράδειγμα που ακολουθεί, όλα τα HTML στοιχεία με όνομα κλάσης “center” (class=“center”) θα είναι κεντραρισμένα.

```
.center {  
  text-align: center;  
  color: red;  
}
```

Γίνεται επίσης να προσδιοριστούν μόνο ορισμένα στοιχεία μίας κλάσης. Στο παράδειγμα που ακολουθεί, όλα τα p στοιχεία με όνομα κλάσης center, κεντραρίζονται.

```
p.center {  
  text-align:center;  
  color:red;  
}
```

### Grouping Selectors

Στα φύλλα στυλ (style sheets) υπάρχουν συχνά στοιχεία με το ίδιο στυλ:

```
h1 {  
  text-align: center;  
  color: red;  
}
```

```
h2 {  
  text-align: center;  
  color: red;  
}
```

```
p {  
  text-align: center;  
  color: red;
```

```
}
```

Για να ελαχιστοποιηθεί ο κώδικας, είναι δυνατή η ομαδοποίηση των group selectors στον κώδικα. Για να γίνει ομαδοποίηση κάνοντας χρήση των group selectors, οι selectors μπαίνουν σε μία σειρά, και χωρίζονται μεταξύ τους με κόμμα. Στο ακόλουθο παράδειγμα υπάρχει ομαδοποίηση των selectors που χρησιμοποιήθηκαν στα τρία παραπάνω κομμάτια κώδικα:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

### 5.2.3 Animations

Ένα animation επιτρέπει σε ένα στοιχείο να αλλάζει σταδιακά από το ένα στυλ στο άλλο.

Ο προγραμματιστής μπορεί να αλλάξει όσες ιδιότητες θέλει, όσες φορές θέλει. Επίσης, μπορεί να καθορίσει πότε η αλλαγή θα συμβεί, ορίζοντας ποσοστό, ή μπορεί να χρησιμοποιήσει τις λέξεις κλειδιά “από” και “έως” (from, to) οι οποίες μπορούν να αναπαρασταθούν με τις τιμές 0-100%.

Το 0%, όπως είναι φυσικό, αντιπροσωπεύει την έναρξη του animation, το 100% είναι όταν η κίνηση είναι πλήρης.

Παράδειγμα χρήσης:

```
/* Chrome, Safari, Opera */  
@-webkit-keyframes myfirst {  
  0% {background: red;}  
  25% {background: yellow;}  
  50% {background: blue;}  
  100% {background: green;}  
}  
  
/* Standard syntax */  
@keyframes myfirst {  
  0% {background: red;}  
  25% {background: yellow;}  
  50% {background: blue;}  
}
```

```
100% {background: green;}  
}
```

Στο προηγούμενο παράδειγμα, το χρώμα αλλάζει σε κίτρινο, από κόκκινο, στο 25%, από κίτρινο σε μπλε στο 50%, και από μπλε σε πράσινο στο 100%.

### 5.3 Προγραμματισμός με jQuery

#### 5.3.1 Τι είναι η jQuery και ποια η σχέση της με την Javascript

Η jQuery είναι μια βιβλιοθήκη Javascript σχεδιασμένη να απλοποιήσει τη υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα.

η jQuery είναι ελεύθερο λογισμικό και παρέχεται στον κόσμο με άδεια MIT.

Ουσιαστικά η jQuery είναι ένα αρχείο Javascript, που περιέχει όλες τις λειτουργίες. Μπορεί να συμπεριληφθεί σε μια ιστοσελίδα παρέχοντας το αρχείο τοπικά. Παράδειγμα χρήσης:

```
<script type="text/Javascript" src="jquery.js"></script>
```

ή έχοντας ένα σύνδεσμο σε ένα από τους πολλούς διακομιστές που τη φιλοξενούν.

```
<script  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

#### 5.3.2 Χαρακτηριστικά και δυνατότητες της jQuery

- DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών Sizzle.
- DOM διάσχιση και τροποποίηση (υποστηρίζοντας CSS 1-3)
- χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
- Events
- Εφέ και κινητά στοιχεία
- AJAX
- Επεκτασιμότητα μέσω plug-ins

- Εργαλεία όπως πληροφορίες user-agent, ανίχνευση χαρακτηριστικών.
- Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προγράμματα περιήγησης.
- Υποστήριξη πολλαπλών φυλλομετρητών

### 5.3.3 Χρήση Ajax μέσω jQuery

Η Ajax τεχνολογία είναι η τέχνη της ανταλλαγής δεδομένων με έναν server και ενημέρωσης / αλλαγής των δεδομένων μίας ιστοσελίδας, χωρίς την επαναφόρτωση ολόκληρης της σελίδας. AJAX = Asynchronous Javascript and XML. Με λίγα λόγια, η AJAX έχει να κάνει με την φόρτωση δεδομένων στο παρασκήνιο και εμφάνισή τους στην ιστοσελίδα, χωρίς να ανανεωθεί ολόκληρη η σελίδα.

Παραδείγματα ορισμένων εφαρμογών που χρησιμοποιούν AJAX: Gmail, Google Maps, Youtube, καθώς και οι καρτέλες Facebook. Η jQuery παρέχει αρκετές μεθόδους για τη λειτουργικότητα της AJAX.

Με τις μεθόδους που παρέχει η jQuery για τη χρήση της AJAX, μπορεί να ζητηθεί κείμενο, HTML, XML ή JSON από έναν απομακρυσμένο διακομιστή χρησιμοποιώντας HTTP GET και HTTP Post, και μπορεί να γίνει φόρτωση των εξωτερικών δεδομένων απευθείας στα επιλεγμένα HTML στοιχεία της ιστοσελίδας.

Το να γράφει κανείς κανονικό κώδικα AJAX μπορεί να αποτελεί περίπλοκο εγχείρημα, γιατί διαφορετικές εφαρμογές περιήγησης έχουν διαφορετική σύνταξη για την εφαρμογή της τεχνολογίας AJAX. Αυτό σημαίνει ότι θα πρέπει να γραφεί επιπλέον κώδικας, έτσι ώστε να μπορεί η εφαρμογή να λειτουργήσει σε όλους τους γνωστούς / κοινούς φυλλομετρητές. Ωστόσο, το API της jQuery απλουστεύει πολύ τη συγκεκριμένη διαδικασία, επιτρέποντας την εφαρμογή της AJAX με μόλις μία γραμμή.

#### Η μέθοδος load() της jQuery

Η μέθοδος load() της jQuery είναι απλή, αλλά αποτελεί δυνατή μέθοδο της τεχνολογίας AJAX. Η μέθοδος load() φορτώνει δεδομένα από ένα server και επιστρέφει τα παραχθέντα δεδομένα σε ένα επιλεγμένο στοιχείο.

#### Σύνταξη:

```
$(selector).load(URL,data,callback);
```

Η παράμετρος URL είναι απαραίτητη-απαιτούμενη, και προσδιορίζει το URL το οποίο επιθυμεί ο χρήστης / προγραμματιστής να φορτώσει.

Η data αποτελεί προαιρετική παράμετρο και προσδιορίζει ένα σει από συνδυασμούς αλφαριθμητικών σε μορφή κλειδιού / τιμής, τα οποία ορίζουν διάφορες ρυθμίσεις για το συγκεκριμένο request.



Τέλος, η callback είναι άλλη μία προαιρετική παράμετρος και αποτελεί το όνομα της συνάρτησης που θα εκτελεσθεί, αφότου ολοκληρωθεί (ολοκληρωθεί, και όχι εκτελεσθεί μόνο) η μέθοδος load().

Έστω για παράδειγμα ότι διαθέτουμε ένα αρχείο με το όνομα demo\_test.txt, το οποίο περιέχει το κείμενο:

