



ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
Πρόγραμμα Σπουδών
ΕΙΔΙΚΕΥΣΗ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΑΤΡΙΒΗ ΕΠΙΠΕΔΟΥ MASTER

ΜΕΛΕΤΗ ΚΑΙ ΠΙΛΟΤΙΚΗ ΧΡΗΣΗ ΣΥΓΧΡΟΝΩΝ ΤΕΧΝΟΛΟΓΙΩΝ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΔΥΝΑΜΙΚΩΝ ΙΣΤΟΧΩΡΩΝ

ΟΝΟΜΑ ΦΟΙΤΗΤΗ: ΗΛΙΑΣ ΤΕΡΠΕΝΗΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΜΙΧΑΗΛ ΒΑΣΙΛΑΚΟΠΟΥΛΟΣ

ΛΕΥΚΩΣΙΑ 2011



ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
Σχολή Θετικών Επιστημών και Τεχνολογίας
Πρόγραμμα Σπουδών
ΕΙΔΙΚΕΥΣΗ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Διατριβή Επιπέδου Master

Τίτλος Διατριβής

**Μελέτη και Πιλοτική Χρήση Σύγχρονων
Τεχνολογιών για την Ανάπτυξη
Δυναμικών Ιστοχώρων**

Όνομα Φοιτητή: Ηλίας Τερπένης
Επιβλέπων καθηγητής : Μιχαήλ Βασιλακόπουλος

ΛΕΥΚΩΣΙΑ 2011

Περιεχόμενα

- 1 | Κεφάλαιο 1
Με μια πρώτη ματιά
- 2 | Κεφάλαιο 2
Ανάλυση απαιτήσεων και Σχεδίαση
- 3 | Κεφάλαιο 3
jQuery
- 4 | Κεφάλαιο 4
CSS3
- 5 | Κεφάλαιο 5
Ajax
- 6 | Κεφάλαιο 6
Διακομιστής και Βάση Δεδομένων
- 7 | Κεφάλαιο 7
Συμπεράσματα

Περίληψη

Η εργασία που ακολουθεί αποτελεί μια μελέτη νέων τεχνολογιών του διαδικτυακού προγραμματισμού με βάση τις προϋποθέσεις που θέτει το W3C. Μελετήσαμε τεχνικές που συνέβαλαν στην απογείωση αυτού που ονομάζεται WEB2.0, και το οδήγησαν στην εκτεταμένη του μορφή που σήμερα αποκαλούμε WEB3.0. Είναι γνωστό πως οι τεχνολογίες αυτές αποτέλεσαν θέματα πολλών συζητήσεων και πολλών βιβλίων που γράφτηκαν για τον σκοπό αυτό. Στα κεφάλαια που ακολουθούν γίνεται μια προσέγγιση ορισμένων από τις τεχνικές αυτές μέσα από την ανάπτυξη μιας εφαρμογής πραγματικού κόσμου η οποία δημιουργείται αποκλειστικά με τις μεθοδολογίες που θα περιγραφούν.

Τεχνικές όπως το JQuery, που αποτελεί ένα Framework του JavaScript, έδωσαν μια καινούρια αντίληψη του τόσο διαδεδομένου JavaScript και το έφεραν για πρώτη φορά τόσο κοντά στα νερά του Flash, με αποτέλεσμα να δημιουργηθούν εντυπωσιακά και καινοτόμα έργα.

Το CSS3 σε συνδυασμό με την HTML5 ακουμπούν τον χώρο της JavaScript, με αποτέλεσμα την κατασκευή εντυπωσιακών γραφικών τα οποία πριν από μερικά χρόνια θα ήταν αδύνατο να δημιουργηθούν χωρίς την χρήση στατικών γραφικών και της JavaScript.

Περιγράφεται εκτεταμένα και μελετάται ο συνδυασμός τεχνολογιών, γνωστός και ως Ajax, που από πολλούς έχει θεωρηθεί και ως το Άγιο Δισκοπότηρο στην κατασκευή δυναμικών ιστοσελίδων, οι οποίες μιμούνται εφαρμογές πραγματικού χρόνου στο επίπεδο του χρήστη (client side interaction) μέσα από το περιβάλλον που δημιουργεί μια εφαρμογή ιστού.

Για να εξασφαλίσουμε μια πιο άμεση επαφή του αναγνώστη με τα πιο πάνω αντικείμενα, όπως τονίσαμε, κατασκευάσαμε μια δυναμική εφαρμογή που αξιοποιεί τις πιο πάνω τεχνικές, και αποτελεί από παράδειγμα των όσων αναλύουμε. Σκοπός της εφαρμογής είναι να καταλάβει ο αναγνώστης τα πλεονεκτήματα και τα μειονεκτήματα των τεχνολογιών αυτών και να αποκτήσει μια νέα αντίληψη στο θέμα του διαδικτυακού προγραμματισμού, ότι δηλαδή πλέον είναι φυσιολογικό οι ίδιες σελίδες να έχουν διαφορετική μορφή στα διαφορετικά προγράμματα περιήγησης (browsers).

Abstract

The work that follows is a study on new technologies of web programming based on conditions set by the W3C. We have studied techniques that contributed to what is called WEB2.0, and resulted in the extensive form that we now call WEB3.0. It is known that many of these technologies were the subjects of much discussion and many books written for this purpose. In the following chapters we are approaching some of these techniques by developing a real life application that includes methodologies that will be described.

Techniques such as jQuery have made rich client-side interaction and asynchronous/seamless user experiences a reality. This leads to easier deployments of web applications, which in turn, increases competition, which in turn, leads to innovation.

CSS3/HTML5 are opening up various new options for styling content on the web, from multiple backgrounds on page elements, better ability to select and style elements with greater specificity, and colour gradients without reliance on static graphics, to simpler aesthetical improvements, such as support for rounded corners, without the need for complicated sliding doors techniques or JavaScript.

We describe extensively and study the combination of technologies known as Ajax, which is regarded by many as the Holy Grail of the construction of dynamic websites, which mimic real time applications at the client side, through the environment that creates a web application.

To make the contact of the reader with the above items more direct we have developed a dynamic application that utilizes the above techniques, and provides a concrete example of what we analyze. The purpose of this application is the reader to capture the advantages and disadvantages of these technologies, and to gain a new perspective on the issue of online programming that is, to see that there's a good chance that you'll start to see websites that don't look the same in every browser.

Αναλυτικός Πίνακας Περιεχομένων

Περίληψη	7
----------	---

Abstract	9
----------	---

Κεφάλαιο 1 Με μια πρώτη ματιά

Παρουσίαση	16
Τι ονομάζουμε εφαρμογή πελάτη-διακομιστή (client-server);	17
Πώς συμπεριφέρεται ο Client;	18
Ποιος ο ρόλος του Server;	18
Τί ονομάζουμε Web εφαρμογή;	19
Ποια είναι τα πλεονεκτήματα μιας Web εφαρμογής;	19
Ποιο είναι το μέλλον των Web εφαρμογών;	20
Ποιες οι διαφορές μεταξύ των δύο αυτών εφαρμογών;	21
Τι είναι το W3C;	22
Ποιες οι νέες τεχνολογίες κατασκευής διαδικτυακών εφαρμογών;	23
Τι είναι το Web 2.0;	25
Ποιες είναι οι γλώσσες προγραμματισμού ιστοσελίδων;	25
Ποιοι είναι οι web servers;	25
Τι είναι η βάση δεδομένων;	26
Τι είναι οι server side γλώσσες;	26
Τι είναι το IDE;	26
Η εργασία βήμα-βήμα	27

Κεφάλαιο 2 Ανάλυση των απαιτήσεων και σχεδίαση

Παρουσίαση	30
Περιγραφή ICONIX	30
Παρουσίαση της Εφαρμογής	30
Καθορισμός Απαιτήσεων	31
Μοντέλο Πεδίου Προβλήματος	31
Μοντέλο Περιπτώσεων Χρήσης	33
Τεκμηρίωση Περιπτώσεων Χρήσης	35
Περίπτωση Χρήσης: LogIn Procedure	36
Ανάλυση Ευρωστίας	37
Διάγραμμα Ευρωστίας	38
Αναθεώρηση Μοντέλου Πεδίου Προβλήματος	39

Σχεδίαση	40
Διάγραμμα Ακολουθίας	41
Διάγραμμα Κλάσεων	43
Συμπέρασμα	44

Κεφάλαιο 3 JQuery

Παρουσίαση	46
Σε ποιους Απευθύνεται;	46
Τι είναι το JQuery;	47
Γιατί JQuery;	47
Βασικές Αρχές του JQuery	48
To Document Ready	50
Κατασκευή DOM	52
Περιτύλιγμα (Wrapping)	53
Βασικοί CSS επιλογείς	53
Επιλογείς με βάση την Θέση	55
Προσαρμοσμένοι επιλογείς JQuery	56
Δίνοντας ζωή στις Σελίδες μας	57
JQuery vs JavaScript	63
Εργαλεία Ανάπτυξης	66
Συμπέρασμα	68

Κεφάλαιο 4 CSS3

Παρουσίαση	70
CSS3	70
Γιατί να ασχοληθούμε αφού ακόμα δεν υποστηρίζεται πλήρως;	71
Selectors	72
Multi-Column Layout	73
CSS3 HSL&HSLA	74
Border Radius	76
Σκίαση	79
@Font-Face	87
HTML 5	90
Συμπέρασμα	93

Κεφάλαιο 5 Ajax

Παρουσίαση	96
Προετοιμασία	96
Ποιο είναι το απίστευτο;	97

Δεδομένα	98
Επεξεργασία πριν την Προβολή	99
Επιλογή Δεδομένων	106
Ajax στην Πράξη	107
Σελιδοποίηση	115
Διάταξη και Σελιδοποίηση	115
Υλοποίηση Σελιδοποίησης	117
Συμπέρασμα	119
Κεφάλαιο 6 Διακομιστής και Βάση Δεδομένων	
Παρουσίαση	122
PHP	122
Πλεονεκτήματα	123
MySQL	123
Τα πλεονεκτήματα της	124
Apache Server	125
Πώς συνδυάζονται όλα αυτά;	126
Εγκατάσταση	127
Παραμετροποίηση	127
Μεταφερσιμότητα	129
Συμπέρασμα	130
Κεφάλαιο 7 Συμπεράσματα	
Ο τρόπος που αντιλαμβανόμαστε το Web!	132
Χώροι Κοινωνικής Δικτύωσης!	132
Η εφαρμογή μας!	133
Βιβλιογραφία	136
Παράρτημα Α	138
Παράρτημα Β	148



Με μια πρώτη ματιά

Σκοπός

Σε αυτό το Κεφάλαιο θα παρουσιάσουμε:

- βασικές ορολογίες απαραίτητες για να μπορεί ο αναγνώστης να παρακολουθήσει βασικές έννοιες σχετικές με τον διαδικτυακό προγραμματισμό,
- τα περιεχόμενα των κεφαλαίων που ακολουθούν.

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να κατανοήσει έννοιες όπως εφαρμογές client-server και web,
- να γνωρίσει τις διαφορές μεταξύ των τεχνολογιών αυτών,
- να κατανοήσει έννοιες όπως το W3C,
- να γνωρίσει τις νέες τεχνολογίες δημιουργίας δικτυακών εφαρμογών, και τέλος
- να εξοικειωθεί με έννοιες όπως τα web2.0,

1.1 Παρουσίαση

Η διπλωματική αυτή εργασία εκπονήθηκε, στα πλαίσια του μεταπτυχιακού προγράμματος «Πληροφοριακά συστήματα», του Ανοικτού Πανεπιστημίου Κύπρου. Στην εργασία αυτή θα ασχοληθούμε με την μελέτη σύγχρονων τεχνικών διαδικτυακού προγραμματισμού (Web Programming), τις οποίες θα αναλύσουμε και θα περιγράψουμε διεξοδικά. Σκοπός μας βέβαια δεν αποτελεί η δημιουργία ενός εγχειριδίου προγραμματισμού, με βάση της μεθοδολογίες αυτές, αλλά η παρουσίαση των πλεονεκτημάτων που προκύπτουν από την χρήση τους, σε σχέση με τις τεχνικές οι οποίες χρησιμοποιούνται ήδη, και τις οποίες οι νέες θα αντικαταστήσουν.

Για τον λόγο αυτό θα θέλαμε να τονίσουμε ότι το κείμενο αυτό, απευθύνεται σε άτομα τα οποία κατέχουν προχωρημένες γνώσεις Προγραμματισμού στο Διαδίκτυο, τόσο όσον αφορά στην σχεδίαση όσο και στην υλοποίηση σύγχρονων ιστοσελίδων. Για να μπορέσει κάποιος να ακολουθήσει την μελέτη αυτήν και να την κατανοήσει, εάν δεν έχει πρώιμη εμπειρία, συνιστάται η μελέτη εγχειριδίων τα οποία αφορούν στην ανάπτυξη εφαρμογών διαδικτύου από την αρχή.

Όπως τονίστηκε, δεν είναι σκοπός η δημιουργία άλλου ενός εγχειριδίου, που να προστεθεί στην πληθώρα εγχειριδίων που υπάρχουν τόσο στην αγορά όσο και στο διαδίκτυο, αλλά η λεπτομερής παρουσίαση των σύγχρονων τεχνικών, με βάση τα πρότυπα του W3C [21] όπως HTML5, CSS3 [13-14], JQuery [5-7], AJAX [9-11], PHP5 [4].

Πέρα βέβαια από την ανάλυση των παραπάνω τεχνολογιών θα αναπτύξουμε μία εφαρμογή πολύ κοντά σε αυτές που μπορεί κάποιος να συναντήσει κατά την περιήγηση του στο διαδίκτυο. Σκοπό της διπλωματικής εργασίας είναι η μελέτη των τεχνικών που ανήκουν στις κορυφαίες της σημερινής εποχής. Για τον λόγο αυτό δεν θα έχουμε κάποια πλήρη εφαρμογή, αλλά περισσότερο θα πάρουμε κομμάτια τα οποία σχετίζονται με την επικοινωνία της εφαρμογής και του διακομιστή, όπως αναζήτηση και εισαγωγή δεδομένων, τροποποίηση των δεδομένων σε επίπεδο χρήστη. Αυτά βέβαια αποτελούν ορισμένα μόνο παραδείγματα τα οποία σχετίζονται με την χρήση των εργαλείων Ajax και JQuery με την βοήθεια των οποίων δημιουργούμε εφαρμογές ζωντανές οι οποίες το μεγαλύτερο φόρτο εργασίας το μεταφέρουν στο επίπεδο χρήστη. Είναι πολύ σημαντικό το γεγονός αυτό γιατί

με τον τρόπο αυτό περιορίζουμε δραστικά την απασχόληση του διακομιστή ο οποίος φιλοξενεί την εφαρμογή μας.

Μέρος της εφαρμογής αυτής είναι η χρήση δυναμικών γραφικών τα οποία θα αντικαταστήσουν όλα τα στατικά στοιχεία όπως τις εικόνες. Με την χρήση του CSS3 του font face και της HTML5, θα μπορούμε πλέον να δημιουργούμε πραγματικά αριστουργήματα στα οποία θα γίνεται χρήση σκίασης, βάθους εικόνας και στρογγυλοποίησης χωρίς να χρησιμοποιείται κάποιο πρόγραμμα όπως το Photoshop και κυρίως χωρίς τη χρήση στατικών γραφικών που καθυστερούν την φόρτωση στοιχείων της εφαρμογής.

Η καινοτομία μας θα είναι να προσεγγίσουμε και να παρουσιάσουμε τα μυστικά μεγάλων εταιρειών όπως είναι η Google, που αφορούν την σύγχρονη κατασκευή δυναμικών ιστοσελίδων. Λέγοντας δυναμικών ιστοσελίδων εννοούμε ιστοσελίδες που δεν έχουν εικόνες αλλά φαίνεται να χρησιμοποιούν εικόνες και επίσης δεν έχουν τόσο μεγάλο κώδικα που φαίνεται να παρουσιάζουν και δεν είναι client side ενώ φαίνεται να είναι.

Προτού βέβαια προχωρήσουμε και μπούμε στα βαθιά θα προσπαθήσουμε να δώσουμε ορισμούς εννοιών οι οποίες είναι απαραίτητες για την παρακολούθηση του περιεχομένου της εργασίας αυτής.

1.2 Τι ονομάζουμε εφαρμογή πελάτη-διακομιστή (client-server);

Με το όρο πελάτης-διακομιστής (client-server), περιγράφουμε μια αρχιτεκτονική στην οποία ο πελάτης ζητά μια ενέργεια ή μια υπηρεσία από τον πάροχο της υπηρεσίας, τον διακομιστή (server). Ως παράδειγμα θεωρήστε ένα πρόγραμμα περιήγησης Web (browser) και ένα διακομιστή Web (server). Όταν πληκτρολογήσουμε μια διεύθυνση στο παράθυρο του προγράμματος περιήγησης (client), τότε ζητά μια σελίδα από ένα διακομιστή Web. Ο διακομιστής με την σειρά του επιστρέφει μια σελίδα HTML στον πελάτη, στην οποία αναλύει την σελίδα (δεδομένα) και το εμφανίζει στον υπολογιστή σας.

Κατά την ανάπτυξη μιας εφαρμογής πελάτη-διακομιστή (client-server), όπως για παράδειγμα το πρόγραμμα περιήγησης στο Web και τον Web server, θα πρέπει να εξεταστεί

πώς θα χειριστούμε την ανάπτυξη της εφαρμογής μας σε ένα ομαδικό περιβάλλον και πώς πρόκειται να χειριστούμε τη μακροπρόθεσμη συντήρησή της. Κατά την ανάπτυξη μιας τέτοιου είδους εφαρμογής κάνουμε την πρώτη γνωριμία μας με τον προγραμματισμό σε κομμάτια κώδικα (modules). Τι σημαίνει αυτό όμως; Χωρίζουμε το πρόγραμμα σε κομμάτια, γεγονός που διευκολύνει την ανάπτυξη του από ομάδες και παρέχει την καλύτερη δυνατότητα συντήρησής του. Σε μια εφαρμογή πελάτη-διακομιστή (client-server), μια ενότητα (module) δεν πρέπει να είναι μέρος του ίδιου προγράμματος ή ακόμα και να τρέχει στον ίδιο υπολογιστή. Κάθε λειτουργία μπορεί να τρέξει σε μια διαφορετική συσκευή.

1.2.1 Πως συμπεριφέρεται ο Client;

Τα προγράμματα-πελάτες (client) ζητούν υπηρεσίες από έναν διακομιστή με την αποστολή ενός μηνύματος. Αναφερόμενοι στο παράδειγμα του Web, ένας web browser είναι ο πελάτης που χρησιμοποιούμε καθημερινά για να ζητήσουμε συγκεκριμένες ιστοσελίδες. Για παράδειγμα, όταν κάνετε κλικ σε έναν σύνδεσμο για να διαβάσετε ένα άρθρο, ο browser σας στέλνει ένα μήνυμα για παράδειγμα σε ένα διακομιστή Web στο Ωστιν, Τέξας. Ως απάντηση ο browser παίρνει μια σελίδα η οποία περιέχει το άρθρο που θέλετε να διαβάσετε. Το παράδειγμα αυτό αποτελεί την πιο γνωστή εφαρμογή πελάτη-διακομιστή (client-server), την οποία οι περισσότεροι από εμάς χρησιμοποιούμε σε καθημερινή βάση.

1.2.2 Ποιος ο ρόλος του Server;

Ο διακομιστής με την σειρά του εκτελεί όλα τα ερωτήματα του πελάτη. Η επικοινωνία όμως αυτή πολλές φορές δεν είναι και τόσο απλή όσο αυτή που περιγράψαμε προηγουμένως. Σκεφτείτε μια πιο περίπλοκη εφαρμογή στην οποία θέλετε να αγοράσετε ένα προϊόν από μια ιστοσελίδα. Στην περίπτωση αυτή, ο πελάτης ενημερώνει το διακομιστή για το προϊόν που θέλουμε να αγοράσουμε και ο διακομιστής με την σειρά του ενημερώνει μια βάση δεδομένων με το αίτημα αγοράς. Στη συνέχεια, ο διακομιστής ενημερώνει τον πελάτη ότι η παραγγελία έχει διευθετηθεί.