```
<h2>hello world, ajax style!</h2>
<p id="p1">Test text for ajax loading purposes. This text resides in
demo_test.txt</p>
```

Το ακόλουθο παράδειγμα φορτώνει το περιεχόμενο του αρχείου demo\_test.txt σε ένα συγκεκριμένο <div> στοιχείο:

```
$("#div1").load("demo_test.txt");
```

Είναι επίσης δυνατό να προστεθεί ένας selector της jQuery στην URL παράμετρο.

Για παράδειγμα, η ακόλουθη εντολή φορτώνει το περιεχόμενο του στοιχείου με το id "p1", το οποίο βρίσκεται μέσα στο αρχείο demo\_test.txt, σε ένα συγκεκριμένο <div> στοιχείο:

```
$("#div1").load("demo_test.txt #p1");
```

Όπως αναφέρθηκε και πιο πάνω, η προαιρετική παράμετρος callback προσδιορίζει την συνάρτηση-επιστροφή η οποία θα εκτελεσθεί όταν τελειώσει η λειτουργία της load μεθόδου. Η callback συνάρτηση μπορεί να έχει διαφορετικές παραμέτρους:

**responseTxt** – Το περιεχόμενό της διαθέτει το αποτέλεσμα που επιστράφηκε, εάν η κλήση στο συγκεκριμένο URL ήταν επιτυχής

**statusTxt** – Περιλαμβάνει την τρέχουσα κατάσταση της κλήσης

**xhr** – Περιλαμβάνει το αντικείμενο XMLHttpRequest

Το ακόλουθο παράδειγμα εμφανίζει ένα alert box αμέσως μόλις ολοκληρωθεί η μέθοδος load(). Εάν η load() μέθοδος έχει επιτυχία, εμφανίζει το μήνυμα "Το εξωτερικό περιεχόμενο φορτώθηκε με επιτυχία!", και εάν δεν πετύχει, επιστρέφει κάποιο αντίστοιχο αρνητικό μήνυμα:

```
$("#button").click(function(){
    $("#div1").load("demo_test.txt",function(responseTxt,statusTxt,xhr){
        if(statusTxt=="success")
            alert("External content loaded successfully!");
        if(statusTxt=="error")
            alert("Error: "+xhr.status+": "+xhr.statusText);
```

```
});  
});
```

#### **5.4 Επίλογος**

το 5ο κεφάλαιο γνώρισε τον αναγνώστη σε έννοιες που αφορούν την εμφάνιση και τον προγραμματισμό διαδικτυακών εφαρμογών. Περιγράφηκε η HTML5, η CSS3, η JQuery καθώς και η χρήση της τεχνολογίας Ajax μέσω της JQuery. Το επόμενο κεφάλαιο αποτελεί το 6ο κομμάτι του θεωρητικού τμήματος αυτής της πτυχιακής και αναλύει το Laravel framework, το οποίο είναι γραμμένο στη γλώσσα προγραμματισμού PHP.

## 6 Το Laravel framework της PHP

### 6.1 Δυνατότητες του Laravel

Το Laravel αποτελεί ένα framework διαδικτυακών εφαρμογών το οποίο προσπαθεί να διευκολύνει τη διαδικασία ανάπτυξης, απλοποιώντας επαναληπτικές διεργασίες που χρησιμοποιούνται στις περισσότερες διαδικτυακές εφαρμογές, περιλαμβάνοντας, αλλά χωρίς να περιορίζεται, το routing, την αυθεντικοποίηση, το caching και τα sessions. Καθώς καταφέρνει να κάνει όλες τις ουσιαστικές εργασίες, από διαδίκτυωση και διαχείριση βάσεων δεδομένων μέχρι και παραγωγή HTML περιεχομένου, το Laravel ονομάζεται και full-stack framework, δηλαδή, framework που δουλεύει σε όλες τις στρώσεις λειτουργιών.

Η συγκεκριμένη κάθετη υλοποίηση (υλοποίηση στρώσας) του περιβάλλοντος ανάπτυξης διαδικτυακών εφαρμογών έχει σκοπό να προσφέρει ένα βελτιωμένο και εύχρηστο μοτίβο ανάπτυξης για τον προγραμματιστή.

Αντίθετα με άλλα περιβάλλοντα που υλοποιούν τη λογική στρώσας στο ενσωματωμένο περιβάλλον ανάπτυξής τους, το Laravel είναι μοναδικό με τον τρόπο του, καθώς δίνει προτεραιότητα στις προγραμματιστικές συμβάσεις, σε σχέση με τη διαμόρφωση ρυθμίσεων. Στην πραγματικότητα, αντίθετα με άλλα PHP frameworks τα οποία απαιτούν αρκετή διαμόρφωση πριν την εκκίνηση του κανονικού project, το Laravel χρειάζεται μόνο λίγες γραμμές από PHP κώδικα που να χρειάζονται διόρθωση, πριν να είναι έτοιμο προς χρήση.

Αποφεύγοντας εξ' ολοκλήρου, ή χρησιμοποιώντας ένα πολύ μικρό αριθμό από αρχεία ρυθμίσεων, το Laravel επιτυγχάνει να διατηρεί παρόμοια δομή σε κάθε project, πράγμα που τα καθιστά παρόμοια μεταξύ τους, άρα και εύκολα στην αναγνώριση και ανάγνωση. Παρά το γεγονός ότι αυτό μπορεί στην αρχή να φανεί ως περιορισμός στον τρόπο με τον οποίο ο προγραμματιστής θα δομήσει την διαδικτυακή εφαρμογή του, αυτοί οι περιορισμοί στην πραγματικότητα επιτρέπουν πλήρη ελαστικότητα και ευκολία στην κατασκευή εφαρμογών.

Στην ουσία, όλα τα νέα projects στο Laravel είναι εξοπλισμένα με ένα πλήρες δομημένο δίκτυο από φακέλους, καθώς και πολλά "placeholder" αρχεία. Η συγκεκριμένη δομή επιτρέπει την γρήγορη εκκίνηση, ανάπτυξη και προτυποποίηση ιδεών και κατά συνέπεια της πραγματικής προγραμματιστικής διαδικασίας. Παρόλα αυτά, αυτή η δομή είναι απόλυτα παραμετροποιήσιμη.

## **6.2 Model, View, Controller**

Το Laravel αποτελεί ένα πλήρες και ολοκληρωμένο MVC framework. Το MVC ραγδαία μετατράπηκε σε ντεφάκτο βιομηχανικό λογισμικό και χρησιμοποιείται σε υπεράριθμα σύγχρονα περιβάλλοντα ανάπτυξης. Πολλά frameworks, όπως τα Ruby on Rails, ASP.NET, CakePHP και το CodeIgniter, το χρησιμοποιούν για να διαχωρίσουν τη λογική της εφαρμογής από το επίπεδο παρουσίασης.

Ένα αρχιτεκτονικό μοτίβο βασισμένο στο MVC, επιτρέπει στην διαδικτυακή εφαρμογή να έχει πολλά διαφορετικά views για ένα κοινό μοντέλο. Για παράδειγμα, στην κατασκευή ενός website ηλεκτρονικού εμπορίου, μία σελίδα κατηγοριών μπορεί να έχει πολλαπλά views, όπως για παράδειγμα το view μίας λίστας με προϊόντα, ή το view μίας γκαλερί με προϊόντα. Στο αντίστοιχο περιβάλλον ανάπτυξης του MVC, θα δημιουργηθεί ένα model για τον πίνακα των κατηγοριών και πολλαπλά views για αυτό το μοντέλο. Η αρχιτεκτονική MVC επιτρέπει στον developer να γράψει κώδικα ο οποίος μπορεί να χωριστεί με βάση των ακόλουθων τριών πραγμάτων:

### **6.2.1 Model**

Ένα model αποτελεί την μέθοδο μέσω της οποίας ο developer μπορεί να διαχειριστεί τα δεδομένα. Αποτελείται από ένα επίπεδο που βρίσκεται ανάμεσα στα δεδομένα και την εφαρμογή. Τα ίδια τα δεδομένα μπορούν να αποθηκευτούν σε διάφορους τύπους βάσεων δεδομένων, όπως για παράδειγμα MySQL, ή ακόμη και σε απλή XML ή και αρχεία Excel.

### **6.2.2 Views**

Τα views είναι η οπτική αναπαράσταση της διαδικτυακής εφαρμογής (επίπεδο παρουσίασης / presentation layer) και είναι υπεύθυνα για την παρουσίαση των δεδομένων που ο controller έλαβε από το model (business logic). Μπορούν εύκολα να κατασκευαστούν χρησιμοποιώντας το Blade, την γλώσσα κατασκευής των templates που πηγαίνει μαζί με το Laravel, ή με κώδικα php. Το Blade περιλαμβάνει κληρονομικότητα στα templates και τομείς (sections). Όταν το Laravel εμφανίζει αυτά τα Views, πρώτα εξετάζει τον τύπο του αρχείου, και, εξαρτάται από το εάν καταλήγει σε “.blade.php” ή απλά σε “.php”, αποφασίζει εάν θα το αντιμετωπίσει το view ως template του Blade ή όχι.

### 6.2.3 Controller

Η βασική λειτουργικότητα του Controller(ελεγκτή) είναι να διαχειρίζεται τα αιτήματα και να περνάει δεδομένα στο Model και στα Views. Άρα ένας Controller μπορεί να θεωρηθεί ως ο συνδετικός κρίκος ανάμεσα στο Model και τα Views.

Ο προγραμματιστής έχει την επιλογή να γράψει τη δικιά του επιχειρηματική λογική, είτε στους Routers (δρομολογητές) είτε στους Controllers. Οι Routers μπορούν να είναι χρήσιμοι όταν η διαδικτυακή εφαρμογή είναι μικρή, σε γρήγορα αναπτύσιμες και στατικές σελίδες. Άρα το να γράφει κανείς Controllers για κάθε μοναδική σελίδα της διαδικτυακής εφαρμογής, δεν είναι απαραίτητο.

### 6.3 Ανάπτυξη templates με το Bootstrap

Το Bootstrap είναι αναμφισβήτητα το πιο δημοφιλές frontend framework ανάπτυξης ιστοσελίδων της βιομηχανίας. Προσφέρει μια πλήρη σειρά φιλικών προς το χρήστη, cross-platform και δοκιμασμένα κομμάτια κώδικα για συχνώς επαναλαμβανόμενα μοτίβα και συμβάσεις στο σχεδιασμό διεπαφών χρήστη. Το Bootstrap επιταχύνει σημαντικά την διαδικασία της οικοδόμησης μιας front-end διαδικτυακής διεπαφής λόγω των έτοιμων, ελεγμένων διεξοδικά μείγμα των συμάνσεων της HTML, CSS φύλλων, και συμπεριφοράς της Javascript. Με αυτά τα αναγκαία θεμέλια να ρυθμίζονται εύκολα και γρήγορα, η διεπαφή μπορεί εύκολα να δέχεται τροποποίηση.

Υπάρχουν πολλοί τρόποι για να κατέβει το Bootstrap, αλλά δεν είναι όλοι αυτοί οι τρόποι λήψης του Bootstrap ισάξιοι. Παρ'όλα αυτά, μπορεί κανείς να χρησιμοποιήσει το Initializr το οποίο παράγει templates βασισμένα στην HTML5 Boilerplate, επιτρέποντας τον developer να επιλέξει ποια συστατικά αυτός ή αυτή επιθυμεί από αυτό.

Κατεβάζοντας τα συστατικά που επιθυμεί, μπορεί να πάρει την παρακάτω δομή φακέλων και αρχείων, ομαδοποιώντας λογικά τα κοινά assets και παρέχοντας compiled και ελαχιστοποιημένες εκδόσεις των αρχείων αυτών. Ακολουθεί το παράδειγμα της δομής που αναφέρθηκε πιο πάνω::

```
bootstrap/  
├── css/  
│   ├── bootstrap.css  
│   ├── bootstrap.min.css  
│   ├── bootstrap-theme.css  
│   └── bootstrap-theme.min.css  
└── js/
```