Οι διακομιστές (Servers) όπως γίνεται αντιληπτό είναι παθητικοί, και ο ρόλος τους επικεντρώνεται στο να περιμένουν ερωτήματα από τον πελάτη. Κατά τη διάρκεια αυτών των περιόδων αναμονής οι διακομιστές μπορούν να εκτελέσουν άλλες εργασίες ή να εκτελούν εργασίες συντήρησης. Σε αντίθεση με τον πελάτη, ο διακομιστής πρέπει να τρέχει συνεχώς, επειδή οι πελάτες μπορούν να ζητήσουν μια υπηρεσία ανά πάσα στιγμή. Οι πελάτες από την άλλη πλευρά το μόνο που χρειάζεται είναι να τρέχουν όταν χρειάζονται να λάβουν μια απάντηση. Πολλές εφαρμογές διακομιστή επιτρέπουν πολλαπλούς πελάτες να ζητήσουν μια υπηρεσία. Για παράδειγμα ενώ διαβάζουμε μια ιστοσελίδα κάποιοι άλλοι ζητούν να ανοίξουν και να διαβάσουν την ίδια με εμάς σελίδα.

1.3 Τι ονομάζουμε Web εφαρμογή;

Σε αντίθεση με την πελάτη-διακομιστή (client-server) μια εφαρμογή web μπορεί να είναι μια οποιαδήποτε εφαρμογή η οποία χρησιμοποιεί ένα πρόγραμμα περιήγησης ως πελάτη. Τι σημαίνει αυτό σε σύγκριση με την πελάτη-διακομιστή (client-server); Αυτό που ισχύει είναι ότι μια web εφαρμογή τρέχει μόνο σε ένα πρόγραμμα περιήγησης το οποίο και χρησιμοποιεί ως client, ενώ μια πελάτη-διακομιστή (client-server) είναι μια εφαρμογή η οποία μπορεί να τρέχει και σε έναν φυλλομετρητή (browser) αλλά και ανεξάρτητα από αυτόν ως ένα πρόγραμμα. Η εφαρμογή μπορεί να είναι τόσο απλή όσο ένας πίνακας μηνυμάτων, ή τόσο σύνθετη όσο ένας επεξεργαστής κειμένου ή ένα υπολογιστικό φύλλο.

1.3.1 Ποια είναι τα πλεονεκτήματα μιας Web εφαρμογής;

Μια εφαρμογή αυτού του είδους απαλλάσσει τον κατασκευαστή από την ευθύνη του να κατασκευάσει έναν client που να μπορεί να τρέχει σε ένα συγκεκριμένο λειτουργικό σύστημα. Δεδομένου του ότι η εφαρμογή (client) τρέχει σε έναν web browser (διακομιστή), μπορεί να κάνει χρήση είτε ενός υπολογιστή Mac είτε ένας IBM. Το ίδιο εύκολα μπορούν να χρησιμοποιούν Windows XP ή Windows Vista. Μπορούν ακόμη και να χρησιμοποιούν τον Internet Explorer ή Firefox, αν και κάποιες εφαρμογές απαιτούν ένα συγκεκριμένο πρόγραμμα περιήγησης (browser) στο Web.

Για να μπορεί να λειτουργήσει μια τέτοιου είδους εφαρμογή, συνδυάζει γλώσσες επικοινωνίας με τον server (server side script) όπως είναι η ASP [27] και PHP [17] μαζί με γλώσσες client-side script όπως είναι HTML [23-24] και η JavaScript [8]. Οι client-side script χρησιμεύουν στο να προβάλουν και να παρουσιάζουν τις πληροφορίες, και οι server side script είναι αυτές που βοηθούν στην επικοινωνία του πελάτη με τον Server.

1.3.2 Ποιο είναι το μέλλον των Web εφαρμογών;

Κατά την παρουσίαση των web εφαρμογών γίνεται αντιληπτό ότι μοιάζουν πολύ με την εφαρμογή πελάτη-διακομιστή (client-server). Γεγονός είναι ότι όχι μόνο μοιάζουν αλλά οι web εφαρμογές κάνουν χρήση της αρχιτεκτονικής *Πελάτη-διακομιστή (client-server)* η οποία παρουσιάστηκε σε προηγούμενη παράγραφο. Το ταχυδρομείο στο διαδίκτυο είναι ένα παράδειγμα αυτού του γεγονότος, με εταιρείες όπως η Yahoo και η MSN που προσφέρουν web-based email σε πελάτες.

Αυτό που θα αποτελέσει την επανάσταση στις εφαρμογές αυτού του είδους, είναι η δημιουργία τους με τέτοιο τρόπο ώστε να μην έχουν ανάγκη την χρήση κάποιου server για την αποθήκευση των πληροφοριών τους. Τα βήματα προς την κατεύθυνση αυτή γίνονται με γοργό ρυθμό, και μέρος της φιλοσοφίας αυτής αποτελεί και η εφαρμογή που συνοδεύει την εργασία η οποία κάνει χρήση όσο το δυνατό λιγότερο της επικοινωνίας με τον διακομιστή (Server). Αυτό πετυχαίνεται με την χρήση τεχνολογιών όπως το Ajax και το jQuery, τα οποία μας βοηθούν να επεξεργαζόμαστε όλες τις πληροφορίες στην μεριά του χρήστη απασχολώντας στο ελάχιστο των server. Πολλές είναι οι εφαρμογές που μέρος της πληροφορίας την αποθηκεύουν στον υπολογιστή μας και όχι στο server τα γνωστά (cookies).

Αν έχετε δει το νέο ηλεκτρονικό ταχυδρομείο Gmail ή το Yahoo, έχετε δει πόσο εξελιγμένες έχουν γίνει οι εφαρμογές web τα τελευταία χρόνια. Μεγάλο μέρος αυτής της εξέλιξης οφείλεται στο AJAX, το οποίο είναι ένα μοντέλο προγραμματισμού για τη δημιουργία εφαρμογών web. Το Google, Microsoft Office Live, και WebEx WebOffice είναι παραδείγματα νεότερης γενιάς διαδικτυακών εφαρμογών.

Μέσω της Java [28], JavaScript, DHTML [29], Flash [30], Silverlight [31] και άλλων τεχνολογιών, είναι δυνατή πλέον η κατασκευή εφαρμογών ειδικών όπως η ζωγραφική στην οθόνη και η αναπαραγωγή ήχου. Πολλοί προγραμματιστές κάνουν χρήση client-side scripting για να δώσουν στις εφαρμογές μεγαλύτερη ζωντάνια, και να αποκτήσουν οι χρήστες την δυνατότητα αλληλεπίδρασης χωρίς να γίνεται ανανέωση της σελίδας. Στο σημείο αυτό θα πρέπει να τονίσουμε ότι η εφαρμογή που συνοδεύει την εργασία αυτή δεν κάνει ούτε μία ανανέωση. Μεγάλο ρόλο στην επίτευξη του γεγονότος αυτού παίζει η ανάπτυξη μεθόδων οι οποίες συνδυάζουν τις client-side scripting γλώσσες με τις server-side. Μία μέθοδος που το πετυχαίνει αυτό και αποτελεί την βίβλο όλων των προγραμματιστών είναι το Ajax, για την οποία έχει αφιερωθεί ένα ολόκληρο κεφάλαιο στην εργασία αυτή και την οποία χρησιμοποιούμε εξολοκλήρου στην εφαρμογή μας.

1.4 Ποιές είναι οι διαφορές μεταξύ των δύο αυτών εφαρμογών;

Κατά την πορεία της περιγραφής των δύο αυτών τεχνολογιών είμαστε σίγουροι πως μάλλον μπερδευτήκατε. Παρακάτω θα προσπαθήσουμε να εντοπίσουμε τις κυριότερες διαφορές τους, με σκοπό να πάρετε μια πιο ξεκάθαρη εικόνα.

- Οι εφαρμογές πελάτη-διακομιστή (client-server) ακολουθούν μια αρχιτεκτονική δύο επιπέδων (two tier architecture), ενώ μια web εφαρμογή κάνει χρήση της αρχιτεκτονικής τριών επιπέδων ή ακόμα και n επιπέδων (n tier architecture).
- Σε εφαρμογές client server δεν χρησιμοποιούμε κάποιον web server (διακομιστή), από την άλλη σε εφαρμογές web θα πρέπει να χρησιμοποιηθεί.
- Τα θέματα ασφάλειας είναι πολύ λιγότερα σε εφαρμογές πελάτη-διακομιστή (client-server), εξαιτίας του μικρού αριθμού των χρηστών. Στην περίπτωση ωστόσο των web εφαρμογών είναι πολύ σημαντική η ασφάλεια του διακομιστή, για τον λόγο του ότι υπάρχει μεγάλος αριθμός χρηστών.
- Η απόδοση είναι κάτι για το οποίο δεν δίνεται μεγάλη σημασία στην περίπτωση του πελάτη-διακομιστή (client-server), αντιθέτως είναι πολύ σημαντικό σε μια web εφαρμογή.

- Στην εφαρμογή πελάτη-διακομιστή έχετε δύο διαφορετικά στοιχεία τα οποία θα πρέπει να ελέγξετε. Οι εφαρμογές καταρχάς θα πρέπει να τρέχουν στον διακομιστή, ενώ τα εκτελέσιμα αρχεία σε κάθε πελάτη χωριστά. Θα πρέπει να γίνονται δοκιμές σε γενικές κατηγορίες και στις δύο πλευρές, τη λειτουργικότητα, το φορτίο, και την αλληλεπίδραση πελάτη-διακομιστή (client-server). Το περιβάλλον αυτό χρησιμοποιείται κυρίως σε δίκτυα Intranet. Γνωρίζετε τον αριθμό των πελατών και των εξυπηρετητών και τις θέσεις τους στο σενάριο δοκιμής.
- Όσον αφορά τις δοκιμές οι οποίες θα πρέπει να γίνουν σε μια web εφαρμογή είναι λίγο πολύπλοκες, και αυτό γιατί δεν υπάρχει μεγάλος έλεγχος πάνω στην εφαρμογή. Για παράδειγμα η εφαρμογή βρίσκεται σε έναν διακομιστή η θέση του οποίου μπορεί να είναι ή να μην είναι γνωστή και σε κανέναν πελάτη δεν υπάρχει κάποιο εκτελέσιμο αρχείο. Έτσι οι εφαρμογές αυτές θα πρέπει να ελεγχθούν τόσο σε διαφορετικούς browsers όσο και σε διαφορετικά λειτουργικά συστήματα.
- Μια client –server εφαρμογή έχει πρόβλημα σε περίπτωση που ο διακομιστής δεν μπορέσει να εξυπηρετήσει κάποιο αίτημα δεν θα μπορέσει η εφαρμογή να ολοκληρωθεί. Στην περίπτωση της web εφαρμογής τέτοια προβλήματα αντιμετωπίζονται και ξεπερνιούνται.

1.5 Τι είναι το W3C

Η μαγική λέξη W3C [21], αποτελεί συντομογραφία των αγγλικών λέξεων World Wide Web Consortium και δημιουργήθηκε τον Οκτώβριο του 1994 από τον Tim Berners-Lee. Οι περισσότεροι γνωρίζουμε το World Wide Web (WWW), που ξεκίνησε ως ένα έργο στο Ευρωπαϊκού Οργανισμού Πυρηνικών Ερευνών (CERN), όπου ο Tim Berners-Lee ήταν αυτός που το οραματίστηκε. Το W3C δημιουργήθηκε το 1994 ως μια συνεργασία μεταξύ του Ινστιτούτου Τεχνολογίας της Μασαχουσέτης (MIT) και του Ευρωπαϊκού Οργανισμού Πυρηνικών Ερευνών (CERN), με την υποστήριξη από το Υπουργείο Άμυνας των ΗΠΑ και της Ευρωπαϊκής Επιτροπής Ερευνών. Ο τρόπος με τον οποίο είναι οργανωμένο το W3C,

είναι με την μορφή ενός οργανισμού με διάφορα μέλη. Ο σκοπός της λειτουργίας του είναι αυτός για τον οποίο αρχικά δημιουργήθηκε, δηλαδή να εξασφαλιστεί η συμβατότητα και η συμφωνία μεταξύ των μελών της βιομηχανίας στην υιοθέτηση των νέων προτύπων. Για παράδειγμα πριν από την δημιουργία του, διάφορες εκδόσεις της HTML, κυκλοφορούσαν από διαφορετικές εταιρείες με αποτέλεσμα να υπάρχει μια ασυνέπεια μεταξύ των διάφορων ιστοσελίδων, πράγμα που δημιουργούσε προβλήματα στους φυλλομετρητές (Browsers). Η κοινοπραξία δημιουργήθηκε για να κάνει όλους αυτούς τους κατασκευαστές (εταιρείες) να συμφωνήσουν σε ένα σύνολο βασικών αρχών και των χαρακτηριστικών που θα υποστηρίζονται από όλους.

1.6 Ποιες είναι οι νέες τεχνολογίες δημιουργίας δικτυακών εφαρμογών;

Σε προηγούμενη ενότητα αναφέραμε λεκτικά κάποιες τεχνολογίες που χρησιμοποιούμε στην κατασκευή σύγχρονων ιστοσελίδων, στο σημείο αυτό θα κάνουμε μια αναφορά στις τεχνολογίες που θα κεντρίσουν το ενδιαφέρον και θα πρωταγωνιστήσουν τόσο κατά την ανάγνωση αυτής της διπλωματικής εργασίας, όσο και γενικότερα στην καθημερινότητα μας μέσα από τις εφαρμογές του διαδικτύου.

Το CSS3 αποτελεί το κλειδί που ξεκλειδώνει διάφορες νέες επιλογές σχετικά με το στυλ στο διαδίκτυο, όπως τα πολλαπλά φόντα, την καλύτερη δυνατότητα να επιλέγουμε το ύφος με μεγαλύτερη ακρίβεια, να δημιουργούμε σκιάσεις και κλιμακωτές διαβαθμίσεις στο χρώμα, χωρίς καμία εξάρτηση από στατικά γραφικά. Δεν θα μπορούσαμε να αγνοήσουμε το γεγονός της δυνατότητας που μας παρέχει όπως η υποστήριξη για στρογγυλοποίηση γωνιών χωρίς την ανάγκη για περίπλοκες τεχνικές ή την χρήση JavaScript. Η σημασία που έχει και θα συνεχίσει να αναπτύσσεται στα επόμενα χρόνια η τεχνολογία αυτή μας έκανε να την ξεχωρίσουμε και να την μελετήσουμε στο κεφάλαιο 3 αυτής της εργασίας.

Η HTML5 αργά αλλά σίγουρα αλλάζει τον τρόπο κατασκευής των σελίδων μας, φέρνοντάς μας πιο κοντά στο Άγιο Δισκοπότηρο του Σημασιολογικού Ιστού (semantic web). Η υποστήριξη ανοικτής μορφής πολυμέσων, όπως βίντεο και ήχου, μας παρέχει καλύτερους τρόπους για να λειτουργούμε σε συνεργασία με το περιεχόμενο ενός δικτυακού

τόπου. Είναι σίγουρο πως τόσο η HTML5 και το CSS3, θα αλλάξουν τον τρόπο που μέχρι τώρα αντιλαμβανόμασταν το διαδίκτυο, γιατί είναι εγγυημένο πως θα κάνουν το web πιο καλαίσθητο μέρος.

Όπως βέβαια είναι ώριμο να συμβαίνει, η HTML5 και το CSS3, έχουν είδη αρχίσει να αλωνίζουν στα χωράφια της JavaScript, έτσι και αυτή για να εξελιχθεί και να συνεχίσει να αποτελεί ένα από τα δυνατότερα εργαλεία προγραμματισμού στο διαδίκτυο, έχωσε το δάκτυλο της στο μέλι που ονομάζεται Flash. Η ανάπτυξη των πλαισίων (Frameworks), όπως η jQuery έδωσε απεριόριστες δυνατότητες για διάδραση στην πλευρά του χρήστη (client side interaction) ο οποίος βιώνει πλέον μια ασύγχρονη / απρόσκοπτη πραγματικότητα. Αυτό οδηγεί σε πιο εύκολες υλοποιήσεις των διαδικτυακών εφαρμογών, οι οποίες με τη σειρά τους, αυξάνουν τον ανταγωνισμό, ο οποίος με τη σειρά του, οδηγεί στην καινοτομία.

Επειδή το CSS3 και η HTML5 θα αντικαταστήσουν πολλά από αυτά που η JavaScript κάνει τώρα, θα δούμε ένα κύμα εξέλιξης της JavaScript που θα επικεντρώνεται αποκλειστικά στο χειρισμό της λογικής του προγραμματισμού στην πλευρά του πελάτη. Με τη νέα σημαντική αναθεώρησης των JavaScript προδιαγραφών, θα δούμε μια πρόοδο προς το καλύτερο των web εφαρμογών που μπορούν να αλληλεπιδρούν πολύ καλύτερα με άλλες εφαρμογές web. Όλες αυτές οι αλλαγές θα παρουσιαστούν πιο διεξοδικά στο κεφάλαιο 2, που αποτελεί μια περιγραφή του εργαλείου JQuery.

Οι παραπάνω τεχνολογίες δεν έρχονται όπως και αναφέραμε να λειτουργήσουν αυτόνομα και ανεξάρτητα μεταξύ τους, σκοπός της εξέλιξης που της συνοδεύει, είναι η ανάπτυξη εργαλείων κάτω από τα οποία θα δημιουργηθεί ένα πλαίσιο συνδυασμού τους. Το DHTML, LAMP, WAMP, XML [26], και το AJAX είναι τεχνολογίες οι οποίες πραγματοποιούν τον παραπάνω σκοπό. Στο AJAX έχουμε αφιερώσει ένα ολόκληρο κεφάλαιο στην εργασία αυτή, γιατί θεωρούμε ότι είναι ένα από τα πιο ισχυρά όπλα τα οποία συνδυάζει, γλώσσες επικοινωνίας με τον server όπως η PHP, και η JavaScript, με τέτοιο τρόπο και σε τέτοιο επίπεδο που μας παρέχεται η δυνατότητα να ενεργούμε πάνω σε αντικείμενα στο επίπεδο του χρήστη. Περισσότερα και πιο αναλυτικά θα τα περιγράψουμε στο κεφάλαιο 4.

Δεν θα μπορούσαμε να κλείσουμε το κεφάλαιο αυτό εάν δεν αναφερόμασταν σε μερικούς ορισμούς εννοιών οι οποίες θα χρησιμοποιηθούν συχνά κατά την πορεία της εργασίας αυτής.

1.7 Τι είναι το WEB 2.0 ;

Τυπικά, το όνομα του Web 2.0 [32] χρησιμοποιείται από προγραμματιστές υπολογιστών για να αναφερθούν σε ένα συνδυασμό α) της βελτίωσης της επικοινωνίας μεταξύ των ανθρώπων μέσω της κοινωνικής δικτύωσης τεχνολογιών, β) της βελτίωσης της επικοινωνίας μεταξύ των διαφορετικών εφαρμογών λογισμικού μέσω ανοικτών προτύπων στον Παγκόσμιο Ιστό με τα οποία περιγράφεται η πρόσβαση στα δεδομένα, και γ) η βελτίωση των διεπαφών Web που μιμούνται το πραγματικό χρόνο απόκρισης σε ένα παράθυρο του προγράμματος περιήγησης.