```
| |— bootstrap.js
| |— bootstrap.min.js
|— fonts/
|— glyphsicons-halflings-regular.eot
|— glyphsicons-halflings-regular.svg
|— glyphsicons-halflings-regular.ttf
|— glyphsicons-halflings-regular.woff
```

Η έμφαση της πτυχιακής βρίσκεται στο Laravel, οπότε η εμπάθιση στο Bootstrap framework, ή την HTML και τη CSS, δεν είναι απαραίτητη.

#### **6.4 Composer**

Ένα άλλο χαρακτηριστικό που κάνει το Laravel να ξεχωρίζει από τα άλλα frameworks, είναι ότι είναι προετοιμασμένο για το Composer. Στην πραγματικότητα, το Laravel το ίδιο είναι ένα μίγμα διαφόρων συστατικών του Composer, πράγμα το οποίο βοηθά στην πολύ σημαντική ανάγκη της διαλειτουργικότητας ανάμεσα στο Composer και στο Laravel.

Το Composer είναι ένα εργαλείο διαχείρισης των εξαρτημένων πακέτων για την PHP. Ουσιαστικά, ο κύριος ρόλος του Composer στο Laravel framework είναι ότι διαχειρίζεται την εξάρτηση των εξαρτήσεων των project. Για παράδειγμα, εάν μία βιβλιοθήκη που χρησιμοποιείται σε ένα project εξαρτάται από τρεις άλλες βιβλιοθήκες, και υπάρχει η ανάγκη αναβάθμισεις των τριών αυτών βιβλιοθηκών, τότε δεν απαιτείται η χειρονακτική εύρεση και αναβάθμισή τους. Καθίσταται δυνατό να ανανεωθούν και οι τέσσερεις βιβλιοθήκες μέσω μίας μόνο εντολής, η οποία είναι “\$ composer update”.

Το Composer έχει τη δυνατότητα να διαχειριστεί μία εξάρτηση μέχρι και στο επίπεδο “v”, το οποίο σημαίνει πως όλες οι εξαρτήσεις ενός project μπορούν να διαχειριστούν με ένα μόνο εργαλείο, πράγμα το οποίο είναι πολύ βολικό σαν διαθέσιμη επιλογή όταν ο προγραμματιστής έχει να κάνει με ένα μεγάλο σύνολο βιβλιοθηκών. Άλλο ένα πλεονέκτημα της χρήσης του Composer είναι ότι παράγει και χειρίζετε ένα αυτοδιαχειριζόμενο αρχείο στη ρίζα του φακέλου του project του προγραμματιστή, και το οποίο διαθέτει όλες τις εξαρτήσεις του project οι οποίες φορτώνουν αυτόματα τις κλάσεις, όταν είναι ενσωματωμένες σε ένα PHP script. Αυτό αφαιρεί από τον προγραμματιστή την υποχρέωση του να θυμάται τα μονοπάτια όλων των εξαρτήσεων και να τα περιλαμβάνει σε κάθε αρχείο του project. Το μόνο που χρειάζεται να κάνει είναι να περιλαμβάνει το αρχείο αυτόματης φόρτωσης που παρέχεται από το Composer.

Στην τεχνική του μορφή, το Composer είναι εγκατεστημένο ως PHP εκτελέσιμου αρχείου το οποίο είναι προστεθημένο στη γνωστή μεταβλητή PATH του περιβάλλοντος. Η μεταβλητή PATH είναι ουσιαστικά μία λίστα με τις τοποθεσίες που διασχίζονται, όταν μία εντολή εκτελείται στο τερματικό. Δηλαδή, αντί να τρέχει ο προγραμματιστής την εντολή μέσω του απόλυτου μονοπατιού της, μπορεί να περιλαμβάνει τις απόλυτες διαδρομές των εντολών στη μεταβλητή PATH, και απλώς να εκτελεί μόνο το όνομα της μεταβλητής. Όταν το Composer είναι σωστά εγκατεστημένο, ο προγραμματιστής μπορεί να το εκτελέσει μέσα από τη γραμμή εντολών από οποιοδήποτε μέρος στο σύστημα αρχείων, χρησιμοποιώντας την εντολή “\$composer”. Το project και οι εξαρτήσεις του είναι ορισμένα μέσα σε ένα αρχείο JSON, το οποίο ονομάζεται composer.json.

Ουσιαστικά το Composer αποτελεί εξελεκτικό βήμα για την κοινότητα της PHP, καθώς υπάρχουν χιλιάδες πακέτα λογισμικού επεκτάσης τα οποία είναι ήδη δοκιμασμένα και διαθέσιμα στο αρχείο του πακέτου Composer. Το Laravel framework έχει σχεδιαστεί με τέτοιο τρόπο ώστε να ενσωματώνει εύκολα πακέτα του Composer. Το μόνο που χρειάζεται το Composer είναι να διαβάσει τα περιεχόμενα του composer.json αρχείου και να συνδεθεί στον αντίστοιχο διαχειριστή πακέτων, ο οποίος είναι ένα online αποθετήριο πακέτων, ώστε να διευθετήσει όλες τις απαιτήσεις των πακέτων, αναδρομικά, δηλαδή, να δει τις ανάγκες κάθε πακέτου που ζητήθηκε, καθώς και τις ανάγκες των πακέτων για αυτά τα πακέτα, κοκ. Μετέπειτα αυτά τα πακέτα κατεβαίνουν στον τοπικό φάκελο που ονομάζεται vendor/ και η κατάστασή τους καταγράφεται σε ένα αρχείο που ονομάζεται composer.lock.

Το Laravel framework, συνδυασμένο με τη δύναμη του Composer, δίνει στον προγραμματιστή πολύ περισσότερη ελευθερία στο να επιλέγει τι είδους πακέτα λογισμικού επιθυμεί να χρησιμοποιήσει στην διαδικτυακή του εφαρμογή. Για παράδειγμα, εάν δεν του αρέσει η προεπιλεγμένη εφαρμογή για τη διαχείριση του Mail που είναι εγκατεστημένη στο Laravel, και η οποία λέγεται Swift Mailer, και επιθυμεί να την αντικαταστήσει με ένα διαφορετικό πακέτο λογισμικού, όπως για παράδειγμα το πακέτο PHPMailer το οποίο είναι συμβατό με το Laravel, τότε μπορεί να το κάνει. Άρα, η αλλαγή ανάμεσα σε δύο πακέτα αποτελεί πολύ εύκολο εγχείρημα. Ο προγραμματιστής μπορεί να αντικαταστήσει πακέτα κώδικα όποτε το επιθυμεί και με ευκολία, όταν υπάρχει η ανάγκη, μέσω του Composer και με των κατάλληλων, γρήγορων και απλών ρυθμίσεων του Laravel.

## **6.5 Artisan**

Ένας προγραμματιστής θα πρέπει να αλληλεπιδρά συνήθως με το Laravel framework χρησιμοποιώντας ένα βοηθητικό πρόγραμμα γραμμής εντολών που δημιουργεί και χειρίζεται το περιβάλλον του project του Laravel. Το Laravel έχει ένα

ενσωματωμένο εργαλείο γραμμής εντολών που ονομάζεται Artisan. Αυτό το εργαλείο επιτρέπει στον προγραμματιστή να εκτελεί το μεγαλύτερο μέρος των επαναλαμβανόμενων και κουραστικών εργασιών που αφορούν τον προγραμματισμό, εργασίες που συνήθως οι περισσότεροι προγραμματιστές δυσανασχετούν να κάνουν χειρονακτικά.