1.8 Ποιες είναι οι γλώσσες προγραμματισμού ιστοσελίδων;

Οι γλώσσες που χρησιμοποιούνται για την αποτύπωση των περιεχομένων μιας ιστοσελίδας είναι η HTML, η οποία έχει φτάσει στην έκδοση 5.0 και αποτελεί από τα ισχυρότερα κομμάτια αυτού που στις μέρες μας ονομάζουμε δυναμική ιστοσελίδα. Επέκταση της αποτελεί η XHTML. Ένας λόγος για τον οποίο επινοήθηκε είναι για να υπάρξει σαφής διαχωρισμός μεταξύ της εμφάνισης του εγγράφου και της δομής του. Η HTML καθορίζει τόσο την παρουσίαση όσο και την δομή, για παράδειγμα που θα τοποθετείται ένα αντικείμενο.

1.9 Ποιοι είναι οι Web Servers ;

Στο κεφάλαιο 5 γίνεται αναφορά για τον διακομιστή που χρησιμοποιούμε για την ανάπτυξη της εφαρμογής που συνοδεύει την εργασία αυτή. Θα μπορούσαμε να δώσουμε μια μικρή γεύση και να πούμε ότι οι δύο δημοφιλέστεροι web servers είναι ο Apache και ο Internet Information Server (IIS) της Microsoft. Ο IIS εξαρτάται πλήρως από το

περιβάλλον των Windows εφόσον σχεδιάστηκε για να συνεργάζεται με τα υπόλοιπα προϊόντα της Microsoft. Όσο για τον Apache θα μιλήσουμε αναλυτικότερα στο κεφάλαιο 5.

1.10 Τι είναι μια βάση δεδομένων ;

Βάση δεδομένων ονομάζουμε μια οργανωμένη συλλογή αντικειμένων. Υπάρχουν διαφορετικές στρατηγικές οργάνωσης δεδομένων οι οποίες διασφαλίζουν εύκολη πρόσβαση στα δεδομένα αυτά. Στις μέρες μας τα πιο δημοφιλή συστήματα βάσεων δεδομένων είναι τα σχεσιακά μοντέλα βάσεων δεδομένων. Η γλώσσα SQL χρησιμοποιείται παγκοσμίως για την ανάκληση δεδομένων και την επεξεργασία τους από τις βάσεις αυτές. Ορισμένες από τις πιο δημοφιλείς σχεσιακές βάσεις δεδομένων είναι οι Microsoft SQL server, Oracle, Sybase, IBM DB2, Informix, PostgreSQL, και η MySQL την οποία και χρησιμοποιούμε για την εφαρμογή μας και την περιγράφουμε στο κεφάλαιο 5.

1.10.1 Τι είναι οι server –side γλώσσες ;

Οι πιο γνωστές γλώσσες της κατηγορίας αυτής είναι η PHP (Php Hypertext Preprocessor) και η ASP (Active Server Pages). Η ASP η γλώσσα που αναπτύχθηκε από την Microsoft για την κατασκευή δυναμικών ιστοσελίδων. Αρχικά κυκλοφόρησε ως με τον για τον Internet Information Services (IIS) μέσω του Windows NT 4.0 (περίπου 1998), στη συνέχεια συμπεριλήφθηκε ως δωρεάν στοιχείο του Windows Server (μετά την αρχική κυκλοφορία του Windows 2000 Server). Την PHP την οποία και χρησιμοποιούμε στην εφαρμογή μας θα την παρουσιάσουμε πιο αναλυτικά στο κεφάλαιο 5 της εργασίας.

1.11 Τι είναι το IDE ;

Όλα τα εργαλεία προγραμματισμού θα μπορούσαμε να τα χρησιμοποιήσουμε γράφοντας κώδικα με σε έναν κειμενογράφο, και αποθηκεύοντας τον κώδικα αυτό με την κατάλληλη κατάληξη (για παράδειγμα. php . asp. html). Το γεγονός αυτό δεν θα δημιουργούσε κανένα πρόβλημα στις εφαρμογές που θα αναπτύσσαμε, απλά θα καθιστούσε πιο χρονοβόρο τον χρόνο δημιουργίας τους. Τα IDE (integrated Development Environments) είναι

ολοκληρωμένα περιβάλλοντα στα οποία μπορεί ο χρήστης να σχεδιάσει και να υλοποιήσει την εφαρμογή του, και ταυτόχρονα να βλέπει πως αυτή θα φαίνεται στον όταν δημοσιευθεί. Επιπλέον του παρέχει την δυνατότητα να συνδυάζει διάφορες τεχνολογίες κάτω από το ίδιο περιβάλλον και να επιβλέπει την αλληλεπίδραση τους. Ένα από τα πιο γνωστά περιβάλλοντα για τον σκοπό αυτό είναι το Adobe Dreamweaver, το Zend για την Php καθώς και Silverlight της Microsoft. Υπάρχουν πολλά περισσότερα αρκεί κάποιος να έχει την διάθεση και τον χρόνο να τα αναζητήσει στο διαδίκτυο.

1.12 Η εργασία βήμα-βήμα

Στην συνέχεια της διπλωματικής εργασίας:

στο Κεφάλαιο 2 θα κάνουμε μια αναφορά στην ανάλυση των απαιτήσεων και την σχεδίαση μιας εφαρμογής με την μεθοδολογία ICONIX,

στο Κεφάλαιο 3 θα μπούμε σε πιο βαθιά νερά και θα μελετήσουμε όσο το δυνατό πιο διεξοδικά το \$ (jQuery),

στο Κεφάλαιο 4 θα μιλήσουμε για το CSS3 και θα κάνουμε ένα πέρασμα από την HTML5 και τα Fontface,

στο Κεφάλαιο 5 θα παρουσιάσουμε την τεχνολογία Ajax, για την οποία αξίζει τον κόπο να αφιερωθεί ολόκληρο βιβλίο,

στο Κεφάλαιο 6 θα μιλήσουμε για τα MySQL, Apache, και Php και θα αναπτύξουμε μεθοδολογία εγκατάστασης του server της εφαρμογής, παραμετροποίησης του, και μεταφοράς της βάσης,

και στο Κεφάλαιο 7 θα παρουσιάσουμε το αποτέλεσμα της εργασίας μας περιληπτικά, θα παρουσιάσουμε την εφαρμογή μας, θα τονίσουμε ορισμένες λεπτομέρειες σχετικά με την εργασία και θα σας αποχαιρετίσουμε ελπίζοντας να είστε σε θέση να εφαρμόσετε τα παραδείγματα που παρουσιάζονται στα διάφορα κεφάλαια τα οποία προέρχονται από τον κώδικα που συνοδεύει την διπλωματική εργασία δίνοντας ζωή στις δικές σας εφαρμογές.

2

Ανάλυση απαιτήσεων και σχεδίαση

Σκοπός

Σε αυτό το Κεφάλαιο θα ασχοληθούμε με:

- τη χρήση της ICONIX και τα βήματα της,
- την παρουσίαση των σταδίων ανάπτυξης,
- την καταγραφή των απαιτήσεων,
- την ανάλυση, και
- τη σχεδίαση.

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να κατανοήσει τη χρησιμότητα της ICONIX,
- να μπορεί να χρησιμοποιεί τα διαγράμματα της Ενοποιημένης Γλώσσας Μοντελοποίησης ,
- να μπορεί να αναπτύσσει τη γραφική διασύνδεση χρήστη, και
- να μπορεί να αξιοποιεί τις δυνατότητες εργαλείων υποστήριξης της διαδικασίας

2.1 Παρουσίαση

Σκοπός του κεφαλαίου αυτού είναι η παρουσίαση των σταδίων ανάπτυξης, μιας εφαρμογής η οποία θα χρησιμοποιηθεί ως παράδειγμα για την παρουσίαση, των μεθόδων που θα περιγραφούν στα επόμενα κεφάλαια. Το Κεφάλαιο αυτό έχει ως στόχο αφενός να περιγράψει θεωρητικά τα βήματα της μεθοδολογίας, εστιάζοντας στην καταγραφή των απαιτήσεων, την ανάλυση, τη σχεδίαση αλλά και την υλοποίηση συστήματος. Αφετέρου, παρουσιάζεται η εφαρμογή της μεθοδολογίας στη συγκεκριμένη μελέτη περίπτωσης χρησιμοποιώντας ως εργαλείο τα αντίστοιχα διαγράμματα της Ενοποιημένης Γλώσσας Μοντελοποίησης (UML [33]).

Απαραίτητη προϋπόθεση για την εφαρμογή των παραπάνω είναι η ύπαρξη ενός προβλήματος, το οποίο θα περιγραφεί αναλυτικά παρακάτω.

2.2 Περιγραφή ICONIX

Η ICONIX είναι μια μεθοδολογία ανάπτυξης λογισμικού που επιτρέπει τη συστηματική μετάβαση από τις αρχικές απαιτήσεις όπως αυτές διατυπώνονται από τον πελάτη, στον κώδικα που τελικά υλοποιεί τις απαιτήσεις αυτές. Είναι μια απλούστερη εκδοχή της ευρέως διαδεδομένης Ενοποιημένης Προσέγγισης (Unified Process [34]). Το μείζον χαρακτηριστικό της μεθοδολογίας ICONIX [8] είναι η επαναληπτικότητα. Αφενός, η διαδικασία είναι επαναληπτική διότι επιτρέπει την παραγωγή λειτουργικού κώδικα για κάθε μια περίπτωση χρήσης του συστήματος ξεχωριστά.

Το ζητούμενο είναι από τις απαιτήσεις του χρήστη να παραχθεί το τελικό προϊόν, δηλαδή ο λειτουργικός κώδικας. Για να καταλήξουμε στο επιθυμητό αποτέλεσμα η μεθοδολογία ακολουθεί τις παρακάτω φάσεις :

- Ανάλυση των απαιτήσεων.
- Ανάλυση – Αρχική Σχεδίαση.
- Σχεδίαση.
- Υλοποίηση.

2.3 Παρουσίαση Εφαρμογής

Το λογισμικό που πρόκειται να αναπτυχθεί αφορά σε ένα σύστημα ανάρτησης προσφορών στο διαδίκτυο, από διάφορες εταιρείες, με στόχο την προσέγγισης νέων ή και παλιών πελατών. Ο πελάτης, στην προκειμένη περίπτωση, ο επιχειρηματίας, θα μπορεί να δημιουργεί λογαριασμό στην ιστοσελίδα με σκοπό την πλήρη προσβασιμότητα του στις υπηρεσίες που αυτήν προσφέρει. Με την δημιουργία του λογαριασμού του, θα προσφέρεται

η δυνατότητα δημιουργίας προφίλ, το οποίο δεν θα είναι για δημόσια χρήση, θα έχει πρόσβαση σε αυτό (τουλάχιστον στην αρχική έκδοση) μόνο ο ίδιος και θα μπορεί να το ενημερώνει με τα στοιχεία του, και να αναρτά τις προσφορές του.

Ο Πελάτης οποίος αναζητά κάποια προσφορά θα μπορεί να κάνει αναζήτηση μέσω δυνατοτήτων που του προσφέρει η συγκεκριμένη ιστοσελίδα με σκοπό να βρει προσφορές που των ενδιαφέρουν. Θα μπορεί να επιλέγει κατηγορία εταιρείας που τον ενδιαφέρει, καθώς και τοποθεσίας πόλης που θα βρίσκεται η εταιρεία, και με μια φόρμα αναζήτηση θα του παρέχεται η δυνατότητα εύρεσης αυτού που αναζητάει. Όπως επίσης θα υπάρχει τρόπος αναζήτησης συγκεκριμένης εταιρείας με βάση του ονόματος της. Επιπλέον θα του δίνεται η δυνατότητα να αξιολογεί τις προσφορές που του παρουσιάζονται, με Likes.

Ο επιχειρηματίας με την σειρά του θα μπορεί να κάνει χρήση τις αξιολόγησης τις για τις τρέχουσες προσφορές τις εταιρείας τις να τις βελτιώνει ή να τις αποσύρει.

Η ιστοσελίδα με την σειρά τις θα προσφέρει την δυνατότητα στον χρήστη επικοινωνίας για τυχόν παράπονα, ή και παρατηρήσεις. Ο χρήστης θα έχει την δυνατότητα να βλέπει τις πέντε εταιρείες με τις περισσότερες προσφορές, καθώς τις και τις πέντε προσφορές με τα περισσότερα LIKES.

Θα θέλαμε να τονίσουμε ότι σκοπός της δεν είναι η δημιουργία μιας επαγγελματικής ιστοσελίδας η οποία θα μελετά τις περισσότερες περιπτώσεις από αυτές που αναφερθήκαμε παραπάνω, αλλά η δημιουργία μιας ρεαλιστικής εφαρμογής, για την μελέτη και εφαρμογή στα πλαίσια έρευνας τεχνολογιών που αρχίζουν να εφαρμόζονται τις σύγχρονες ιστοσελίδες. Στην πορεία και μετά από την κατανόηση όλων αυτών ο αναγνώστης θα είναι σε θέση να εμπλουτίσει την εφαρμογή αυτή κατά το δοκούν, αφού η περαιτέρω ανάπτυξη θα μπορεί να στηριχθεί τις γνώσεις που θα αποκτηθούν από την ανάγνωση του συγκεκριμένου συγγράμματος.

2.3.1 Καθορισμός Απαιτήσεων

2.3.1.1 Μοντέλο Πεδίου Προβλήματος

Το πρώτο στάδιο ανάπτυξης τις συστήματος λογισμικού για το οποίο έχουν δοθεί οι προδιαγραφές από τον πελάτη, είναι στη μεθοδολογία ICONIX η κατασκευή του μοντέλου του πεδίου προβλήματος. Υπάρχουν αρκετές προσεγγίσεις για την εξαγωγή κλάσεων πεδίου από τις απαιτήσεις υψηλού επιπέδου, που περιλαμβάνουν γραμματική-συντακτική ανάλυση του κειμένου για την εύρεση ουσιαστικών, η χρήση λεξιλογίου από αντίστοιχα συστήματα λογισμικού ή η γνωμοδότηση ειδικών. Σε κάθε περίπτωση προκύπτει μια λίστα εννοιών οι οποίες αποτελούν υποψήφιες κλάσεις πεδίου.

Για το συγκεκριμένο πρόβλημα, τα ουσιαστικά τα οποία αποτελούν τις υποψήφιες κλάσεις του προβλήματος είναι τα παρακάτω:

Λίστα Ουσιαστικών	
Σύστημα Ανάρτησης Προσφορών	Εταιρεία
Πελάτης	Επιχειρηματίας
Ιστοσελίδα	Προφίλ Εταιρείας
Προσφορά	Πόλη
Λογαριασμός του Χρήστη	Όνομα Εταιρείας
Κατηγορία	Τοποθεσία
Βαθμολογία	

Η ανωτέρω λίστα μπορεί να περιοριστεί σημαντικά απαλείφοντας :

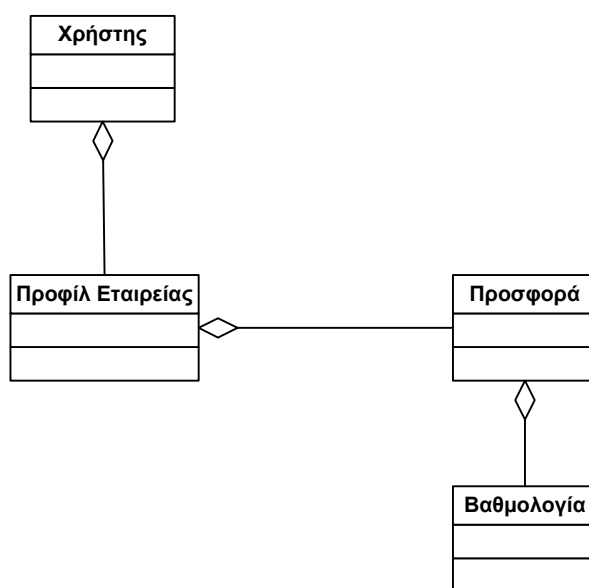
- Αναφορές στο ίδιο το σύστημα λογισμικού.
- Αναφορές σε χειριστές του συστήματος που πρόκειται να αναπτύξουμε καθώς βρίσκονται "έξω" από τα όρια του συστήματος.
- Αναφορές σε οντότητες που βρίσκονται εκτός του πεδίου του προβλήματος και κατά συνέπεια δεν πρόκειται να συμμετάσχουν.
- Ουσιαστικά που πιθανόν να αποτελέσουν ιδιότητες άλλων κλάσεων

Με βάση τις ανωτέρω εξαιρέσεις οι υποψήφιες κλάσεις του πεδίου προβλήματος είναι:

Υποψήφιες Κλάσεις
Προφίλ Εταιρείας
Χρήστης
Προσφορά
Βαθμολογία

Αφού έχουμε εξάγει τις υποψήφιες κλάσεις του προβλήματος το επόμενο στάδιο είναι ο **καθορισμός σχέσεων μεταξύ κλάσεων**. Πρωταρχικός στόχος κατά την κατασκευή του μοντέλου του πεδίου προβλήματος είναι ο εντοπισμός σχέσεων μεταξύ των υποψηφίων κλάσεων. Ωστόσο, καθώς στην αντικειμενοστρεφή σχεδίαση υπάρχουν πολλά είδη σχέσεων μεταξύ κλάσεων, δεν κρίνεται σκόπιμο στο στάδιο αυτό να απεικονιστούν ιδιαίτερες λεπτομέρειες. Συνήθως αρκεί η απεικόνιση στο μοντέλο σχέσεων τύπου "έχει" (*has*) και σχέσεων τύπου "είναι" (*is*). Το πρώτο είδος αναφέρεται σε σχέσεις περιεκτικότητας μεταξύ δύο κλάσεων που στην απλούστερη μορφή καλούνται συναρμολογήσεις (ή συσσωματώσεις). Μια συναρμολόγηση υποδηλώνει ότι αντικείμενα μιας κλάσης περιέχουν αντικείμενα μιας άλλης κλάσης. Η συναρμολόγηση απεικονίζεται ως μια γραμμή μεταξύ των κλάσεων με έναν λευκό ρόμβο στο άκρο της περιέχουσας κλάσης. Το δεύτερο είδος αναφέρεται σε σχέσεις κληρονομικότητας μεταξύ κλάσεων. Μια σχέση κληρονομικότητας υποδηλώνει ότι μια κλάση (υποκλάση) αποτελεί ειδικότερη κατηγορία μιας άλλης (υπερκλάση) και κληρονομεί τις ιδιότητες και τη συμπεριφορά της. Η κληρονομικότητα απεικονίζεται ως μια γραμμή με ένα τρίγωνο στο άκρο της υπερκλάσης.

Από το κείμενο των απαιτήσεων υψηλού επιπέδου, του προβλήματος, προκύπτει ένας λογαριασμός χρήστη, στην προκειμένη περίπτωση θα είναι ο ιδιοκτήτης της εταιρείας, μπορεί να περιλαμβάνει ένα προφίλ εταιρείας, και κάθε προφίλ περιλαμβάνει τις προσφορές που αντιστοιχούν στην εταιρεία που έχει το προφίλ σε αυτή, και κάθε προσφορά θα έχει και κάποια βαθμολογία (LIKES). Μπερδευτήκατε, δεν νομίζω, εάν ναι το παρακάτω σχήμα θα ξεκαθαρίσει όλες τις απορίες. Έτσι με βάση τις πληροφορίες που έχουμε στην φάση αυτήν θα κατασκευάσουμε το Domain Model, το οποίο είναι το ακόλουθο.