Το Artisan μπορεί να χρησιμοποιηθεί ώστε να δημιουργηθεί ένας αρχικός κώδικας-σκελετός, το schema της βάσης δεδομένων καθώς και τις απαραίτητες συνοδευτικές λειτουργίες για επαναφορά και διόρθωσή του συστήματος της βάσης. Είναι επίσης δυνατή η αρχικοποίηση της βάσης μέσω των λεγόμενων σπόρων βάσεων. Κάτι τέτοιο είναι δυνατό ώστε να επιτρέπεται η παραγωγή μερικών βασικών Model, View και Controller αρχείων απευθείας, μέσω της γραμμής εντολών, να γίνει διαχείριση αυτών των στοιχείων καθώς και των αντίστοιχών τους ρυθμίσεων.

Το Artisan επιτρέπει τη δημιουργία προσωπικών εντολών, καθώς και τη χρήση τους σε βολικές εργασίες, όπως, αποστολή isan let us even create our very own commands and do convenient things with them such as sending email που βρίσκονται σε αναμονή (σε παραλήπτες, ή οπουδήποτε είναι απαραίτητο ώστε να εκτελεσθεί σωστά η διαδικτυακή εφαρμογή). Επίσης, καθίσταται δυνατή η εκτέλεση ελέγχων μονάδας (unit tests) για τη διαδικτυακή εφαρμογή, μέσω του Artisan.

## **6.6 Επίλογος**

Στο 6ο κεφάλαιο έγινε πλήρης περιγραφή του Laravel. Προσδιορίστηκε και περιγράφηκε η χρήση του MVC στο Laravel, πως το συγκεκριμένο framework υλοποιεί το Model, το View και τον Controller. Περιγράφηκε το Composer και το Artisan, και δώθηκαν λίγα στοιχεία για την ανάπτυξη templates στο Laravel, κάνοντας χρήση του Bootstrap. Το 7ο κεφάλαιο περιγράφει τη φύση και τον τρόπο απόκτησης ονόματος χώρου και web server, το περιβάλλον στο οποίο στήθηκε το project, και συγκεκριμένα μέσω εικονικής μηχανής και του λειτουργικού συστήματος Debian, όπως επίσης και τον τρόπο εγκατάστασης και λειτουργίας του framework.



## 7 Εξοικείωση, οργάνωση και διαχείριση του web project

### 7.1 Απόκτηση ονόματος χώρου

Η απόκτηση ενός ονόματος χώρου είναι ένα εύκολο εγχείρημα. Στον ελληνικό χώρο το Μητρώο Ονομάτων Internet με κατάληξη .gr (GR-Hostmaster) του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας (ΙΠ-ΙΤΕ) είναι υπεύθυνο για την τεχνική υποστήριξη και λειτουργία του Μητρώου για το Top Level Domain [.GR].

Σύμφωνα με την Απόφαση 693/9 (04/06/2013) της ΕΕΤΤ ο χρήστης που επιθυμεί την εκχώρηση ενός ονόματος πρέπει να καταθέσει σχετική δήλωση σε έναν από τους καταχωρητές του Χώρου Ονομάτων με κατάληξη [.gr] (Ένας από τους πιο γνωστούς καταχωρητές ονομάτων χώρων είναι το paraki.gr.)

Μαζί με την δήλωση καταχώρησης ο χρήστης υποχρεούται στην κατάθεση του σχετικού τέλους εκχώρησης, το οποίο έχει οριστεί από τον καταχωρητή.

Η δήλωση προωθείται από τον καταχωρητή εντός μίας εργάσιμης ημέρας με ηλεκτρονικό τρόπο στο Μητρώο, το οποίο επιστρέφει στον καταχωρητή και στις email διευθύνσεις επικοινωνίας που δηλώνονται σε κάθε δήλωση τον αριθμό πρωτοκόλλου της δήλωσης, όπου αναφέρεται και η ημερομηνία μέχρι λεπτού της ώρας.

Ο καταχωρητής δίδει στον καταχωρούμενο αποδεικτικό κατάθεσης της δήλωσης με τα παραπάνω στοιχεία.

Το Όνομα Χώρου [.gr] ελέγχεται αυτόματα από το Μητρώο. Εφόσον είναι σύμφωνο με όσα ορίζονται στον Κανονισμό Διαχείρισης και Εκχώρησης Ονομάτων Χώρου [.gr] ενεργοποιείται εντός τριών ωρών και εκχωρείται με την παρέλευση πέντε ημερών από την υποβολή της δήλωσης καταχώρησης στο Μητρώο.

Εάν πρόκειται για όνομα με κατάληξη .gov.gr ή όνομα 2ου επιπέδου με κατάληξη .gr που περιλαμβάνεται στη λίστα γεωγραφικών όρων της ΕΕΤΤ, τότε δεν ενεργοποιείται προσωρινά, αλλά η δήλωση εξετάζεται από την ΕΕΤΤ και εγκρίνεται ή απορρίπτεται εντός είκοσι ημερών από την υποβολή της στο Μητρώο.

### 7.2 Απόκτηση εξυπηρετητή διαδικτύου

Με τον όρο Web Server αναφερόμαστε είτε σε έναν ηλεκτρονικό υπολογιστή (hardware), είτε σε ένα λογισμικό (software) που μας επιτρέπει μέσω του Internet και με την βοήθεια του HTTP (Hyper Text Transfer Protocol - Πρωτόκολλο

Μεταφοράς Υπερκειμένου) να αποκτούμε πρόσβαση σε ιστοσελίδες που είναι αποθηκευμένες σε αυτόν.

Κάθε Web Server έχει έναν βασικό διαχειριστή ο οποίος ελέγχει τον διακομιστή. Έτσι όταν ο χρήστης ζητάει άδεια από τον Web Server να "ανοίξει" μια συγκεκριμένη ιστοσελίδα το αίτημά του ελέγχεται πρώτα από τον διαχειριστή της ιστοσελίδας ή του διακομιστή και στην συνέχεια αφού παραχωρηθεί η άδεια ο χρήστης βλέπει την ιστοσελίδα. Η διαδικασία αυτή φαντάζει να είναι πολύπλοκη και χρονοβόρα, όμως, όπως πολύ σωστά αναφέρεται σε σχετικό άρθρο στο [www.topwebhosting.gr](http://www.topwebhosting.gr), "Οι Web Servers είναι ικανοί να χειριστούν εκατοντάδες χιλιάδες αιτήματα το δευτερόλεπτο και μπορούν να ανατρέξουν σε πληροφορίες γρήγορα, παρέχοντας στον επισκέπτη της σελίδας την πληροφορία που αναζητεί σε μηδενικό χρόνο."

Υπάρχουν διάφορα είδη Web Server όπως οι dedicated servers, οι load balancing servers, οι database servers. Επίσης, καθένας μπορεί να μετατρέψει τον ηλεκτρονικό υπολογιστή του σε έναν Web Server εγκαθιστώντας κατάλληλο λογισμικό που να συνδέει τον υπολογιστή του με το Internet.