Σχήμα 2.1: Διάγραμμα Πεδίου Προβλήματος (αρχικό)

2.3.1.2 Μοντέλο Περιπτώσεων Χρήσης

Το μοντέλο περιπτώσεων χρήσης είναι μείζονος σημασίας στις αντικειμενοστρεφείς μεθοδολογίες ανάπτυξης λογισμικού όπως η ICONIX και η RUP καθώς θέτει τις βάσεις για τη δημιουργία συστημάτων που δίνουν έμφαση στις απαιτήσεις του πελάτη. Στο μοντέλο περιπτώσεων χρήσης καταγράφονται οι απαιτήσεις των χρηστών διερευνώντας εξαντλητικά όλα τα πιθανά σενάρια χρήσης του συστήματος. Μία **περίπτωση χρήσης** (use case) είναι μια ακολουθία ενεργειών που ένας χρήστης του συστήματος (συνήθως πρόκειται για άνθρωπο αλλά μπορεί να είναι οποιαδήποτε εξωτερική οντότητα όπως ένα άλλο σύστημα) πραγματοποιεί στο σύστημα για να επιτύχει ένα συγκεκριμένο σκοπό.

Μία περίπτωση χρήσης απεικονίζεται διαγραμματικά ως μία έλλειψη, μέσα στην οποία αναγράφεται το όνομα της. Το σύνολο των περιπτώσεων χρήσης ενός συστήματος συνιστούν το διάγραμμα περιπτώσεων χρήσης. Στο συγκεκριμένο πρόβλημα δεν θα αναλύσουμε όλες τις περιπτώσεις χρήσης απλά, για λόγους καθαρά εκπαιδευτικού ενδιαφέροντος, θα αναλύσουμε μία περίπτωση χρήσης, οι επόμενες αφήνονται στην ευχέρεια του αναγνώστη

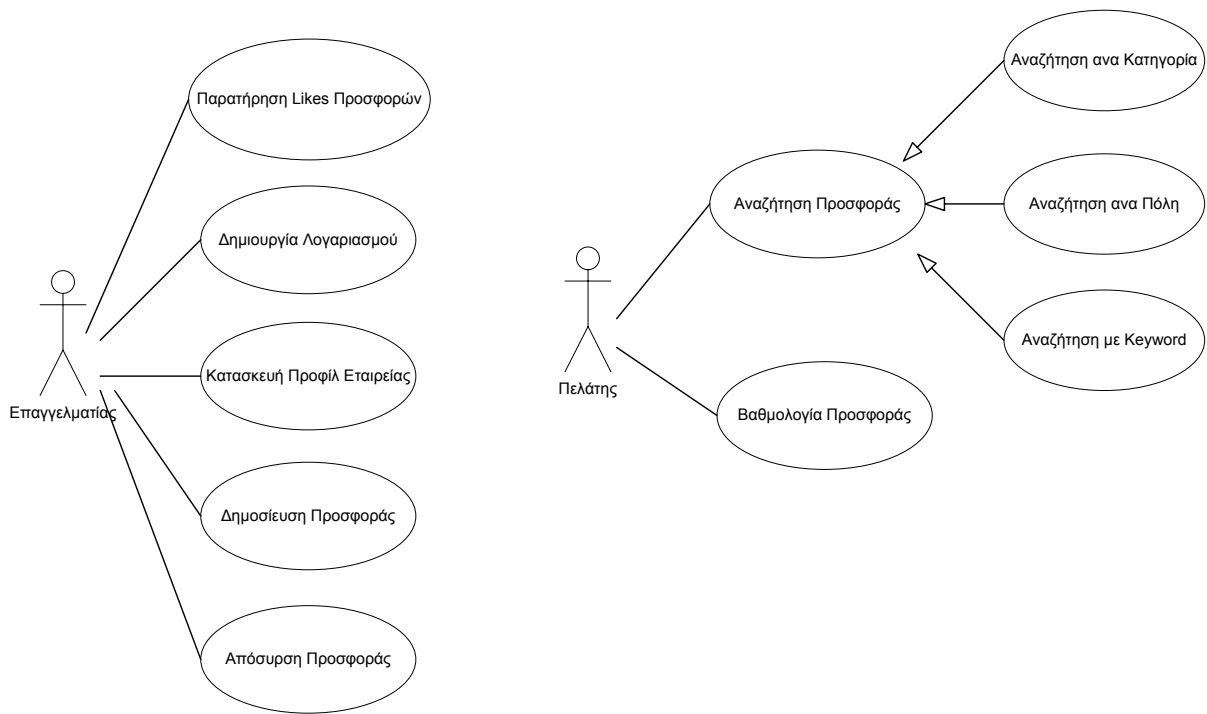
να προσπαθήσει να τις βρει και να τις αναλύσει. Καταρχάς θα πρέπει να βρούμε τους χειριστές του συστήματος. Κάθε χειριστής (actor) αντιπροσωπεύει ένα ρόλο που μπορεί να παίξει ένας χρήστης του συστήματος ή μια οντότητα (όπως π.χ. ένα άλλο σύστημα λογισμικού ή μια βάση δεδομένων), που βρίσκεται εκτός του συστήματος που εξετάζουμε και αλληλεπιδρά με αυτό. Το σύνολο των χειριστών που περιέχονται στο μοντέλο περιπτώσεων χρήσης αντικατοπτρίζει οτιδήποτε χρειάζεται να ανταλλάσει πληροφορίες με το σύστημα.

Με βάση την παρούσα ανάλυση του πεδίου του προβλήματος και την περιγραφή των λειτουργικών προδιαγραφών του υπό σχεδίαση συστήματος ανάρτησης προσφορών, προκύπτουν δύο μόνο χειριστές (σχήμα 2.2) που αντιστοιχούν στον Επαγγελματία (που είναι και η εταιρεία) και στον ενδιαφερόμενο Πελάτη (ο οποίος αναζητά τις προσφορές):



Σχήμα 2.2: Χειριστές Πεδίου Προβλήματος

Για το σύστημα ανάρτησης προσφορών το διάγραμμα περιπτώσεων χρήσης φαίνεται στο σχήμα 2.3.



Σχήμα 2.3: Διάγραμμα Περιπτώσεων Χρήσης Συστήματος

2.3.2 Τεκμηρίωση περιπτώσεων χρήσης

Ο ορισμός μιας περίπτωσης χρήσης περιλαμβάνει όλες τις δυνατές συμπεριφορές, περιλαμβανομένης της κανονικής ή βασικής ροής (όπου για παράδειγμα εκτελούνται όλα όσα αναμένει ο χρήστης για την επίτευξη του στόχου) αλλά και όλων των "εναλλακτικών" ροών, όπου κάτι για παράδειγμα αποκλίνει από την "κανονική" συμπεριφορά. Μια περίπτωση χρήσης είναι επιτυχής όταν προδιαγράφει μόνο "ποια" είναι η συμπεριφορά του συστήματος χωρίς καμία αναφορά στο "πώς" αυτή επιτυγχάνεται, χωρίς δηλαδή περιγραφή τεχνικών λεπτομερειών υλοποίησης.

Στη βιβλιογραφία έχουν προταθεί διάφορα πρότυπα για την τεκμηρίωση των περιπτώσεων χρήσης. Τα πρότυπα μπορούν να ομαδοποιηθούν σε τρεις γενικές κατηγορίες:

1. Απλές περιγραφές κειμένου χωρίς ιδιαίτερη δομή.
2. Πιο εκτεταμένες περιγραφές όπου διατυπώνεται ρητά ποια είναι η βασική ροή και ποιες είναι οι εναλλακτικές ροές.
3. Λεπτομερή πρότυπα.

Στην συνέχεια, ακολουθεί η λεκτική περιγραφή μιας περίπτωσης χρήσης, του συστήματος που αναπτύσσουμε. Θα μπορούσαμε να περιγράψουμε όλες τις περιπτώσεις, σκοπός μας όμως δεν είναι η δημιουργία μιας ολοκληρωμένης πρότασης του συστήματος, αλλά η περιγραφή μόνο της μεθοδολογίας. Η περιγραφή και η ανάλυση των υπολοίπων

περιπτώσεων αφήνεται, στην διάθεση του αναγνώστη ως άσκηση για την κατανόηση της μεθοδολογίας.

Είναι απαραίτητο να κάνουμε τις εξής υποθέσεις, καταρχάς όταν αναφέρεται η φράση "ο χρήστης επιλέγει", θεωρούμε ότι η επιλογή γίνεται μέσω δεικτικής συσκευής. Όπου αναφέρεται η φράση "ο χρήστης εισάγει", θεωρούμε ότι ο χρήστης πληκτρολογεί σε κατάλληλο πεδίο της διεπαφής.

2.3.2.1 Περίπτωση Χρήσης: LogIn Procedure

1ο Πρότυπο

Ο χρήστης του συστήματος, επιλέγει από την κύρια οθόνη του συστήματος το πλήκτρο εγγραφή (LogIn). Το σύστημα εμφανίζει την Οθόνη Εγγραφή Χρήστη, η οποία περιλαμβάνει μια φόρμα στην οποία καλείται ο χρήστης του συστήματος να εισάγει το Username και το Password του. Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης και πιέζει το πλήκτρο Submit. Το σύστημα ελέγχει στον κατάλογο χρηστών, για να διαπιστώσει εάν ο συγκεκριμένος λογαριασμός χρήστη υπάρχει. Εάν υπάρχει το σύστημα στην συνέχεια ελέγχει, εάν ο κωδικός πρόσβασης είναι έγκυρος. Το σύστημα ανακτά όλες τις πληροφορίες σχετικά με τον χρήστη, ξεκινάει μια authenticated session, και προωθεί τον χρήστη στην κύρια οθόνη του συστήματος εμφανίζοντας ένα μήνυμα χαιρετισμού.

Εάν ο χρήστης πληκτρολογήσει λανθασμένα τα στοιχεία του, το σύστημα εμφανίζει μήνυμα «username or password invalid» και τον παρακινεί να τα επανεισάγει.

Εάν ο κωδικός πρόσβασης «password» δεν είναι έγκυρος το σύστημα εμφανίζει μήνυμα «username or password invalid» και τον παρακινεί να τα επανεισάγει.

Ο χρήστης μπορεί να πιέσει το πλήκτρο και να ακυρώσει το LogIn, το σύστημα στην περίπτωση αυτή προωθεί τον χρήστη στην κύρια οθόνη του συστήματος.

2ο Πρότυπο.

Βασική Ροή

1. Ο χρήστης επιλέγει από την κύρια οθόνη το πλήκτρο εγγραφή (LogIn).
2. Το σύστημα εμφανίζει την Οθόνη Εγγραφή Χρήστη, η οποία περιλαμβάνει μια φόρμα στην οποία καλείται ο χρήστης του συστήματος να εισάγει το Username και το Password του.
3. Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης και πιέζει το πλήκτρο Submit.
4. Το σύστημα ελέγχει στον κατάλογο χρηστών, για να διαπιστώσει εάν ο συγκεκριμένος λογαριασμός χρήστη υπάρχει.
5. Το σύστημα στην συνέχεια ελέγχει, εάν ο κωδικός πρόσβασης είναι έγκυρος.

6. Το σύστημα ανακτά όλες τις πληροφορίες σχετικά με τον χρήστη, ξεκινάει μια authenticated session, και προωθεί τον χρήστη στην κύρια οθόνη του συστήματος εμφανίζοντας ένα μήνυμα χαιρετισμού.

7.

Εναλλακτική Ροή 1

4.a.1 Εάν ο χρήστης πληκτρολογήσει λανθασμένα τα στοιχεία του και το σύστημα δεν βρει λογαριασμό με συγκεκριμένα στοιχεία.

4.a.2 Το σύστημα εμφανίζει μήνυμα «username or password invalid» και τον παρακινεί να τα επανεισάγει.

4.a.3 Η περίπτωση χρήσης συνεχίζει από το βήμα 2 της βασικής ροής.

Εναλλακτική Ροή 2

5.β.1 Εάν ο χρήστης πληκτρολογήσει λανθασμένα κωδικός πρόσβασης «password».

5.β.2 Το σύστημα εμφανίζει μήνυμα «username or password invalid» και τον παρακινεί να τα επανεισάγει.

5.β.3 Η περίπτωση χρήσης συνεχίζει από το βήμα 2 της βασικής ροής.

Εναλλακτική Ροή 3

3.a.1 Ο χρήστης μπορεί να πιέσει το πλήκτρο και να ακυρώσει το LogIn, απλά πιέζοντας το X του παραθύρου.

3.a.2 Το σύστημα προωθεί τον χρήστη, στην κύρια οθόνη του συστήματος.

3.a.3 Η περίπτωση χρήσης συνεχίζει από το βήμα 1 της βασικής ροής.

2.3.3 Ανάλυση Ευρωστίας (Robustness Analysis)

Η ανάλυση ευρωστίας (robustness analysis) αποτελεί μια τεχνική, ενταγμένη στη φάση της ανάλυσης απαιτήσεων, για τη μετάβαση από τις περιπτώσεις χρήσης σε ένα λεπτομερές σχέδιο. Με βάση την Μεθοδολογία ICONIX, αποτελεί την γεφύρωση του χάσματος μεταξύ της ανάλυσης (που απαντά στο ερώτημα τι θα κάνει το σύστημα) και της σχεδίασης που (απαντά στο ερώτημα πώς θα το κάνει το σύστημα). Είναι όμως μια εξαιρετική τεχνική για την εκλέπτυνση και αποσαφήνιση του κειμένου των περιπτώσεων χρήσης και τον εντοπισμό ενός αρχικού συνόλου αλληλεπιδρώντων αντικειμένων (κλάσεων) για την ικανοποίηση της ζητούμενης λειτουργικότητας. Η ανάλυση ευρωστίας στοχεύει στον έλεγχο της ορθότητας των περιπτώσεων χρήσης και της εφικτότητας υλοποίησης της συμπεριφοράς του συστήματος, ελέγχει την πληρότητα των περιγραφών των περιπτώσεων χρήσης και υποβοηθά τον εντοπισμό επιπρόσθετων κλάσεων και τον εμπλουτισμό του μοντέλου του πεδίου του προβλήματος πριν το λεπτομερή σχεδιασμό των διαγραμμάτων ακολουθίας.

2.3.3.1 Διάγραμμα Ευρωστίας “ LogIn Procedure ”

Κατά την ανάλυση ευρωστίας το κείμενο των περιπτώσεων χρήσης μεταφράζεται σταδιακά (πρόταση προς πρόταση) σε μια γραφική απεικόνιση κλάσεων και συμπεριφοράς (διάγραμμα ευρωστίας). Κάθε οντότητα η οποία σύμφωνα με το μοντέλο πεδίου προβλήματος (ή σύμφωνα με νεότερη κρίση των αναλυτών-σχεδιαστών) αποτελεί μια κλάση του συστήματος, απεικονίζεται στο διάγραμμα ευρωστίας, κατηγοριοποιώντας την με βάση ένα από τα ακόλουθα τρία στερεότυπα κλάσεων:

- Συνοριακές κλάσεις: κλάσεις που αποτελούν διασύνδεση μεταξύ του συστήματος και του "εξωτερικού κόσμου", δηλαδή των χειριστών του.
- Κλάσεις οντοτήτων: κλάσεις του μοντέλου πεδίου προβλήματος.
- Κλάσεις ελέγχου: κλάσεις που αποτελούν την "κόλλα" μεταξύ των συνοριακών και των κλάσεων οντοτήτων και αναπαριστούν τη συμπεριφορά του συστήματος.

Κατά τη σχεδίαση διαγραμμάτων ευρωστίας οφείλουν να τηρούνται οι ακόλουθοι τρεις απλοί κανόνες:

- Ουσιαστικά (χειριστές, συνοριακές κλάσεις και κλάσεις οντοτήτων) δεν μπορούν να συσχετίζονται απευθείας με άλλα ουσιαστικά.
- Ουσιαστικά μπορούν να συσχετίζονται με ρήματα (κλάσεις ελέγχου) και το αντίθετο.
- Ρήματα μπορούν να συσχετίζονται με άλλα ρήματα.

Για την εξαγωγή του διαγράμματος ευρωστίας, εξετάζουμε αναλυτικά την περίπτωση χρήσης, **LogIn Procedure** σύμφωνα με τα πρότυπα του ICONIX.

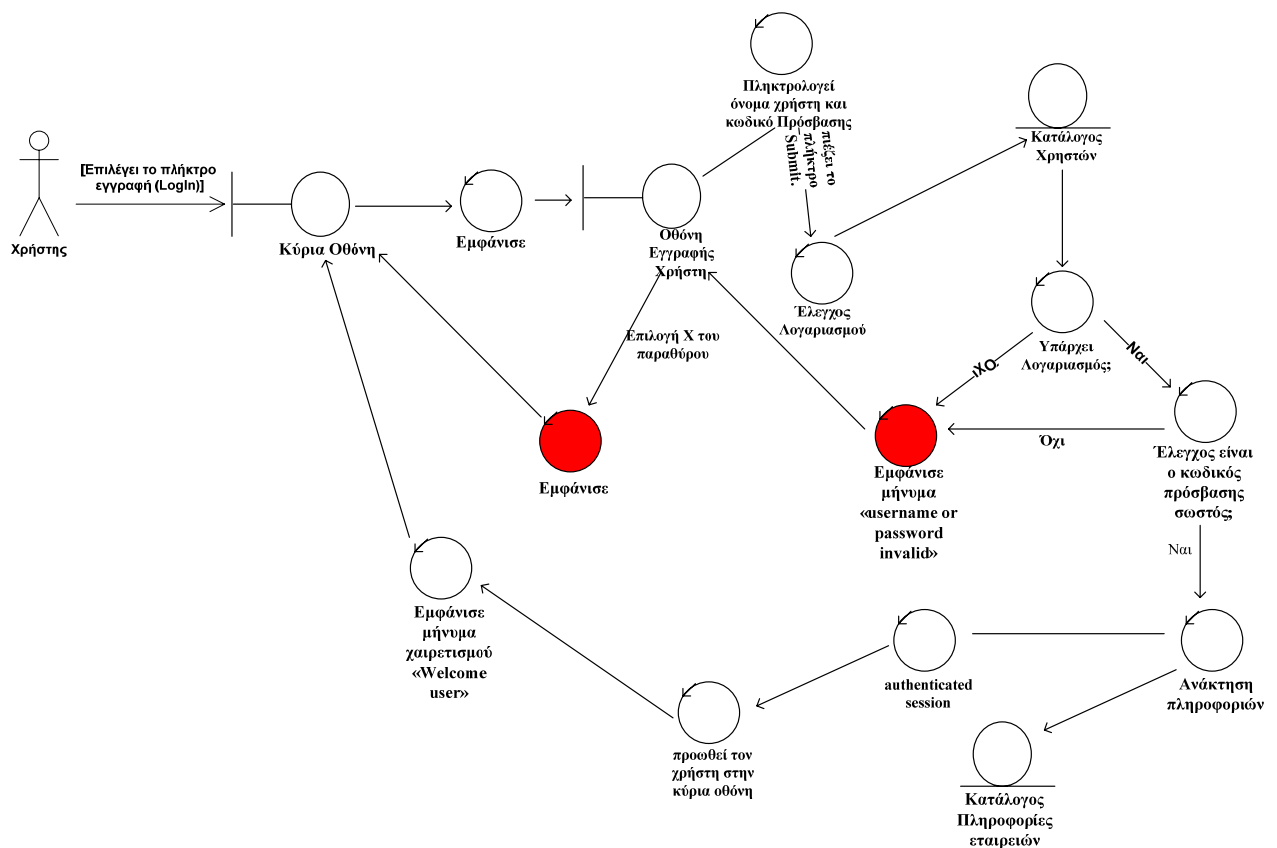
Η φράση " *Ο χρήστης επιλέγει από την κύρια οθόνη το πλήκτρο εγγραφή (LogIn)*" απεικονίζεται ως μια ακμή μεταξύ του χρήστη και της συνοριακής κλάσης που αντιστοιχεί στην Κύρια Οθόνη. Η φράση " *Το σύστημα εμφανίζει την Οθόνη Εγγραφή Χρήστη*" υποδηλώνει την ύπαρξη στο διάγραμμα ευρωστίας μιας συνοριακής κλάσης για την " *Οθόνη Εγγραφή Χρήστη* ". Τηρώντας τον κανόνα που δεν επιτρέπει την απευθείας συσχέτιση μεταξύ συνοριακών κλάσεων, εισάγεται μια κλάση ελέγχου που αντιστοιχεί στην ενέργεια της εμφάνισης της οθόνης (από την Κύρια Οθόνη), ως απόκριση στην επιλογή του χρήστη.