### **7.3 Δουλεύοντας σε Virtual Machine**

Μία εικονική μηχανή είναι μία τελείως διαφορετική εγκατεστημένη έκδοση λειτουργικού συστήματος πάνω από το τυπικό εγκατεστημένο λειτουργικό σύστημα. Η λογική που χρησιμοποιεί είναι η εξομοίωση μέσω κατάλληλου λογισμικού και εικονικοποίηση μέσω του υλικού, δηλαδή κοινή χρήση του υλικού και από το υπάρχων λειτουργικό σύστημα αλλά και από το εικονικό.

Ουσιαστικά, μία εικονική μηχανή αποτελεί μία υλοποίηση λογισμικού μίας φυσικής μηχανής-υπολογιστή, που λειτουργεί και εκτελεί εντολές αναλογικά με αυτή. Οι εικονικές μηχανές μπορούν να χωριστούν σε δύο κατηγορίες, σύμφωνα με τη χρήση και την ομοιότητά τους με πραγματικές μηχανές.

- Εικονικές μηχανές συστημάτων
- Εικονικές μηχανές διεργασιών

Η πρώτη κατηγορία παρέχει ένα ολοκληρωμένο σύστημα πλατφόρμας το οποίο εξομοιώνει ένα ολοκληρωμένο λειτουργικό σύστημα, ενώ μία εικονική μηχανή διεργασιών εξομοιώνει-τρέχει ένα μοναδικό πρόγραμμα.

Είναι συχνό φαινόμενο το να γίνεται ενοποίηση πολλών server μέσω πολλαπλών εικονικών μηχανών, οι οποίες διαθέτουν τα δικά τους λειτουργικά συστήματα, και όπου διαφορετικές υπηρεσίες τρέχουν σε διαφορετικά εικονικά περιβάλλοντα, παρόλα αυτά, στην ίδια φυσική μηχανή.

Τα βασικότερα πλεονεκτήματα των εικονικών μηχανών, είναι τα ακόλουθα:

- Μπορούν να υπάρχουν πολλαπλά περιβάλλοντα λειτουργικών συστημάτων στο ίδιο μηχάνημα, απομονωμένα το ένα από το άλλο.
- Μία εικονική μηχανή μπορεί να παρέχει αρχιτεκτονική που διαθέτει ένα ρεπερτόριο εντολών το οποίο με τη σειρά του διαφέρει από έναν πραγματικό υπολογιστή.
- Εύκολη συντήρηση, provisioning εφαρμογών, διαθεσιμότητα, καθώς και βολική επαναφορά συστήματος / αντιγράφων ασφαλείας.

Τα βασικότερα μειονεκτήματα είναι:

- Μία εικονική μηχανή δε μπορεί να είναι το ίδιο αποτελεσματική όπως μία αληθινή μηχανή, όταν υπάρχει πρόσβαση στο υλικό.
- Όταν τρέχουν πολλές εικονικές μηχανές, ταυτοχρόνα, σε έναν μόνο υπολογιστή, κάθε μία εικονική μηχανή ξεχωριστά μπορεί να παρουσιάσει μη σταθερή απόδοση, το οποίο εξαρτάται από τον φόρτο εργασίας των υπόλοιπων εικονικών μηχανών που τρέχουν στον συγκεκριμένο υπολογιστή.

#### **7.4 Κάνοντας χρήση του MySQL Workbench**

Το MySQL Workbench είναι ένα ενιαίο οπτικό εργαλείο για τους αρχιτέκτονες της βάσης δεδομένων, τους προγραμματιστές και τους διαχειριστές. Το MySQL Workbench παρέχει μοντελοποίηση των δεδομένων, την ανάπτυξη SQL, και ολοκληρωμένα εργαλεία διαχείρισης για την παραμετροποίηση του διακομιστή, διαχείριση χρηστών, δημιουργία αντιγράφων ασφαλείας, και πολλά άλλα. Το MySQL Workbench είναι διαθέσιμο για Windows, Linux και Mac OS X.

##### **Σχεδιασμός**

Το MySQL Workbench επιτρέπει σε ένα διαχειριστή βάσεων δεδομένων, προγραμματιστή ή αρχιτέκτονα δεδομένων να σχεδιάσει οπτικά, μοντελοποιήσει παράγει και να διαχειριστεί βάσεις δεδομένων. Περιλαμβάνει όσα χρειάζεται ένας σχεδιαστής δεδομένων για να δημιουργήσει περίπλοκα μοντέλα οντοτήτων-συσχετίσεων, μπροστινή και αντίστροφη μηχανική, και παρέχει επίσης βασικά χαρακτηριστικά για την εκτέλεση δύσκολων εργασιών όπως είναι η αλλαγή δικαιωμάτων και τροποποίηση της τεκμηρίωσης, εργασίες που συνήθως χρειάζονται πολύ περισσότερο χρόνο και κόπο.

##### **Ανάπτυξη και προγραμματισμός**

Το MySQL Workbench προσφέρει οπτικά εργαλεία για την δημιουργία, την εκτέλεση και τη βελτιστοποίηση των ερωτημάτων SQL. Η SQL Editor παρέχει τη δυνατότητα επισύμανσης της σύνταξης, auto-complete, επαναχρησιμοποίηση

αποσπασμάτων κώδικα SQL, καθώς και εκτέλεσει εντολών SQL από το ιστορικό. Ο πίνακας συνδέσεων σε βάσεις δεδομένων επιτρέπει στους προγραμματιστές να διαχειρίζονται εύκολα τις συνδέσεις σε αυτές τις βάσεις. Ο Object Browser παρέχει άμεση πρόσβαση σε σχήματα βάσεων δεδομένων καθώς και αντικειμένων.

## **Διαχείριση**

Το MySQL Workbench παρέχει μια οπτική κονσόλα για την ευκολία στη διαχείριση των MySQL περιβάλλοντων και την απόκτηση καλύτερης ορατότητας στις βάσεις δεδομένων. Οι προγραμματιστές και οι διαχειριστές βάσεων δεδομένων μπορούν να χρησιμοποιήσουν τα οπτικά εργαλεία για τη διαμόρφωση των servers, τη διαχείριση των χρηστών, την εκτέλεση δημιουργίας αντιγράφων ασφαλείας και ανάκτησης δεδομένων, τον έλεγχο των δεδομένων ελέγχου, καθώς και την προβολή της υγείας της βάσης δεδομένων.

## **7.5 Εγκατάσταση και προετοιμασία του Laravel**

Το Laravel εγκαταστάθηκε σε εικονική μηχανή, η οποία διαθέτει το λειτουργικό σύστημα Debian 7. Στο λειτουργικό σύστημα προστέθηκαν web server, ssh server, mysql-server, phpmyadmin, καθώς και διάφορα άλλα εργαλεία για μικρές εργασίες που απαιτήθηκαν.

Το phpMyAdmin είναι δωρεάν και ανοικτό λογισμικό το οποίο είναι γραμμένο σε PHP και χρησιμοποιείται για τη διαχείριση της MySQL, με τη χρήση ενός φυλλομετρητή. Αποτελεί μία πλήρη σουίτα διαχείρισης, η οποία μπορεί να διαχειριστεί εργασίες όπως: δημιουργία, αλλαγή ή διαγραφή βάσεων, πινάκων, πεδίων ή γραμμών, να εκτελέσει αιτήματα SQL, να διαχειριστεί χρήστες και δικαιώματα πρόσβασης και διαχείρισης.

Το συνηθισμένο workflow για το ορθό στήσιμο ενός τέτοιου περιβάλλοντος είναι, πρώτα εγκατάσταση της PHP και της MySQL, μετέπειτα του Composer και στη συνέχεια του Laravel.

Η εγκατάσταση του Composer μπορεί να γίνει με την εντολή:

```
curl -sS https://getcomposer.org/installer | php
```

Και μετέπειτα γίνεται η εγκατάσταση του Laravel:

```
composer global require "laravel/installer=~1.1"
```

## **7.6 Επίλογος**

Στο 7ο κεφάλαιο κεφάλαιο περιγράφηκε η φύση και ο τρόπος απόκτησης ενός ονόματος χώρου καθώς και ενός web server, το περιβάλλον στο οποίο στήθηκε το project, και συγκεκριμένα μέσω εικονικής μηχανής και του λειτουργικού συστήματος Debian, όπως επίσης και τον τρόπο εγκατάστασης και λειτουργίας του framework. Το 8ο κεφάλαιο κλείνει περιγράφοντας και μοντελοποιώντας το σύστημα που αναπτύχθηκε, οριοθετώντας τα μοντέλα καθώς και τις μεταξύ τους συσχετίσεις.

## 8 Υλοποίηση εφαρμογής μηχανοργάνωσης δικηγορικού γραφείου

### 8.1 Μηχανοργανώνοντας ένα δικηγορικό γραφείο

Αν και η μηχανοργάνωση με τη μηχανογράφηση φαίνεται να είναι ταυτόσημοι όροι, η μηχανοργάνωση εστιάζει στην παραγωγή έργου με μειωμένο κόστος μέσω της μεταφοράς εργασίας στον υπολογιστή, εργασίας που θα γινόταν “στο χέρι”, ενώ η μηχανογράφηση στη συλλογή και επεξεργασία δεδομένων. Σίγουρα και στις δύο περιπτώσεις, η πληροφορική, έχει σαν αντικείμενο τη δομή και ροή της πληροφορίας, και η μία έννοια εμπεριέχει την άλλη σαν μέρος της, αφού αφ' ενός χρησιμοποιεί υπολογιστές και αφ' ετέρου τα δεδομένα αποτελούν συστατικό της πληροφορίας.

Η γραμματειακή υποστήριξη δικηγορικών υποθέσεων και κατά συνέπεια η μηχανοργάνωσή τους είναι αρκετά πολύπλοκη εργασία. Αυτό οφείλεται κυρίως στον μεγάλο όγκο που χαρακτηρίζει την επικοινωνία μεταξύ γραμματείας/διαχείρισης και πελατών. Σε αυτό το πλαίσιο προτάθηκε ένα web-based σύστημα γραμματειακής υποστήριξης(μηχανοργάνωσης) και διαχείρισης δικηγορικών υποθέσεων.

Ονομάζουμε συσχέτιση τη σχέση πελάτη-υπόθεσης. Μια συσχέτιση περιλαμβάνει κωδικό πελάτη, κωδικό υπόθεσης, τύπο σχέσης, ημερομηνία, ημερομηνία έναρξης κτλ. Ένας πελάτης μπορεί να έχει πολλές συσχετίσεις με μία υπόθεση. Ένας πελάτης μπορεί να έχει πολλές ενεργές συσχετίσεις, μια για κάθε υπόθεση.

### 8.2 Οντότητες και έννοιες που περιλαμβάνει ένα δικηγορικό γραφείο

Οι βασικότερες οντότητες σε ένα δικηγορικό γραφείο είναι οι εξής:

#### 8.2.1 Η Υπόθεση

Κεντρική οντότητα στο σύστημα είναι η υπόθεση. Μία υπόθεση περιλαμβάνει τουλάχιστον τον κωδικό, τίτλο, περιγραφή, κατηγορία, ημερομηνία δημοσίευσης, ημερομηνία λήξης, δικαστήριο, κωδικός δίκης κτλ.

#### 8.2.2 Ο πελάτης

Επίσης, κεντρική οντότητα στο σύστημα είναι ο πελάτης. Κάθε πελάτης έχει κωδικό, ονοματεπώνυμο, επάγγελμα, διεύθυνση, email, τηλ, ΑΦΜ, κα. Κάθε πελάτης έχει μια ή περισσότερες σχέσεις με μία υπόθεση.

#### 8.2.3 Ο δικηγόρος

Ο δικηγόρος είναι η οντότητα στην οποία αποδίδονται υποθέσεις, και τα χαρακτηριστικά της είναι: όνομα, επώνυμο, τηλέφωνο, email, κτλ.

### **8.3 Κάτι για τους ρόλους του δικηγορικού γραφείου και για το μοντέλο MVC**

#### **8.4 Λίγα λόγια για το SEO**

Το SEO (Search Engine Optimization) είναι το σύνολο των μεθοδολογιών που ακολουθούνται για την βελτίωση των αποτελεσμάτων ενός Ιστοτόπου στις μηχανές αναζήτησης. Η βασικότερη μηχανή αναζήτησης σήμερα είναι η Google. Υπάρχουν διάφορες μέθοδοι για την βελτίωση και ρύθμιση ενός Ιστοτόπου με σκοπό την πετυχημένη του παρουσία στις μηχανές αναζήτησης.

Οι βασικές τεχνικές για βελτιστοποίηση στις μηχανές αναζήτησης περιλαμβάνουν τα εξής:

- Δημιουργία ή βελτίωση του περιεχομένου ώστε να είναι φιλικό και ενδιαφέρον στον επισκέπτη αλλά και προσπελάσιμο από τις μηχανές αναζήτησης όπως το Google
- Λέξεις κλειδιά στον τίτλο και στα υπόλοιπα χαρακτηριστικά μιας σελίδας και των στοιχείων της (ALT, rel, title, κλπ)
- Λέξεις κλειδιά σε συνδέσμους από άλλες ιστοσελίδες προς μια συγκεκριμένη ιστοσελίδα
- Λέξεις κλειδιά που εμφανίζονται μέσα στο κείμενο της ιστοσελίδας
- Δημοτικότητα μιας ιστοσελίδας όπως μετράται από τους αλγόριθμους κατάταξης (π.χ. PageRank της Google)

Ένα από τα σημαντικότερα στοιχεία που πρέπει να συμπεριλαμβάνεται στην ιστοσελίδα ώστε να βοηθά στην καλύτερη κατάταξη της στις μηχανές αναζήτησης είναι η συνάφεια (relevancy) μεταξύ του τίτλου, της περιγραφής, των λέξεων κλειδιών και του περιεχομένου της κάθε σελίδας.

Βασικό ρόλο, παίζει ο τίτλος κάθε σελίδας. Θα πρέπει να περιλαμβάνει όσο το δυνατό περισσότερες από τις λέξεις-κλειδιά που έχουν καθοριστεί, αλλά ο τίτλος καλό είναι να παραμένει μικρότερος από δώδεκα λέξεις. Οι μηχανές αναζήτησης δίνουν ιδιαίτερη προσοχή στις ετικέτες h1 και h2.

Η χρήση των λέξεων-κλειδιών είναι ζωτικής σημασίας για την επισήμανση ενός ιστοτόπου από τις μηχανές αναζήτησης. Αν στον ιστοτόπο περιλαμβάνονται περισσότερες από μια σελίδες, οι λέξεις-κλειδιά πρέπει να εμφανίζονται σε όλες. Βασικός παράγοντας, επίσης, είναι οι λέξεις-κλειδιά να συμπεριλαμβάνονται όσο το δυνατό περισσότερες φορές στον κορμό του κυρίως κειμένου της σελίδας (body text). Αυτό συμβαίνει επειδή οι περισσότερες μηχανές αναζήτησης τις "ψάχνουν" και στο κυρίως κείμενο. Αν δε συναντήσουν μια λέξη-κλειδί στο κυρίως κείμενο, απλά την αγνοούν.

Πρέπει να δίνεται όμως ιδιαίτερη προσοχή να μην γίνεται "υπερβολική επανάληψη" (flood) των λέξεων-κλειδιών, καθώς οι μηχανές αναζήτησης μπορεί να τη θεωρήσουν ανεπιθύμητη ("keyword spam"). Μόλις δημιουργήσετε το κείμενο σας, είναι απαραίτητο να δείτε πόσες λέξεις κλειδιά έχετε τοποθετήσει στο άρθρο σας. Το Google αποδέχεται τη συχνότητα των ίδιων λέξεων περίπου στο 7%.

Αμφιλεγόμενο σημείο, ωστόσο, παραμένουν οι μετα-επισημάνσεις (META tags). Δεν έχει διαπιστωθεί επακριβώς πόσο προσμετρώνται από τις μηχανές αναζήτησης, ωστόσο η σύσταση των ειδικών είναι να συμπληρώνονται σωστά τα πεδία μετα-επισημάνσεων για λέξεις-κλειδιά και μετα-περιγραφή (Meta-description).