Η φράση " *Το σύστημα ελέγχει στον κατάλογο χρηστών, για να διαπιστώσει εάν ο συγκεκριμένος λογαριασμός χρήστη υπάρχει*", επιβάλλει την εισαγωγή μια κλάσης οντότητας που αντιστοιχεί στον Κατάλογο Χρηστών και μιας κλάσης ελέγχου που αντιστοιχεί κλάση ελέγχου που ονομάζεται υπάρχει ο λογαριασμός από τον κατάλογο.

Οι φράσεις " *Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης*" και πατάει το πλήκτρο " *Submit* " και το " *Το σύστημα ελέγχει στον κατάλογο χρηστών, για να διαπιστώσει εάν ο συγκεκριμένος λογαριασμός χρήστη υπάρχει*". Στην περίπτωση αυτήν έχουμε την εναλλακτική περίπτωση κατά την οποία ο υπάλληλος πληκτρολογήσει στοιχεία τα οποία δεν

ανταποκρίνονται σε κανέναν λογαριασμό τότε εμφανίζεται μήνυμα προειδοποίησης. Υποθέτουμε ότι ο Κατάλογος Χρηστών περιέχει τους Χρήστες, εταιρείες που χρησιμοποιούν το συγκεκριμένο σύστημα οι οποίοι είναι εγγεγραμμένοι.

Το συνολικό διάγραμμα ευρωστίας με βάση την ανωτέρω ανάλυση παρουσιάζεται στο ακόλουθο σχήμα. Η λύση που προτείνεται δεν είναι η μοναδική.

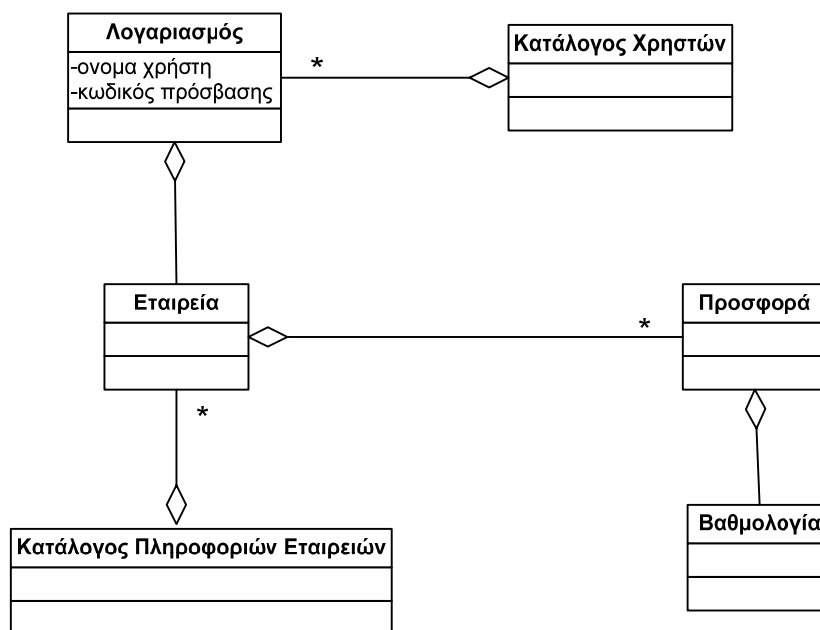


Σχήμα 2.4: Διάγραμμα Ευρωστίας για την Περίπτωση Χρήσης " LogIn Procedure"

2.3.4 Αναθεώρηση του μοντέλου πεδίου προβλήματος

Από τη διερεύνηση των περιπτώσεων χρήσης και τη δημιουργία των διαγραμμάτων ευρωστίας είναι αναμενόμενο και επιθυμητό να εντοπιστούν νέες κλάσεις που δεν συμπεριελήφθησαν αρχικά στο μοντέλο πεδίου προβλήματος. Από την ανάλυση της προηγούμενης περίπτωσης χρήσης προκύπτει ότι στο μοντέλο πεδίου προβλήματος πρέπει να συμπεριληφθεί η κλάση Κατάλογος Χρηστών (*usersCatalog*) και η κλάση Κατάλογος Πληροφοριών Εταιρειών (*CompDetCatalog*). Τέτοιες κλάσεις, που λειτουργούν ως συλλογές αντικειμένων κλάσεων οντοτήτων είναι αναμενόμενες στις περισσότερες εφαρμογές, ακόμα και αν δεν αναφέρονται ρητά στις απαιτήσεις, καθώς επιτρέπουν τον εντοπισμό και την ανάκτηση των αντικειμένων των κλάσεων οντοτήτων. Οι δύο αυτές κλάσεις περιλαμβάνουν

αντικείμενα των κλάσεων *χρήστης* (user) και *προφίλ εταιρείας* (Company's Profile) αντίστοιχα, και για το λόγο αυτό συνδέονται με σχέση συναρμολόγησης με αυτές με πολλαπλότητα *πολλά* στο άκρο των κλάσεων οντοτήτων. Από το διάγραμμα ευρωστίας που περιγράψαμε προκύπτει το αναθεωρημένο διάγραμμα του πεδίου του προβλήματος που φαίνεται στο σχήμα 1.5.



Σχήμα 2.5: Αναθεωρημένο Διάγραμμα Πεδίου Προβλήματος

2.3.5 Σχεδίαση

Με την ολοκλήρωση των σταδίων της ανάλυσης οι προδιαγραφές του συστήματος υπό μορφή περιπτώσεων χρήσης έχουν παγιωθεί και μπορούν να θεωρηθούν πλήρεις, ορθές, λεπτομερείς και σαφείς. Επιπλέον έχουν εντοπιστεί οι περισσότερες κλάσεις, δηλαδή ολοκληρώθηκε το μεγαλύτερο τμήμα της στατικής δομής του λογισμικού. Ο ακριβής καθορισμός του τρόπου με τον οποίο οι κλάσεις (τα αντικείμενά τους) αλληλεπιδρούν μεταξύ τους, δηλαδή ο προσδιορισμός μέρους της δυναμικής συμπεριφοράς του συστήματος, αποτελούν αντικείμενο της σχεδίασης. Ως επακόλουθο αυτής της κατανομής λειτουργικότητας προκύπτει το αναθεωρημένο διάγραμμα κλάσεων όπου εκτός από τις κλάσεις και τις ιδιότητές τους, απεικονίζονται οι μέθοδοί τους και τυχόν νέες σχέσεις μεταξύ τους. Το τελικό αυτό διάγραμμα κλάσεων αποτελεί την είσοδο για την έναρξη της κωδικοποίησης του συστήματος (αν και είναι αναμενόμενο το διάγραμμα κλάσεων να τροποποιηθεί ως ένα βαθμό κατά την υλοποίηση).

Λαμβάνοντας υπόψη την ανάλυση του πεδίου του προβλήματος, την καταγραφή της λειτουργικότητας και την ανάλυση ευρωστίας που πραγματοποιήσατε στα προηγούμενα βήματα, θα συνεχίσουμε στην τελική σχεδίαση του συστήματος παράγοντας το αντίστοιχο

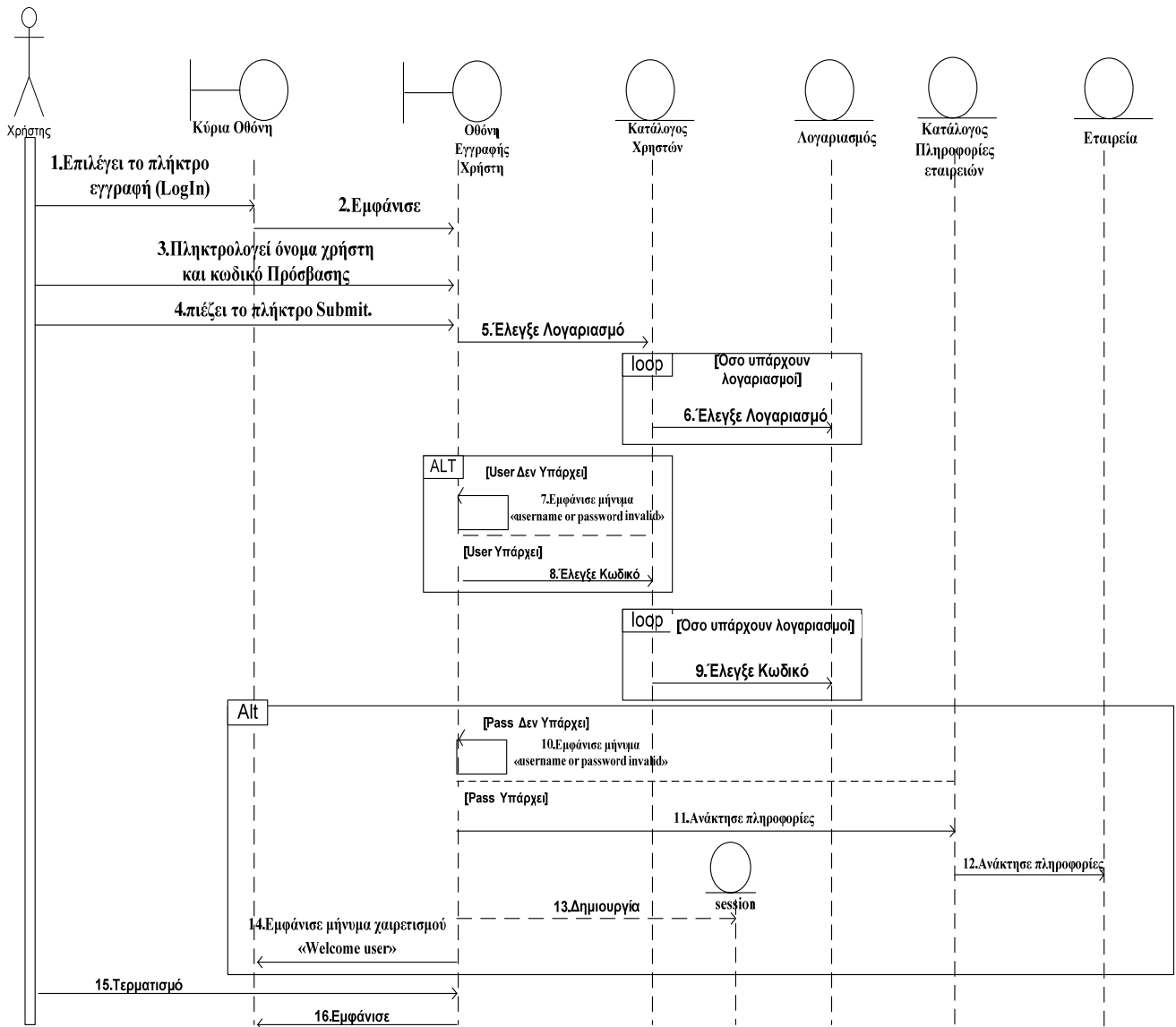
διάγραμμα ακολουθίας (Sequence Diagram). Παράλληλα θα αποτυπώσουμε στο τελικό **διάγραμμα κλάσεων** όλα τα στοιχεία που προέκυψαν από την ανάλυση και σχεδίαση αναφορικά με το συγκεκριμένο τμήμα λειτουργικότητας. Ορισμένες παρατηρήσεις που κρίνονται απαραίτητες να γίνουν για την καλύτερη κατανόηση του διαγράμματος που θα ακολουθήσει είναι οι εξής:

- για κάθε περίπτωση χρήσης, οι χειριστές που εμπλέκονται, τα αντικείμενα που αντιστοιχούν σε συνοριακές κλάσεις αλλά και τα αντικείμενα των κλάσεων οντοτήτων μεταφέρονται αυτούσια στο διάγραμμα ακολουθίας.
- Δεν μεταφέρονται όμως και τα αντικείμενα των κλάσεων ελεγκτών καθώς συνήθως οι κλάσεις αυτές μεταφράζονται σε μηνύματα μεταξύ των αντικειμένων.
- Ο συνηθέστερος τύπος μηνύματος αντιστοιχεί στην κλήση μιας λειτουργίας ενός αντικειμένου και συμβολίζεται ως μια προσανατολισμένη ακμή με το βέλος από τον αποστολέα προς τον παραλήπτη.
- Επάνω από κάθε μήνυμα σημειώνεται το όνομα της λειτουργίας που καλείται καθώς και οι ενδεχόμενες παράμετροι που απαιτούνται.

2.3.5.1 Διάγραμμα Ακολουθίας 1: Δημιουργία Πιάτου

Το διάγραμμα ακολουθίας καταστρώνεται ως εξής: Στον οριζόντιο άξονα των αντικειμένων τοποθετούνται, εξετάζοντας το αντίστοιχο διάγραμμα ευρωστίας, οι χειριστές της περίπτωσης χρήσης (στην προκειμένη περίπτωση ο Χρήστης) τα αντικείμενα των συνοριακών κλάσεων (η Κύρια Οθόνη, η Οθόνη Εγγραφής Χρήστη) και τα αντικείμενα των κλάσεων οντοτήτων (Κατάλογος Χρηστών, Κατάλογος Πληροφοριών Εταιρειών). Όπως αναφέραμε και σε προηγούμενη παράγραφο, οι κλάσεις ελέγχου δεν εμφανίζονται ως αντικείμενα αλλά ως μηνύματα τα οποία ανταλλάσσονται μεταξύ των υπολοίπων κλάσεων. Λαμβάνοντας υπόψη ότι σε ένα διάγραμμα ακολουθίας απεικονίζεται η αλληλεπίδραση μεταξύ χρήστη και συστήματος για ένα συγκεκριμένο σενάριο χρήσης, το πρώτο μήνυμα (με αριθμηση 1) είναι η επιλογή από το χρήστη του πλήκτρου "LogIn" που αποστέλλεται στην Κύρια Οθόνη. Με βάση το διάγραμμα ευρωστίας η Κύρια Οθόνη αποστέλλει το μήνυμα εμφάνισης (αντικείμενο ελεγκτή "εμφάνισε" στην Οθόνη Εγγραφής Χρήστη - αντικείμενο συνοριακής κλάσης). Το πρώτο μήνυμα που αποστέλλει η οθόνη Εγγραφής Χρήστη είναι το αίτημα λήψης ονομάτων συστατικών προς τον Κατάλογο Χρηστών. Ο ίδιος ο κατάλογος δεν περιλαμβάνει μια λίστα των ονομάτων όλων των Χρηστών, απλά περιέχει μια δομή δεδομένων με αναφορές προς τους Λογαριασμούς Χρηστών που περιλαμβάνει. Κατά συνέπεια τα στοιχεία του κάθε Λογαριασμού Χρήστη μπορεί να "αντληθεί" αποστέλλοντας μήνυμα σε κάθε αντικείμενο της κλάσης Λογαριασμός που περιλαμβάνει ο κατάλογος. Για το λόγο αυτό, παρόλο που κάτι τέτοιο δεν καταγράφεται στο διάγραμμα ευρωστίας, επιλέγουμε να απεικονίσουμε στο διάγραμμα ακολουθίας την επαναληπτική αποστολή

μηνυμάτων λήψης ονόματος από το αντικείμενο της κλάσης Κατάλογος Χρηστών προς αντικείμενα της κλάσης Λογαριασμός (το οποίο για το λόγο αυτό προστίθεται στον οριζόντιο άξονα του διαγράμματος ακολουθίας). Σύμφωνα με το διάγραμμα ευρωστίας δημιουργείται authenticated session για την απεικόνιση επιλέγουμε την δημιουργία κλάσης Session. Το διάγραμμα ακολουθίας που προκύπτει φαίνεται στο παρακάτω σχήμα.



Σχήμα 2.6: Διάγραμμα ακολουθίας για την Περίπτωση Χρήσης " Login Procedure"

2.3.6 Διάγραμμα Κλάσεων

Ο κύριος σκοπός δημιουργίας των διαγραμμάτων ακολουθίας είναι η κατανομή της λειτουργικότητας (μεθόδων) στις κλάσεις του συστήματος και κατά δεύτερο λόγο ο εντοπισμός τυχόν κλάσεων, ιδιοτήτων και μεθόδων που δεν αναδείχθηκαν στο στάδιο της

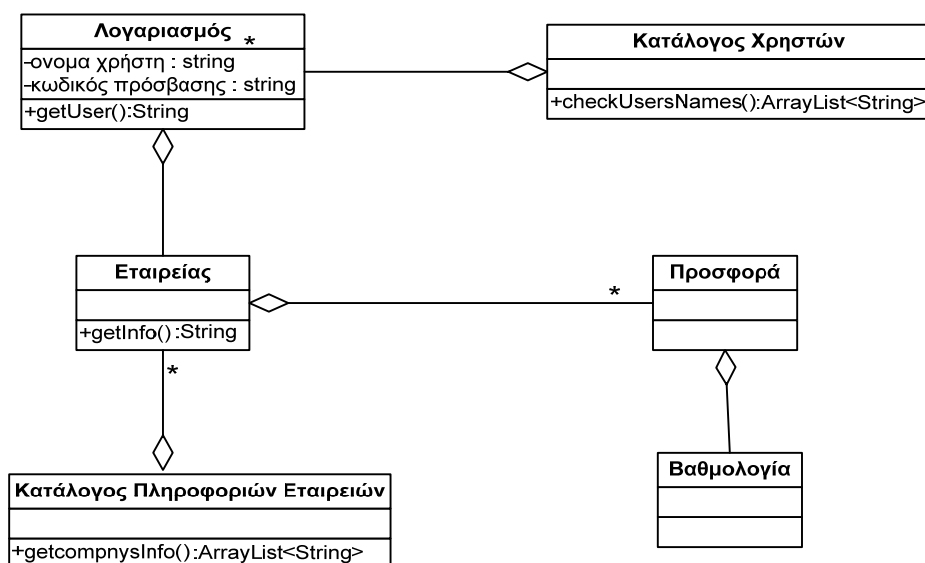
ανάλυσης. Στο αναθεωρημένο διάγραμμα κλάσεων περιλαμβάνονται πλέον οι μέθοδοι κάθε κλάσης με την πλήρη υπογραφή τους, δηλαδή μαζί με τυχόν παραμέτρους που απαιτούνται και τον επιστρεφόμενο τύπο τους. Η λήψη ενός μηνύματος από ένα αντικείμενο μιας κλάσης σε ένα διάγραμμα ακολουθίας υποδηλώνει την ύπαρξη μιας μεθόδου στην κλάση που είναι ο αποδέκτης του μηνύματος. Η μέθοδος συνιστά τη λειτουργία που θα εκτελείται στην κλάση-αποδέκτη με τη λήψη του αντίστοιχου μηνύματος.

Από το διάγραμμα ακολουθίας που αφορά στην περίπτωση χρήσης LogIn προκύπτουν οι εξής μέθοδοι για τις κλάσεις των αντικειμένων που εμφανίζονται στο διάγραμμα (εξαιρώντας τις συνοριακές κλάσεις):

Λογαριασμός	Session	Κατάλογος Χρηστών	Κατάλογος Πληροφοριών Εταιρειών	Εταιρεία
λήψη User getUser ()	καταχώρηση (συστατικό, ποσότητα) storeIngredient (Ingredient,int)	Έλεγχος Users checkUsersNames ()	Ανάκτηση Πληροφοριών getcompnysInfo ()	Λήψη πληροφοριών getInfo ()

Με βάση τις μεθόδους που εντοπίστηκαν κατά την διαδικασία της κατασκευής του διαγράμματος ακολουθίας είναι πλέον δυνατόν να προκύψει το αναθεωρημένο διάγραμμα κλάσεων, το οποίο στο σημείο αυτό "πλησιάζει" αρκετά τον κώδικα που πρόκειται να υλοποιηθεί στη συνέχεια. Το διάγραμμα κλάσεων του συστήματος προκύπτει προσθέτοντας μία προς μία τις μεθόδους που αναφέρονται στους ανωτέρω πίνακες με δημόσια ορατότητα (public visibility).

Το αναθεωρημένο διάγραμμα κλάσεων που προκύπτει είναι το παρακάτω.



Σχήμα 2.7: Αναθεωρημένο Διάγραμμα Πεδίου Προβλήματος

2.4 Συμπέρασμα

Κατά την περιήγηση μας στο κεφάλαιο, αυτό είχαμε την δυνατότητα παρουσίασης της σχεδίασης και ανάλυσης ενός συστήματος. Με την εφαρμογή της αντικειμενοστραφούς μεθοδολογίας ανάλυσης και σχεδίασης, περιγράψαμε με απλά και συστηματικά βήματα πως είναι δυνατόν να μεταβούμε από τις αρχικές απαιτήσεις του χρήστη σε ένα λεπτομερές σχέδιο του συστήματος λογισμικού, έτσι ώστε να είμαστε σε θέση να μπορέσουμε πατώντας πάνω σε αυτό να υλοποιήσουμε τον κώδικα. Στα κεφάλαια που θα ακολουθήσουν θα περιγραφούν οι μέθοδοι που χρησιμοποιήθηκαν, για την υλοποίηση του έργου, με σκοπό όχι την δημιουργία ενός ακόμα εγχειριδίου, (υπάρχουν άπειρα στην αγορά και το διαδίκτυο), αλλά τον σχολιασμό, με παραδείγματα μέσα από την εφαρμογή που έχει δημιουργηθεί, για τα πλεονεκτήματα τους σε σχέση με προγενέστερες μεθόδους διαδικτυακού προγραμματισμού. Θα προσπαθήσουμε μέσα από τον σχολιασμό μέρους του κώδικα να τις παρουσιάσουμε και όσο είναι δυνατόν να παρακινήσουμε τον αναγνώστη στην περαιτέρω μελέτη αυτών.

3

JQuery

Σκοπός

Σε αυτό το Κεφάλαιο θα εξετάσουμε:

- γιατί θα πρέπει να χρησιμοποιούμε JQuery;
- πώς να επιλέγουμε στοιχεία κάνοντας χρήση του JQuery;
- το JQuery έναντι του JavaScript ;

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να κατανοήσει τη χρησιμότητα του JQuery,
- να αποκτήσει δυνατότητα χρήσης του εργαλείου, και
- να μπορεί να παραμετροποιεί έτοιμα Frameworks με σκοπό την αξιοποίηση τους σε διάφορα έργα.

3.1 Παρουσίαση

Ο τίτλος που θα ταίριαζε στο κεφάλαιο αυτό θα μπορούσε να είναι «Πως θα κάνουμε την ζωή μας πιο εύκολη». Αυτό μην σας παρασέρνει δεν πρόκειται για ένα κεφάλαιο συμβουλευτικού χαρακτήρα, το οποίο θα σας δίνει πληροφορίες στο πως θα αντιμετωπίζεται προβλήματα της καθημερινότητας, όπως ατελείωτες ουρές, άγχος της οικονομικής πραγματικότητας κ.α. Αυτό μάλλον το κεφάλαιο θα ήταν δυνατό να το γράψει ένας μάντης ή ένας αστρολόγος και όχι κάποιος κοινός θνητός. Απ'ότι καταλάβατε αστειευόμαστε, και μάλλον το χιούμορ είναι ένας τρόπος να αντιμετωπίσουμε τόσο τις δυσκολίες της ζωής, όσο και τα προβλήματα που αντιμετωπίζουμε στην ανάπτυξη διαδικτυακών εφαρμογών.

Αρκετά με την φιλοσοφία και ας περάσουμε στον κύριο λόγο του κεφαλαίου αυτού που δεν είναι άλλος από το JQuery [25]. Η ερώτηση που καλούμαστε να σας βοηθήσουμε να απαντήσετε, είναι γιατί θα πρέπει ένας web developer να γράφει ατελείωτες γραμμές πολύπλοκου κώδικα όταν το μόνο που θέλει να πετύχει είναι ένας απλός τρόπος διεπαφής με τον χρήστη; Στην μέχρι τώρα βιβλιογραφία δεν υπάρχει κανένας που να υποστηρίζει ότι η πολυπλοκότητα αποτελεί το αναγκαίο κακό για να γράψουμε επιτυχημένες δικτυακές εφαρμογές.

Η δημιουργία του JQuery βασίστηκε στην απαίτηση των προγραμματιστών να κάνουν πάρα πολλά, με μικρή προσπάθεια. Ο σκοπός του κεφαλαίου αυτού είναι να σας βοηθήσει να καταλάβετε πώς να κάνετε πολλά σε μια διαδικτυακή εφαρμογή γράφοντας λιγότερο κώδικα. Για τον λόγο αυτό θα συμπεριλάβουμε παραδείγματα από το διαδίκτυο και φυσικά από την εφαρμογή που πραγματοποιήθηκε για τους σκοπούς του συγγράμματος αυτού. Μέσα από τα παραδείγματα που έχουν επιλεγεί θα μπορέσετε να πάρετε μία πρώτη γεύση για το εργαλείο το οποίο σας παρουσιάζεται, και να πειστείτε για τις ατελείωτες δυνατότητες που σας προσφέρει.

3.1.1 Σε ποιους απευθύνεται

Το κεφάλαιο αυτό απευθύνεται σε προγραμματιστές οι οποίοι επιθυμούν να πάρουν τον έλεγχο από τις JavaScript [8] εφαρμογές τους, και να παράγουν τέλειες και πλούσιες διαδικτυακές εφαρμογές, όσον αφορά την διεπαφή τους με τον χρήστη.

3.2 Τι είναι το JQuery;

Δεν είναι κάτι το οποίο δημιουργήθηκε από πουθενά, όπως σχεδόν τα πάντα στην πληροφορική ακολουθούν μια συνέχεια. Το JQuery, δεν είναι τίποτα άλλο παρά μία βιβλιοθήκη και μία σειρά από Frameworks, της JavaScript. Κάποιος καχύποπτος θα έλεγε και γιατί να ασχοληθώ με αυτό και όχι με JavaScript, το οποίο ήδη ξέρω. Απάντηση δεν θα δώσουμε από τώρα, γιατί ποιος ο λόγος να διαβάσετε αυτό που περιγράφουμε στην συνέχεια. Αλώςτε αυτό δεν κάνουν οι συγγραφείς όλων των Best Sellers;

3.3 Γιατί JQuery;






Στην προσπάθεια σου να προσθέσεις δυναμικότητα σε μια εφαρμογή, τις περισσότερες φορές ακολουθείς την ίδια πατέντα, επιλέγεις ένα αντικείμενο ή μια ομάδα αντικειμένων, και πάνω σε αυτά εφαρμόζεις κάποιες μεθόδους. Κάνοντας χρήση της JavaScript είναι σίγουρο ότι θα πρέπει να γράψεις γραμμές ατελείωτου κώδικα για να πετύχεις κάτι πολύ απλό όπως για παράδειγμα το να κάνεις ένα αντικείμενο αόρατο. Οι εμπνευστές του JQuery κατασκεύασαν βιβλιοθήκες που, απαιτήσεις σαν και την προηγούμενη, αποτελούν πλέον ρουτίνα.

Δεν σας πείσαμε ακόμα. Τι θα λέγατε εάν σας βοηθούσαμε να κατασκευάσετε έναν πίνακα του οποίου το χρώμα των γραμμών εναλλάσσονται με μόνο μισή γραμμή κώδικα ενώ ξέρουμε πολύ καλά ότι κάτι τέτοιο θέλει τουλάχιστον δέκα γραμμές σε JavaScript; Ξέρουμε τώρα πιθανό να λέτε «τι λέει τώρα», κοιτάζτε προσεκτικά των κώδικα που ακολουθεί και βγάλτε τα συμπεράσματα από μόνοι σας.

```
$ ("table tr:nth-child (even)").addClass ("striped");
```

Μην ανησυχείτε αν σας φαίνεται λίγο “κινέζικο”, αλλά όταν το καταλάβετε θα δακρύσετε από τη χαρά σας.

Το αποτέλεσμα του πιο πάνω κώδικα: για δέστε καλά το παρακάτω σχήμα και μετά δακρύστε.

Company	Offer	Location
 Moustarda	I have a little problem though: I have set width and height for the search results container and made overflow:auto, such that if search results are longer than the set height a scroll will appear. But when I try to scroll on the results, the container with the results simply disappears.	Myconos
 Moustarda	I have a little problem though: I have set width and height for the search results container and made overflow:auto, such that if search results are longer than the set height a scroll will appear. But when I try to scroll on the results, the container with the results simply disappears.	Myconos
 Liakouras Ltd	I have a little problem though: I have set width and height for the search results container and made overflow:auto, such that if search results are longer than the set height a scroll will appear. But when I try to scroll on the results, the container with the results simply disappears.	Myconos
 Liakouras Ltd	I have a little problem though: I have set width and height for the search results container and made overflow:auto, such that if search results are longer than the set height a scroll will appear. But when I try to scroll on the results, the container with the results simply disappears.	Myconos
	I have a little problem though: I have set width and height for the search results container and made	

Σχήμα 3.1

3.3.1 Βασικές αρχές του JQuery;

Αυτό που πετυχαίνουμε κάνοντας χρήση του JQuery, είναι να παίρνουμε αντικείμενα τα οποία είναι γραμμένα σε HTML και να εκτελούμε διάφορα συμβάντα πάνω σε αυτά.

Με ποιον τρόπο όμως το πετυχαίνουμε αυτό; Για να μπορέσουμε να συλλέξουμε ένα γκρουπ αντικειμένων τα οποία ανήκουν τις περισσότερες φορές σε κάποια ομάδα χρησιμοποιούμε τον επιλογέα του JQuery ως εξής :

\$ (selector)
ή
jQuery (selector)

Αρχικά το σήμα του δολαρίου \$ () μπορεί να σας φαίνεται παράξενο, όσο όμως θα σας κερδίζει το JQuery θα εξοικειωθείτε απόλυτα με αυτό. Για παράδειγμα, για να πάρουμε όλα τα links τα οποία είναι μέσα σε μια παράγραφο αρκεί να γράψουμε το ακόλουθο κομμάτι κώδικα : \$ ("p a").

Το `$ ()`, επιστρέφει ένα ειδικό αντικείμενο της JavaScript, το οποίο αποτελείται από έναν πίνακα με αντικείμενα DOM [35] τα οποία είναι στη προκειμένη περίπτωση τα `links` μέσα σε μια παράγραφο. Η κατανόηση των παραπάνω αποτελεί τα `άλφα` και το `ωμέγα` στην περαιτέρω κατανόηση του κεφαλαίου, για τον λόγο αυτό πάρτε λίγο χρόνο και προσπαθήστε να ξαναθυμηθείτε έννοιες όπως JavaScript Arrays, και DOM elements. Για να σας βοηθήσουμε παραπάνω θα αναφέρουμε ακόμα ένα παράδειγμα, έστω ότι θέλουμε να κάνουμε μια `Div` με `id ilias` να εξαφανιστεί. Το βρήκε κανείς, αν όχι κοιτάξτε παρακάτω διαφορετικά κάντε πως η γραμμή που ακολουθεί απλά δεν υπάρχει.

```
$("#ilias").fadeOut ();
```

Εάν θέλαμε να εξαφανίσουμε μια `Div` με κλάση `ilias` της οποίας το `id` δεν ήταν γνωστό τι θα κάναμε; Ακριβώς αυτό που σκεφτήκατε.

```
$("div.ilias").fadeOut ();
```

Μεταξύ των δύο παραπάνω δεν υπάρχει εμφανής διαφορά, αφού τα δύο κομμάτια κώδικα εξυπηρετούν τον ίδιο σκοπό. Εν μέρη έτσι είναι, όμως υπάρχει μια σημαντική διαφορά την ύπαρξη της οποίας δεν πρέπει σε καμία περίπτωση να αγνοήσουμε. Αφενός γιατί αποτελεί την κύρια δύναμη του JQuery και αφετέρου γιατί θα οδηγηθούμε σε ανεξέλεγκτα προβλήματα μέσα στην εφαρμογή. Αρκετά κράτησε η αγωνία, για όσους δεν το κατάλαβαν είναι ότι στην πρώτη περίπτωση που γνωρίζουμε την ταυτότητα της `div` που θέλουμε να εξαφανίσουμε, απλά το πραγματοποιούμε γράφοντας τον κώδικα που περικλείει την ταυτότητα της. Στην επόμενη περίπτωση που γνωρίζουμε την κλάση της `div` την οποία θέλουμε να εξαφανίσουμε γράφουμε τον κώδικα που περιλαμβάνει τον επιλογέα `$("div.ilias")` και το πετυχαίνουμε, όμως μαζί με αυτή εξαφανίζουμε και όλες τις άλλες `div` που τυχόν έχουν την ίδια κλάση με την προηγούμενη. Είναι ένα παράδειγμα το οποίο έχει χρησιμοποιηθεί σε μεγάλο βαθμό στην εφαρμογή που συνοδεύει αυτό το εγχειρίδιο. Για να κατανοήσετε περαιτέρω την δύναμη του επιλογέα του JQuery, θα σας δώσουμε μερικά παραδείγματα παρακάτω :

```
$("p:even");
```

Επιλέγει όλα τα `<p>` στοιχεία τα οποία είναι `even`.

```
$("tr:nth-child (1)");
```

Επιλέγει την πρώτη σειρά κάθε πίνακα.

```
$("body > div");
```

Επιλέγει το πρώτο `<div>` μέσα στο `<body>`.

```
$("a[href$=pdf]");
```

Επιλέγει τα `links` σε αρχείο `pdf`.

```
$ ("body > div:has (a)")
```

Επιλέγει το πρώτο<div> μέσα στο <body>το οποίο έχει links.

Μια πλήρης λίστα με επιλογείς αυτής της μορφής μπορείτε να βρείτε στην διεύθυνση <http://docs.jquery.com/Selectors>.

3.3.2 To document ready

Συχνά εκτελούνται ενέργειες πάνω στα αντικείμενα μιας δικτυακής εφαρμογής από διεργασίες οι οποίες δεν αποτελούν τμήμα του κώδικα ο οποίος εμπεριέχει τα αντικείμενα αυτά ή αυτά δεν αποτελούν μέρος του κώδικα. Για τις περιπτώσεις αυτές θα πρέπει να δημιουργήσουμε έναν τρόπο, ο οποίος να περιμένει μέχρι ως ότου, όλα τα DOM αντικείμενα έχουν φορτώσει και μετά να εκτελέσει τις διεργασίες πάνω σε αυτά. Για παράδειγμα στην περίπτωση του πίνακα (σχήμα 3.1) που αναφέραμε προηγουμένως θα πρέπει, πρώτα να φορτωθεί ο πίνακας και στην συνέχεια να βάψουμε της γραμμές του. Παραδοσιακά για τον σκοπό αυτό υπάρχει η onload function, η οποία για το παράδειγμα του πίνακα θα μπορούσε να εφαρμοστεί ως εξής:

```
window.onload = function () {
  $ ("table tr:nth-child (even)").addClass ("even");};
```

Η συνάρτηση αυτή εφαρμόζει την εναλλαγή του χρώματος στις γραμμές ενός πίνακα, αφού το όλο έγγραφο έχει φορτωθεί. Δυστυχώς ο φυλλομετρητής (Browser) δεν περιμένει, για να εκτελέσει την λειτουργία των γραμμών, μόνο το DOM δέντρο να ολοκληρωθεί, αλλά περιμένει να φορτώσει όλο το έγγραφο μαζί και οι εικόνες και το έγγραφο να εμφανιστεί στο παράθυρο του Browser. Τραγικό θα τολμούσαμε να πούμε από άποψη χρόνου. Το αποτέλεσμα είναι ο χρήστης να παρατηρεί μια καθυστέρηση από την στιγμή που βλέπει το έγγραφο και την στιγμή που εμφανίζεται κάποιο script. Ακόμα χειρότερα, φανταστείτε μία εικόνα να καθυστερεί για κάποιον λόγο να φορτώσει, ο χρήστης θα πρέπει να περιμένει μέχρι η εικόνα να εμφανιστεί, για να μπορέσει οποιαδήποτε δυναμική εφαρμογή της σελίδας.

Μια σημαντικά καλύτερη λύση θα ήταν ο Browser, να περιμένει μέχρι να ολοκληρωθεί η μετατροπή της HTML που σχετίζεται με το δυναμικό κομμάτι του κώδικα, στην προκειμένη περίπτωση ο πίνακας, στο DOM δέντρο του. Αφού γίνει αυτό στην συνέχεια να εκτελεί το script το οποίο θα εμπλουτίζει την μορφή του, στην προκειμένη περίπτωση το χρώμα των γραμμών. Μπερδευτήκατε, σκεφτείτε ένα παράδειγμα της καθημερινότητας, έστω ότι είμαστε σε ένα εργοστάσιο παραγωγής αυτοκινήτων. Με την onload, για να βάψουμε το σασί

θα έπρεπε να περιμένουμε μέχρι να ολοκληρωθεί η συναρμολόγηση του αυτοκινήτου ακόμα και με την πιο απίθανη λεπτομέρεια. Εάν όμως υπήρχε μια προϋπόθεση ότι για μπουν τα φανάρια το σασί θα έπρεπε να είναι βαμμένο, κάνοντας χρήση της παραπάνω συνάρτησης η διαδικασία δεν θα ολοκλήρωνε ποτέ. Ακόμα και να μην υπήρχε καμία προϋπόθεση ο χρόνος που θα σπαταλούσαν περιμένοντας για παράδειγμα μία βαλβίδα η οποία χάθηκε, θα ήταν καταστροφικός.

Την λύση έρχεται να δώσει η Ready, η οποία για το παράδειγμα του πίνακα θα γραφεί ως εξής :

```
$(document).ready(function () {  
    $("table tr:nth-child (even)").addClass ("even");  
});
```

Αυτό που κάνει είναι να περιτυλίγει, ένα στιγμιότυπο του εγγράφου με την JQuery (), θυμηθείτε ότι είναι το ίδιο με \$ (). Στην συνέχεια να εφαρμόζει στο στιγμιότυπο αυτό μια συνάρτηση ready, μέσα στο σώμα της οποίας περνάει μια συνάρτηση η οποία θα εκτελεστεί όταν αυτό είναι έτοιμο. Η συνάρτηση που εφαρμόζεται μέσα στην ready, δίνει πληροφορίες στον Browser να περιμένει μόνο μέχρι το DOM είναι έτοιμο, τονίζουμε στο σημείο αυτό μόνο το DOM. Εάν εφαρμόσουμε την μέθοδο αυτήν περνώντας στην ready περισσότερες από μία συναρτήσεις, ο Browser θα τις εκτελέσει όλες με την σειρά όπως αυτές εμφανίζονται μέσα στην ιστοσελίδα.

```
$(document).ready(function () {  
  
    // add * to required field labels  
    $('label.required').append ('&nbsp;<strong>*</strong>&nbsp;');  
  
    $("a").click(function () {  
        $(this).blur ();  
    });  
  
    //hide toolbar and make visible the 'show' button  
    $("span.downarr a").click(function () {  
        $("#toolbar").slideToggle ("fast");  
        $("#toolbarbut,#navigation li.contact").fadeIn ("slow");  
    });  
  
    //show toolbar and hide the 'show' button  
    $("span.showbar a, li.contact ").click(function () {
```

```

$("#toolbar").slideToggle ("fast");
$("#toolbarbut,#navigation li.contact").fadeOut ();
});
});

```

Είναι φανερό το πλεονέκτημα που μας προσφέρει η μεθοδολογία αυτήν, όσον αφορά στον χρόνο εκτέλεσης, η καλύτερα ετοιμασίας προς χρήση μιας ιστοσελίδας.