Για να έχουμε ικανοποιητικά αποτελέσματα χρειάζεται να γίνει διεξοδικός έλεγχος των κύριων ανταγωνιστών μας αλλά και μια έρευνα πάνω στα keywords με εργαλεία όπως αυτά που θα αναφέρω παρακάτω. Όταν δεν γίνει μια σοβαρή δουλειά πάνω στο θέμα αυτό, συχνά το SEO αποτυγχάνει και επίσης καταλήγουμε να βρισκόμαστε συνεχώς σε ένα φαύλο κύκλο στον οποίο «δοκιμάζουμε keywords – δεν βλέπουμε αποτέλεσμα».

Πρέπει να αναγνωρίσουμε πως υπάρχουν πολλοί περισσότεροι παράγοντες που ενδέχεται να συμβάλλουν αρνητικά ή θετικά στη προσπάθειάς μας. Από τους σημαντικότερους είναι η ταχύτητα φόρτωσης της σελίδας. αν μια σελίδα χρειάζεται περισσότερο από 8 δευτερόλεπτα για να φορτώσει (υπό κανονικές συνθήκες πάντα) τότε υπάρχουν πολύ μεγάλες πιθανότητες να δυσαρεστηθεί ο επισκέπτης και να την κλείσει. Φυσικά υπάρχουν εξίσου μεγάλες πιθανότητες αυτή η δυσαρέσκεια να οδηγήσει στον μαυροπινακισμό της σελίδας σας και να μην την ξαναεπισκεφθεί. Οι μηχανές αναζήτησης δεν εκτιμούν καθόλου τις «αργές σελίδες».

#### **8.4.1 Μεθοδολογίες και μηχανισμοί για καλύτερα αποτελέσματα**

Ακολουθεί μία σύντομη λίστα προτάσεων για καλύτερα αποτελέσματα στις μηχανές αναζήτησης:

- Χρησιμοποίηση ευκρινούς ονόματος χώρου άμεσα σχετιζόμενο με το αντικείμενο του Ιστότοπου
- Σωστή χρήση του meta-description και των meta-keywords. Το description και τα keywords αποτελούν μία σύντομη περιγραφή του Ιστότοπου και η οποία εμφανίζεται στα αποτελέσματα των μηχανών αναζήτησης. Στην περίπτωση του Joomla, καλό είναι να φροντίζει ο χρήστης την τοποθέτηση ξεχωριστών meta-keywords για κάθε άρθρο.



- Χρήση φιλικών URL. Καθώς τα μεγαλύτερα CMS είναι βασισμένα στην PHP, τα URL τείνουν να γίνονται μεγάλα και μη κατανοητά, καθώς δεν περιλαμβάνουν λέξεις, αλλά παραμέτρους. Κάνοντας χρήση των φιλικών URL γίνεται αντικατάσταση των παραμετρικών URL με φιλικές προς τον χρήστη διευθύνσεις.
- Μαζί με τον παραπάνω κανόνα, σωστή χρήση λέξεων για τα URLs των άρθρων. Στο Joomla, τα URL ορίζονται από την παράμετρο alias.
- Επικυρωμένο περιεχόμενο HTML. Εργαλεία όπως ο Validator της W3C μπορούν να παρέχουν ένα σύστημα εύρεσης λαθών στον κώδικα HTML. Λανθασμένος κώδικας HTML δεν σημαίνει λάθος εμφάνιση, παρόλα αυτά δυσκολεύει τις αράχνες των μηχανών αναζήτησης να προσπελάσουν σωστά το έγγραφο της Ιστοσελίδας.

## **8.5 Επίλογος**

Όπως είχε αναφερθεί και στην αρχή, αποτελεί δύσκολο εγχείρημα η ολοκλήρωση ενός πλήρους και απόλυτα λειτουργικού συστήματος τέτοιας κλίμακας, και γι'αυτό το λόγο σκοπός αυτής της πτυχιακής ήταν η έμφαση στη λογική και χρήση ενός PHP framework αντί της σκέτης χρήσης μίας γλώσσας, καθώς και η δημιουργία ενός λειτουργικού πρωτοτύπου που να υλοποιεί την εφαρμογή σε ένα βασικό επίπεδο, και κυρίως να δείχνει τις δυνατότητες του Laravel, καθώς εφαρμογές που να υλοποιούν ήδη την ζητηθήσα λειτουργικότητα, υπάρχουν ήδη. Ευελπιστούμε να βοηθήσαμε μελλοντικούς αναγνώστες στο να επιλέξουν ορθότερα το framework της επιλογής τους.

## Αναφορές

<http://el.wikipedia.org/wiki/SQL>  
[http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)  
[http://en.wikipedia.org/wiki/Domain\\_Name\\_System](http://en.wikipedia.org/wiki/Domain_Name_System)  
<http://en.wikipedia.org/wiki/HTML>  
[http://en.wikipedia.org/wiki/Intelligent\\_agent](http://en.wikipedia.org/wiki/Intelligent_agent)  
<http://en.wikipedia.org/wiki/Model-view-controller>  
[http://en.wikipedia.org/wiki/Relational\\_model#SQL\\_and\\_the\\_relational\\_model](http://en.wikipedia.org/wiki/Relational_model#SQL_and_the_relational_model)  
[http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)  
[http://en.wikipedia.org/wiki/Database\\_management\\_system](http://en.wikipedia.org/wiki/Database_management_system)  
[http://en.wikipedia.org/wiki/DNS\\_root](http://en.wikipedia.org/wiki/DNS_root)  
[http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))  
[http://en.wikipedia.org/wiki/Online\\_shopping](http://en.wikipedia.org/wiki/Online_shopping)  
<http://en.wikipedia.org/wiki/Php>  
<http://en.wikipedia.org/wiki/SQL>  
[http://el.wikipedia.org/wiki/Σύστημα\\_Διαχείρισης\\_Περιεχομένου](http://el.wikipedia.org/wiki/Σύστημα_Διαχείρισης_Περιεχομένου)  
[http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)  
<http://en.wikipedia.org/wiki/NOP>  
[http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework)  
[http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)  
[http://en.wikipedia.org/wiki/Reverse\\_proxy](http://en.wikipedia.org/wiki/Reverse_proxy)  
<http://en.wikipedia.org/wiki/Nginx>  
<http://en.wikipedia.org/wiki/PhpMyAdmin>  
<http://el.wikipedia.org/wiki/JQuery>  
[http://el.wikipedia.org/wiki/Ηλεκτρονικά\\_Καταστήματα](http://el.wikipedia.org/wiki/Ηλεκτρονικά_Καταστήματα)  
[http://el.wikipedia.org/wiki/Όνομα\\_τομέα](http://el.wikipedia.org/wiki/Όνομα_τομέα)  
<http://el.wikipedia.org/wiki/Εξυπηρετητής>  
<http://tools.ietf.org/html/rfc1034>  
<http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>  
<http://en.wikipedia.org/wiki/Model-view-controller>  
<http://laravel.com/>

## Βιβλιογραφία

Luke Welling, Laura Thomson (2011), Ανάπτυξη Web εφαρμογών με PHP και MySQL: Εκδόσεις Γκιούρδας

Roy Thomas Fielding (2000), doctoral dissertation, REST

Καράσης Κωνσταντίνος (2014), διπλωματική εργασία, Δημιουργία ηλεκτρονικής επιχείρησης με χρήση τεχνολογιών open-source & cloud

Πομπόρτσης, Ανδρέας Σ. (2002), Εισαγωγή στο Ηλεκτρονικό Εμπόριο: Εκδόσεις Τζιόλα

Search Engine Optimization (SEO) by Google