3.3.3 Κατασκευή DOM (Document Object Models)

Είναι κατά την γνώμη μας κάτι που οι περισσότεροι από εσάς θα λατρεύετε. Μπορούμε δυναμικά να δημιουργήσουμε στοιχεία DOM, περνώντας μέσα στον επιλογέα \$ (), απλά κώδικα HTML. Για να γίνει πιο αντιληπτό θα μπορούσαμε πολύ εύκολα να κατασκευάσουμε μία παράγραφο με τον ακόλουθο πολύ απλό τρόπο:

```
$("<p>Hi there!</p>")
```

Κάποιος θα τολμούσε να πει σιγά το πράγμα, τι άλλο μπορεί να πετύχουμε, αφού κάτι παρόμοιο το κάνουν και άλλες script Γλώσσες; Θα συμφωνούσαμε και εμείς, όμως για να σας πείσουμε για τις δυνατότητες αυτού του εργαλείου θα σας περιγράψουμε ένα σχετικά απλό κομμάτι κώδικα.

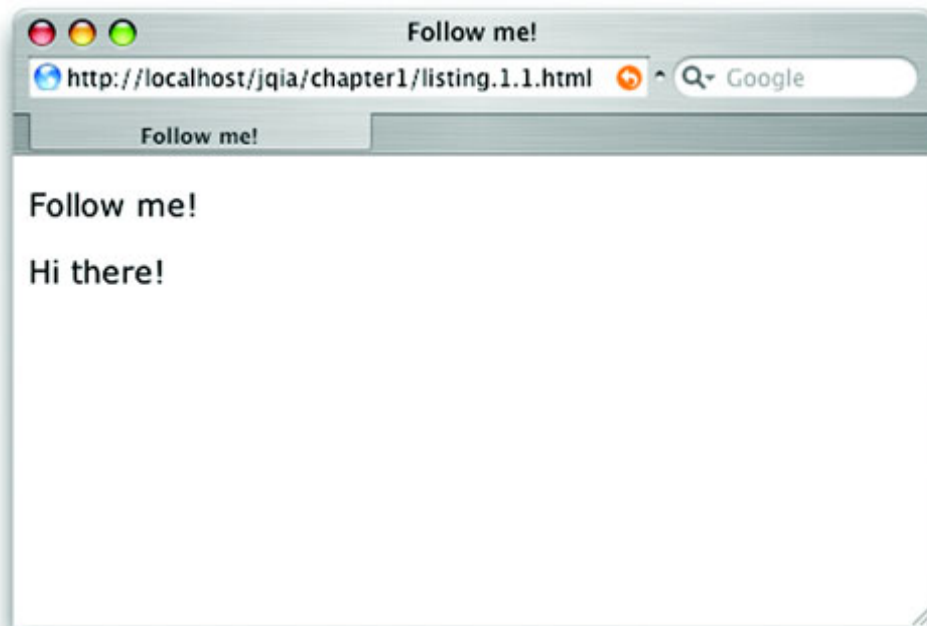
```

1 <html>
2 <head>
3 <title>Follow me!</title>
4 <script type="text/javascript" src="../scripts/jquery-1.2.js">
5   </script>
6   <script type="text/javascript">
7     $(document).ready (function(){
8       $("#p>Hi there!</p>").insertAfter("#followMe");
9     });
10  </script>
11 </head>
12 <body>
13 <p id="followMe">Follow me!</p>
14 </body>
15 </html>

```

Στο παράδειγμα αυτό το οποίο πρέπει να είναι κατανοητό για τους περισσότερους από εσάς, οι υπόλοιποι θα πρέπει να φρεσκάρете λίγο την JavaScript και την HTML, έχουμε μία παράγραφο με ταυτότητα followMe, μέσα στο σώμα του εγγράφου. Στο Head του εγγράφου κατασκευάζουμε μια ready συνάρτηση, η οποία περικλείει το απλούστερο κομμάτι κώδικα «\$("<p>Hi there!</p>").insertAfter("#followMe");» σύμφωνα με το οποίο ο Browser αφού περιμένει να ολοκληρωθεί η δημιουργία της παραγράφου followMe,

κατασκευάζει μια νέα παράγραφο την οποία την προσκολλά, ακριβώς μετά την followMe. Το αποτέλεσμα αυτής διαδικασίας φαίνεται στο παρακάτω σχήμα.



Σχήμα 3.2

3.4 Περιτύλιγμα (Wrapping)

Είδαμε λοιπόν, μέχρι αυτό το σημείο πώς με μικρό κώδικα μπορούμε να επιλέξουμε ένα αντικείμενο του κώδικα μας και να το τροποποιήσουμε δυναμικά. Η λέξη επιλέγω θα πρέπει να σας γίνει συνώνυμο με το JQuery, γιατί στην ουσία κάνοντας χρήση οποιασδήποτε μεθόδου του JQuery, αυτομάτως κάνουμε και επιλογή κάποιου αντικειμένου πάνω στο οποίο, η συγκεκριμένη μέθοδος θα ενεργήσει. Πολλές φορές τα περισσότερα από αντικείμενα που θα θέλουμε να επιλέξουμε θα είναι πολύ εύκολο να τα περιγράψουμε, όπως για παράδειγμα όλα τα Link μιας ιστοσελίδας. Άλλες φορές η περιγραφή αυτή θα έχει μια σχετική δυσκολία, για παράδειγμα όλα τα στοιχεία λίστας τα οποία έχουν κλάση «iIias» και περιέχουν Links.

Ευτυχώς για εμάς, το JQuery μας παρέχει μια σχετικά απλή μέθοδος προσδιορισμού με ακρίβεια οτιδήποτε μέσα σε μια σελίδα θέλουμε να επιλέξουμε.

3.4.1 Βασικοί CSS επιλογείς

Για να μπορούν να εφαρμόζουν στυλ στις ιστοσελίδες τους, πολύ web developers κάνουν χρήση ενός μικρού όμως πολύ δυναμικού γκρουπ από μεθόδους επιλογής, οι οποίες

δουλεύουν σε όλους σχεδόν του Browsers. Οι μέθοδοι, χρησιμοποιούν τα ID, CSS κλάσεις, τα tags και τις ιεραρχίες των DOM elements. Ένα μικρό παράδειγμα αυτών είναι :

- a—επιλέγει όλα τα (<a>).
- #specialID—επιλέγει όλα τα στοιχεία που έχουν id specialID.
- .specialClass—επιλέγει τα στοιχεία με class specialClass.
- a#specialID.specialClass—επιλέγει όλα τα link με id of specialID και class specialClass.
- p a.specialClass—επιλέγει όλα τα links με class specialClass τα οποία βρίσκονται μέσα σε <p>.

Όλους τους τύπους των επιλογών, μπορούμε να τους συνδυάσουμε και να πάρουμε πληθώρα από ομάδες στοιχείων μέσα σε ένα έγγραφο. Για να συνδυάσουμε τους επιλογείς που ξέρουμε με το JQuery, αρκεί να τους περικλείσουμε μέσα στον επιλογέα του. Για παράδειγμα \$ (“p a.ialias”). Θα μπορούσαμε να μιλάμε με παραδείγματα σχετικά με τους επιλογείς για αρκετά κεφάλαια, δεν είναι όμως ο σκοπός αυτού του εγχειριδίου. Παρακάτω σας δίνεται πίνακας με τους πιο βασικούς επιλογείς της CSS που υποστηρίζονται από το JQuery.

Selector	Description
*	Matches any element.
E	Matches all element with tag name E.
E F	Matches all elements with tag name F that are descendents of E
E>F	Matches all elements with tag name F that are direct children of E.
E+F	Matches all elements F immediately preceded by sibling E.
E~F	Matches all elements F preceded by any sibling E.
E:has (F)	Matches all elements with tag name E that have at least one descendent with tag name F.
E.C	Matches all elements E with class name C. Omitting E is the same as *.C.
E#I	Matches element E with id of I. Omitting E is the same as *#I.
E[A]	Matches all elements E with attribute A of any value.
E[A=V]	Matches all elements E with attribute A whose value is exactly V.
E[A^=V]	Matches all elements E with attribute A whose value begins with V.
E[A\$=V]	Matches all elements E with attribute A whose value ends with V.
E[A*=V]	E[A*=V] Matches all elements E with attribute A whose value contains V.

Σχήμα 3.3

3.4.2 Επιλογείς με βάση την θέση

Μία εξίσου δυναμική μέθοδος επιλογής είναι με βάση την θέση του στοιχείου ως προς κάποιο άλλο μέσα στην σελίδα μας. Μπορούμε για παράδειγμα να επιλέξουμε το πρώτο Link μέσα στην σελίδα ή του τελευταίο στοιχείο μιας λίστας. Θα έχουμε λοιπόν :

a:first
li:last-child.

Ένας συγκεντρωτικός πίνακας με όλους τους επιλογείς θέσης οι οποίοι υποστηρίζονται από το JQuery φαίνονται στο σχήμα 2.4.

Selector	Description
:first	The first match of the page. li a:first returns the first link also under a list item.
:last	The last match of the page. li a:last returns the last link also under a list item.
:first-child	The first child element. li:first-child returns the first item of each list.
:last-child	The last child element. li:last-child returns the last item of each list.
:only-child	Returns all elements that have no siblings.
:nth-child (<i>n</i>)	The <i>n</i> th child element. li:nth-child (2) returns the second list item of each list.
:nth child (<i>even odd</i>)	Even or odd children. li:nth-child (<i>even</i>) returns the even children of each list.
:nth-child ($Xn+Y$)	The <i>n</i> th child element computed by the supplied formula. If <i>Y</i> is 0, it may be omitted. li:nth-child (3 <i>n</i>) returns every third item, whereas li:nth-child (5 <i>n</i> +1) returns the item after every fifth element.
:even and :odd	Even and odd matching elements page-wide. li:even returns every even list item.
:eq (<i>n</i>)	The <i>n</i> th matching element.
:gt (<i>n</i>)	Matching elements after (and excluding) the <i>n</i> th matching element.
:lt (<i>n</i>)	Matching elements before (and excluding) the <i>n</i> th matching element.

Σχήμα 3.4

3.4.3 Προσαρμοσμένοι επιλογείς JQuery

Όλοι οι παραπάνω επιλογείς μας δίνουν την δυνατότητα να επιλέγουμε στοιχεία μέσα σε μια σελίδα με βάση ορισμένα από τα χαρακτηριστικά τους, Τι θα γίνει όμως εάν θελήσουμε να επιλέξουμε ένα checkbox το οποίο είναι τσεκαρισμένο; Και σε αυτήν την περίπτωση το JQuery υποστηρίζει ένα μεγάλο αριθμό επιλογέων οι οποίοι ονομάζονται Custom σχήμα 3.5.

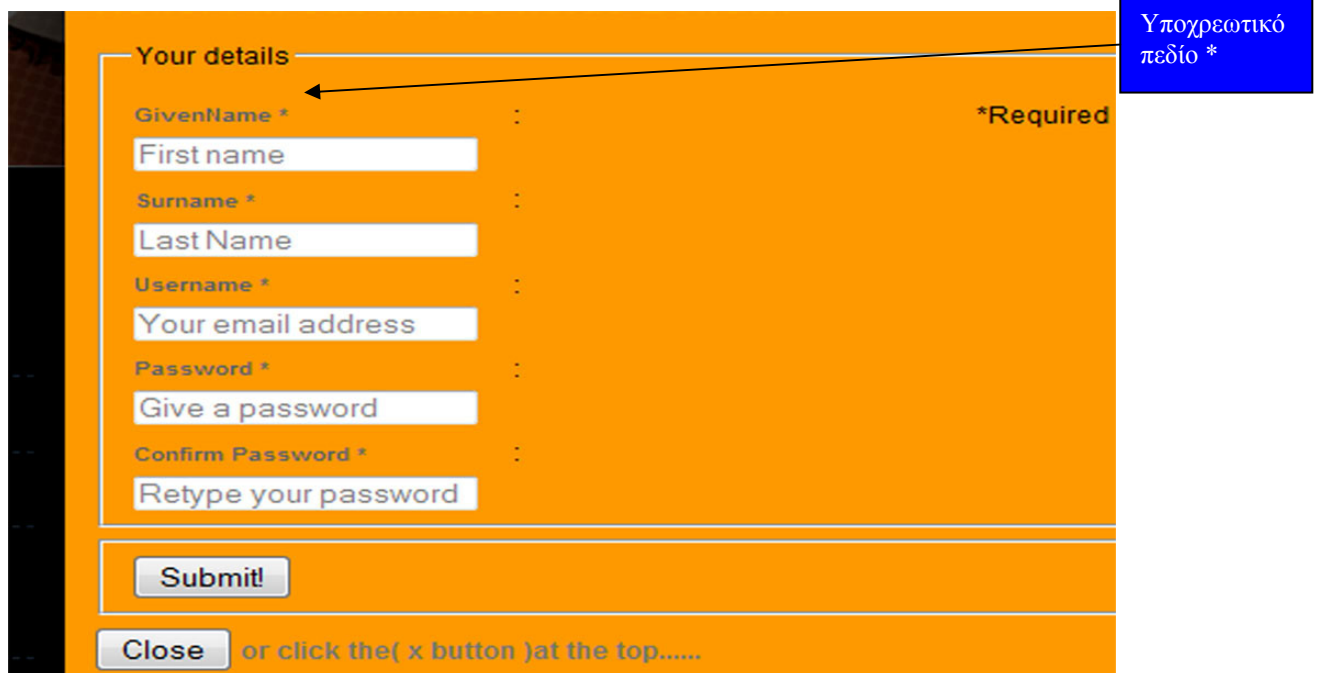
Selector	Description
:animated	Selects elements that are currently under animated control. Chapter 5 will cover animations and effects.
:button	Selects any button (<code>input[type=submit]</code> , <code>input[type=reset]</code> , <code>input[type=button]</code> , or <code>button</code>).
:checkbox	Selects only check box elements (<code>input[type=checkbox]</code>).
:checked	Selects only check boxes or radio buttons that are checked (supported by CSS).
:contains (foo)	Selects only elements containing the text <i>foo</i> .
:disabled	Selects only form elements that are disabled in the interface (supported by CSS).
:enabled	Selects only form elements that are enabled in the interface (supported by CSS).
:file	Selects all file elements (<code>input[type=file]</code>).
:header	Selects only elements that are headers; for example: <code><h1></code> through <code><h6></code> elements.
:hidden	Selects only elements that are hidden.
:image	Selects form images (<code>input[type=image]</code>).
:input	Selects only form elements (<code>input</code> , <code>select</code> , <code>textarea</code> , <code>button</code>).
:not (filter)	Negates the specified filter.
:parent	Selects only elements that have children (including text), but not empty elements.
:password	Selects only password elements (<code>input[type=password]</code>).
:radio	Selects only radio elements (<code>input[type=radio]</code>).
:reset	Selects reset buttons (<code>input[type=reset]</code> or <code>button[type=reset]</code>).
:selected	Selects option elements that are selected.
:submit	Selects submit buttons (<code>button[type=submit]</code> or <code>input[type=submit]</code>).
:text	Selects only text elements (<code>input[type=text]</code>).
:visible	Selects only elements that are visible.

Σχήμα 3.5

Πολλοί από εσάς θα αναρωτιέστε, ποιος ο λόγος που αποφασίσαμε να συμπεριλάβουμε τις ενότητες με τους επιλογείς, καθώς επίσης και τους πίνακες που σχετίζονται με αυτούς; Ο λόγος είναι ότι πιστεύουμε εάν μάθουμε πώς να επιλέγουμε σύνολα αντικειμένων μέσα από μια ιστόθεση, θα είμαστε σε θέση να καταλαβαίνουμε κομμάτια κώδικα καθώς επίσης και να τα δημιουργούμε. Για το λόγο αυτό έχοντας στα χέρια σας όλα τα προηγούμενα καθώς και τους πίνακες, θα περάσουμε σε πιο βαθιά νερά και θα μάθουμε πως στις σελίδες που θα αναπτύξουμε μπορούμε να δώσουμε ζωή. Θα μπορούσαμε θεωρητικά να σας αναφέρουμε κάποια πράγματα, όμως κρίναμε ότι μέσα από παραδείγματα, εξασκούμε τον τρόπο σκέψης μας και μπορούμε να αντιληφθούμε κάποια πράγματα καλύτερα.

3.5 Δίνοντας ζωή στις Σελίδες μας

Ένα από τα πιο βασικά που μπορούμε να τροποποιήσουμε, όταν αναφερόμαστε σε DOM αντικείμενα, είναι οι ιδιότητες τους και τα χαρακτηριστικά τους. Όλα αυτά αναθέτονται στα αντικείμενα αυτά κατά την δημιουργία τους με την HTML, και μπορούν δυναμικά να τροποποιηθούν με την χρήση του JQuery. Πώς για παράδειγμα θα μπορούσαμε να τοποθετήσουμε το αστεράκι (*) σε μια σειρά από label, που αφορούν μία φόρμα προτρέποντας τον χρήστη να συμπληρώσει υποχρεωτικά όλα τα πεδία της φόρμας σχ.3.6;



The image shows a registration form with the following fields and labels:

- GivenName * (with a blue callout box pointing to the asterisk and the text "Υποχρεωτικό πεδίο *")
- Surname *
- Username *
- Password *
- Confirm Password *

Each label is followed by a colon and the text "*Required". The form also includes a "Submit" button and a "Close" button with the text "or click the(x button)at the top.....".

Σχήμα 3.6

Θα μπορούσε κάποιος να πει ότι θα γράφαμε τον κώδικα σε HTML, και δίπλα στο όνομα του Label, σε κάθε ένα από αυτά θα βάζαμε και ένα αστεράκι (*). Εγώ θα απαντούσα ότι

αυτό δεν θα ήταν λάθος. Γνωρίζοντας όμως τις δυνατότητες του JQuery καθώς και την δυνατότητα όλων των επιλογέων, θα τολμούσαμε να σας πούμε ότι αυτό γίνεται με μισή γραμμή κώδικα. Και εξηγούμε για να μην παρεξηγηθούμε:

```
<div class="form-row">
<div class="field-label"><label for="surname" class="input
required">Surname</label>:</div>
<div class="field-wiget"><input id="surname" name="surname" type="text"
placeholder="Last Name" required><span class="liakos"></span></div></div>
```

Όπως φαίνεται και στο παραπάνω κομμάτι κώδικα, το οποίο δεν είναι και κάτι τραγικό, εάν τρομάξατε καλά είναι να ανοίξετε κανένα βιβλίο της HTML, ορίζουμε το Label ενός πεδίο με όνομα surname το οποίο και φέρει το όνομα επίσης του πεδίου. Να τονίσουμε η συμπλήρωση του πεδίου αυτή είναι υποχρεωτική, άρα το αστεράκι θα πρέπει να προστεθεί. Πώς θα το πετύχουμε αυτό; Για κοιτάξετε παρακάτω.

```
$ ('label.required').append ('&nbsp;<strong>*</strong>&nbsp;');
```

Ελπίζουμε από το σήμα του δολαρίου και μόνο να καταλάβετε ότι έχει να κάνει η παραπάνω γραμμή με JQuery. Τι κάναμε λοιπόν; Μέσα στο Label που δημιουργήσαμε στα χαρακτηριστικά του, και ειδικότερα στις κλάσεις του προσθέσαμε την κλάση required. Το ίδιο κάναμε σε όλα τα υπόλοιπα Labels τα οποία είναι υποχρεωτικά. Οι ιδιότητες που πρόσθεσε η κλάση αυτήν στο συγκεκριμένο πεδίο δεν αποτελεί μέρος αυτού του συγγράμματος και αφορά σε CSS, το οποίο θεωρούμε ότι είναι ήδη γνωστό. Ο λόγος που προσθέσαμε την κλάση αυτή είναι, ότι με τον κατάλληλο επιλογέα θα μπορούσαμε να επιλέξουμε όλα τα Label τα οποία έχουν την κλάση αυτή και να τα τροποποιήσουμε, για παράδειγμα σε όλα τα να προσθέσουμε το αστεράκι.

Έτσι και κάνουμε γράφουμε, \$ ('label.required'), το οποίο διαλέγει όλα τα labels τα οποία έχουν την κλάση required μέχρι εδώ δεν θα πρέπει να υπάρχει κανένα πρόβλημα. Το αποτέλεσμα αυτού του επιλογέα θα είναι ένας πίνακας ο οποίος έχει ως στοιχεία του όλα τα labels με την κλάση αυτή. Στην συνέχεια θέλουμε να κολλήσουμε σε κάθε label το αστεράκι, append (' * '), εδώ σίγουρα θα πρέπει να πάθατε πανικό. Το append λέει στον Browser, σε κάθε ένα από αυτά τα αντικείμενα να προσθέσει κάτι, αυτό το κάτι συνήθως γράφεται μέσα σε μονά αυτάκια (‘’) και είναι κώδικας HTML. Στην προκειμένη, (*), δηλαδή ένα αστεράκι. κώδικας αυτός θα μπορούσε να συμπεριλαμβάνει περισσότερες εντολές της HTML, οι οποίες θα έπρεπε να χωρίζουν μεταξύ τους με ελληνικό ερωτηματικό. Από αυτό και μόνο το παράδειγμα γίνονται κατανοητές οι δυνατότητες στην επιλογή και τροποποίηση των αντικειμένων, με την βοήθεια του JQuery.

Ένα πιο περίπλοκο παράδειγμα ακολουθεί παρακάτω, προσπαθήστε πριν διαβάσετε την περιγραφή του κώδικα που ακολουθεί να καταλάβετε τι κάνει σε κάθε γραμμή.

```
1. $.Ajax ({
2.   type: "POST",
3.   url: "do-contact.php",
4.   data: dataString,
5.   success: function (){//sos me to. html antika8istw ta perioxomena toy contact me kainouria
6.     $('#contact').html ("<div id='message1'></div>");
7.     $('#message1').html ("<h2>Contact Form Submitted!</h2>")
8.     .append ("<p>We will be in touch soon.</p>")
9.     .hide ()
10.    .fadeIn (1500, function () {
11.      $('#message1').append ("<img id='checkmark' src='images/check.png' />");
12.    });
13.    setTimeout (function (){
14.      var login_response2 ='<form id="myform_cont" name="myform_cont" method="post" action="">'+
15.      '<ul class="form">'+
16.      '<li >'+
17.      '<label for="form_name"class="input required">Name</label>'+
18.      '<input id="form_name" name="form_name" type="text" required />'+
19.      '</li>'+
20.      '<li class="email">'+
21.      '<label for="form_email" class="input required">Email</label>'+
22.      '<input id="form_email" name="form_email" type="text" required />'+
23.      '<div id="email_error"><span ></span></div>'+
24.      '</li>'+
25.      '<li class="comments">'+
26.      '<label for="form_comment" class="input required">Message</label>'+
27.      '<textarea id="form_comment" name="form_comment" rows="" cols="" required></textarea></br>'+
28.      '<span id="message_error"></span>'+
29.      '</li>'+
30.      '<li class="submit1">'+
31.      '<input id="submit7" type="submit" value="send" class=" button small red awesome glossy icon" />'+
32.      '<div id="contact_response">'+
33.      '&nbsp;Processing...</div></li></ul></form>';
34.    }
35.  }
36.  });
```

35. });//end Ajax

Όσοι από εσάς καταλάβατε, είστε σε καλό δρόμο οι υπόλοιποι, διαβάστε την περιγραφή που ακολουθεί προσεκτικά, μην διστάσετε να την ξαναδιαβάσετε έως ότου μπορέσετε να καταλάβετε αυτά που περιγράφονται.

Ο κώδικας αρχίζει με κάτι που δεν αποτελεί στοιχείο μελέτης αυτού του κεφαλαίου και δεν είναι τίποτε άλλο από τον Ajax, το εργαλείο αυτό θα περιγραφεί σε επόμενο κεφάλαιο γιατί θεωρούμε ότι είναι ένα από τα σημαντικότερα εργαλεία που δίνουν ζωή σε μία διαδικτυακή εφαρμογή. Εμείς θα ασχοληθούμε με αυτό που απεικονίζεται από την γραμμή 5 έως και την γραμμή 34. Στο σημείο αυτό θα σας υπενθυμίσουμε ότι το JQuery, αποτελεί μια προέκταση του JavaScript, γεγονός που σημαίνει όλα αυτά που ξέρετε για αυτό να ισχύει και για το JQuery. Στο παραπάνω παράδειγμα δεν είναι εμφανές που ακριβώς χρησιμεύει ο παραπάνω κώδικας, ξεκινάει με μια συνάρτηση η οποία δεν παίρνει κανένα όρισμα και ορίζεται ως `function () {}`. Μέσα στο σώμα της συνάρτησης είναι αυτό που θα μας απασχολήσει, γιατί πρόκειται για την δυνατότητα που μας δίνει το JQuery να δημιουργούμε κομμάτια κώδικα τα οποία δεν προϋπήρχαν.

```
6. $('#contact').html("<div id='message1'></div>");
```

Στην σειρά αυτή έχουμε επιλογή του αντικειμένου `contact`, το οποίο γίνεται εύκολα αντιληπτό, στην συνέχεια όμως έχουμε την «.html». Με την μέθοδο αυτή λέμε στον Browser να αντικαταστήσει ότι έχει μέσα στην `contact`, με αυτό που του λέει η μέθοδος `html`, στην προκειμένη περίπτωση «<div id='message1'></div>». Αφού δημιουργηθεί το αντικείμενο `contact` και τοποθετηθεί σε αυτό η `message1`, δίνουμε στον Browser την εντολή να προσθέσει πλέον, το τονίζουμε να προσθέσει ένα νέο αντικείμενο το (`<p>We will be in touch soon.</p>`). Για να το επιτύχουμε αυτό κάνουμε χρήση της μεθόδου `append()`, η μέθοδος αυτή τοποθετεί, χωρίς να αντικαθιστά, οτιδήποτε βρίσκετε μέσα στο σώμα της `append("<p>We will be in touch soon.</p>")`. Μετά από αυτήν την εντολή θα έχουμε δημιουργήσει ένα αντικείμενο μέσα στο οποίο θα υπάρχει μία πρόταση η «Contact Form Submitted!» και μία παράγραφος που θα λέει «We will be in touch soon».

Από τις δύο και μόνο γραμμές γίνεται αντιληπτό ότι με πολύ λίγο προγραμματισμό γεννάμε κώδικα που δεν υπήρχε πριν. Δίνουμε λοιπόν ζωή σε μια στατική σελίδα. Σύμφωνα με την ροή του κώδικα έχουμε:

```
.hide ()
.fadeIn (1500, function () {
$('#message1').append ("<img id='checkmark' src='images/check.png' />");
});
```

Αυτό που γίνεται στις σειρές αυτές είναι, εφαρμόζουμε στην `contact` την `hide()`, αποτέλεσμα αυτού είναι ότι βρίσκεται μέσα στο σώμα της να γίνει μη ορατό. Αμέσως μετά κάνουμε εφαρμογή της `fadeIn`, η οποία είναι μια μέθοδος για να πετύχουμε το εφέ επανεμφάνισης με αργό τρόπο, της εξαφανισμένης πλέον `contact`. Η `fadeIn()`, μέσα στο σώμα της μπορεί να έχει έναν αριθμό που μετράει σε ms, τον χρόνο που πρέπει να περιμένει ο Browser έως ότου να εκτελέσει αυτό που ακολουθεί, και δεν είναι άλλο από μια συνάρτηση η οποία στο `message1` επισυνάπτει μία εικόνα.

Σίγουρα έχετε μπερδευτεί, δεν πειράζει όμως αυτός ήταν ο σκοπός μας. Κάνοντας μια μικρή επανάληψη είναι σίγουρο ότι θα ξεκαθαρίσουν πολλά πράγματα στο μυαλό σας. Θα σας αποκαλύψουμε που χρησιμεύει ένα τέτοιο κομμάτι κώδικα, είναι σίγουρο ότι οι περισσότεροι από εσάς το έχετε συναντήσει άπειρες φορές. Είναι σύνηθες όταν θέλουμε να επικοινωνήσουμε με μια εταιρεία μέσω της ιστοσελίδας της, μας παρέχετε ένας σύνδεσμος, ο οποίος μας οδηγεί σε μια φόρμα που έχει την μορφή του σχήματος σχ.3.7.



Σχήμα 3.7

Αφού λοιπόν γράψουμε ότι έχουμε να πούμε, και αφού γίνει έλεγχος ότι όλα αυτά που γράψαμε είναι τα επιθυμητά, επιλέγουμε το πλήκτρο για την αποστολή του μηνύματος. Ο Browser, αποστέλλει το μήνυμα σε Βάση δεδομένων και στην συνέχεια μας εμφανίζει κάποιο μήνυμα που μας ενημερώνει για την επιτυχής ή όχι αποστολή σχ.2.8. Ο παραπάνω κώδικας κάνει αυτό ακριβώς το πράγμα, η διαφορά όμως είναι ότι το κάνει χωρίς να

ανανεώνει την σελίδα, με αποτέλεσμα το φόρτωμα όλων των στοιχείων που θα οδηγούσε σε σπατάλη πολύτιμου χρόνου.



Σχήμα 3.8

Είναι πολύ σοβαρό να αποφεύγετε η ανανέωση των σελίδων όπου αυτό είναι εφικτό, γιατί οδηγεί σε άσκοπη καθυστέρηση. Στο σχήμα 3.7 βλέπουμε ακριβώς αυτό των κώδικα που ερμηνεύσαμε πιο πάνω. Αφού μας βγάλει το μήνυμα για την επιτυχή ή μη αποστολή, πρέπει να μας δίνεται η δυνατότητα να στείλουμε και άλλο μήνυμα εάν το επιθυμούμε. Ο κώδικας που ακολουθεί κάνει ακριβώς αυτό το πράγμα. Αρχίζει με μια συνάρτηση την `setTimeout`, μια εξαιρετικά χρήσιμη συνάρτηση με την οποία μπορούμε να εμφανίσουμε μια λειτουργία με χρονοκαθυστέρηση. Το μήνυμα δηλαδή «contact form submitted we will be in touch soon», επιθυμούμε να μείνει ένα εύλογο χρονικό διάστημα προτού εξαφανιστεί και την θέση του πάρει η φόρμα επικοινωνίας. Η `setTimeout`, αποτελείται από μια συνάρτηση, μέσα στο σώμα της οποίας εκτελείτε η εμφάνιση της φόρμας επικοινωνίας. Δεν θα αναφερθούμε σε λεπτομέρειες οι οποίες δεν αποτελούν σκοπό του συγγράμματος αυτού, εάν και πιστεύουμε ότι με την προηγούμενη ανάλυση του κώδικα θα είστε πλέον σε θέση να το καταλάβετε από μόνοι σας. Ένα ενδιαφέρον τμήμα είναι αυτό από την γραμμή 14 έως και την 32, στις οποίες ορίζουμε μια μεταβλητή στην οποία δίνουμε αρχική τιμή όλο τον κώδικα της φόρμας, με την μορφή κειμένου. Η μεταβλητή αυτή συμπεριφέρεται ως φόρμα πλέον στην οποία μπορούμε να εκτελέσουμε διάφορες ενέργειες. Είμαι σίγουρος με το παράδειγμα αυτό και οι δύσπιστοι, πλέον αρχίσατε να το ξανασκεφτόσαστε όσον αφορά την JQuery. Παραδείγματα σαν και αυτό μπορείτε να βρείτε πάρα πολλά, η μελέτη των οποίων θα σας βοηθήσει στην καλύτερη κατανόηση του εργαλείου αυτού.

3.6 JQuery vs JavaScript

Θεωρήσαμε σημαντικό να αφιερώσουμε ένα μέρος του κεφαλαίου αυτού, προσπαθώντας να συγκρίνουμε το JQuery με την JavaScript. Δεν είναι λίγοι αυτοί που διαβάζοντας όλα τα παραπάνω θα είπαν, ποιος ο λόγος να ασχοληθώ με κάτι τόσο νέο και να μην παραμείνω σε αυτό που τόσο καλά ξέρω; Θα τους απαντούσαμε ότι και το JQuery δεν είναι κάτι το οποίο δημιουργήθηκε από το πουθενά, αποτελεί ένα Framework του JavaScript.

Όπως όλοι γνωρίζουμε τα JavaScript frameworks, μας έχουν δώσει την δυνατότητα στο να δημιουργούμε με πολύ μεγάλη ευκολία, εφαρμογές πλούσιες σε κίνηση καθώς επίσης και interfaces τα οποία παλαιότερα θα ήθελαν πολύ περισσότερο χρόνο αν γράφαμε τον κώδικα από την αρχή. Το ερώτημα που καλούμαστε ωστόσο να απαντήσουμε είναι στο κατά πόσο όλοι αυτοί που χρησιμοποιούν βιβλιοθήκες, όπως το JQuery, θα έπρεπε να είναι σε θέση να μάθουν πώς λειτουργεί αυτό που βρίσκεται πίσω από την βιβλιοθήκη αυτή και δεν είναι άλλο από το JavaScript. Προσωπικά θα απαντούσαμε ότι επιβάλλεται στο να καταλαβαίνει κάποιος γιατί, για παράδειγμα είναι καλύτερο, να χρησιμοποιούμε τον ένα σε σχέση με τον άλλο επιλογέα και με ποιόν τρόπο θα φτιάξουμε κάποια λειτουργία που τυχόν το JQuery δεν την εκτελεί σωστά. Ο μόνος τρόπος για να πετύχουμε τα παραπάνω είναι να γνωρίζουμε JavaScript.

Ας αρχίσουμε λοιπόν την αναμέτρηση JQuery vs JavaScript. Όταν πρόκειται για μονομαχία τέτοιου είδους επικρατεί πάντα μια αγωνία για το ποιος είναι ο νικητής, το ερώτημα είναι έχουμε νικητή; Για να δούμε.

Όταν προφέρουμε την λέξη JQuery θα πρέπει να το έχουμε συνώνυμο με το Web 3.0. Δεν φοβόμαστε καμία περίπτωση να το ταυτίσουμε, και στην συνέχεια θα αναφερθούμε εκτεταμένα στο JQuery και ο λόγος είναι γιατί το προτιμούμε σε σχέση με άλλα frameworks και το γνωρίζουμε καλύτερα. Το JQuery είναι αλήθεια ότι βοήθησε έναν μεγάλο αριθμό προγραμματιστών να πλησιάσουν όσο το δυνατό περισσότερο αυτό που λέμε web scripting. Και δεν είναι τυχαίο ότι οι browser την εποχή μας, αξιολογούνται με βάση του πόσο γρήγορα και πόσο αποτελεσματικά μπορούν να υποστηρίξουν JavaScript και JQuery το ίδιο. Οι περισσότεροι από εσάς που διαβάζεται αυτό το κείμενο σίγουρα έχετε γνώσεις σχετικά με το JQuery, αν όχι εκτεταμένες σίγουρα όμως εκτενείς, για τον λόγο αυτό θα θέλαμε να αναφερθούμε σε αυτό που είναι πίσω του και δεν είναι από το ίδιο το JavaScript.

Κάθε ένας από εμάς αντιλαμβάνεται το JavaScript εντελώς διαφορετικά, και αυτό οφείλετε στον ποικίλο τρόπο χρησιμοποίησής του. Ένα είναι όμως γενικά παραδεκτό ότι αποτελεί ένα δυναμικό εύκολο στην χρήση του και στην μάθηση του εργαλείο, το οποίο είναι τόσο διαδεδομένο όσο κανένα άλλο ανάλογο με αυτό. Η περιβόητη C αποτελεί άπιαστο όνειρο για κάθε σύγχρονο προγραμματιστή του διαδικτύου λόγω των δυνατοτήτων που αυτή η γλώσσα προσφέρει. Τις πιο πολλές φορές είμαστε όλοι μας μάρτυρες της χρήσης ενός συντακτικού σχεδόν πανομοιότυπου με την C, χωρίς να το συνειδητοποιούμε. Το jQuery

πραιτέρω θολώνει τα πράγματα με τη δική του, καινοτόμο σύνταξη, η οποία αποτελεί και την μεγαλύτερη δύναμη του, που τείνει να είναι επίσης και η μεγαλύτερη αδυναμία του. Πολλοί προγραμματιστές που το χρησιμοποιούν έκαναν την ζωή τους ευκολότερη, γιατί πλέον είναι σε θέση να παράγουν JavaScript εφέ που διαφορετικά θα ήταν αδύνατο. Για παράδειγμα εάν ο πελάτης επιθυμεί ένα slide show, μέσα στη ιστοσελίδα του, αυτό μπορεί να επιτευχθεί με μερικές γραμμές κώδικα, που τις περισσότερες φορές είναι δυνατό να προέρχεται από αντιγραφή και επικόλληση. Με την δυνατότητα αυτήν πολλές φορές κάνουν χρήση απείρων γραμμών τέτοιου κώδικα, με στόχο τον έλεγχο της εφαρμογής, χωρίς να έχουν γράψει ούτε μία γραμμή JavaScript.

Όλα καλά και όλα ωραία, το πρόβλημα όμως ξεκινάει όταν κάτι δεν λειτουργεί καλά και πάνω στην απελπισία μας προσπαθούμε να αλλάξουμε κάποιες παραμέτρους και σπαταλήσουμε ατελείωτες ώρες χαζεύοντας αποχωνωμένοι τον κώδικα του JQuery. Νομίζατε ότι είναι κάτι που έχει συμβεί μόνο σε εσάς; Ε ναι λοιπόν δεν είστε οι μόνοι, και αυτό σε καμία περίπτωση δεν πρέπει να σας αποθαρρύνει στο να δημιουργείτε αξιόλογες εφαρμογές. Δείτε το ως πρόκληση, σκεφτείτε ότι εάν καταλάβουμε το κώδικα που βρίσκετε πίσω από το JQuery τότε τίποτε δεν μπορεί να σας σταθεί σαν εμπόδιο και δεν είναι έξω από τον έλεγχο σας, στον κόσμο της JavaScript.

Σχεδόν κάθε στοιχείο σε μια ιστοσελίδα, μέσα από το πιο δημοφιλές λειτουργικό σύστημα στον πλανήτη - το πρόγραμμα περιήγησης, είναι προσιτή και μπορούν τα χαρακτηριστικά της να αλλάξουν. Νέα μπορούν να δημιουργηθούν, παλαιά να τροποποιηθούν ή και να διαγραφούν. Το πληκτρολόγιο μας πλέον αποτελεί το πινέλο και το πρόγραμμα περιήγησης γίνεται ο καμβάς μας. Το JQuery, μας δίνει την δυνατότητα αυτήν όμως πέρα από τα ψηφία και τους κώδικες που αποτελούν το πινέλο μας θα πρέπει να κατανοήσουμε και το χαρτί που βρίσκεται κάτω από αυτά. Είμαστε σίγουρος ότι ένα μεγάλο ποσοστό όλων αυτών που κάνουν χρήση του JQuery, δεν σχέση με την JavaScript και δεν μπορούν να αντιληφθούν ότι το JQuery δεν είναι τίποτα άλλο από μια συλλογή JavaScript's. Δεν είναι ένα νέο περιβάλλον προγραμματισμού, είναι ένας σωρός JavaScript's που δημιουργούν ένα τοίχος μεταξύ αυτού που το χρησιμοποιεί και του ίδιου του κώδικα, με αποτέλεσμα να βλέπουμε οτιδήποτε άλλο εκτός από JavaScript's.

Τα περισσότερα προγράμματα περιήγησης γνωρίζουν το σύμβολο \$ (), το οποίο μας επιτρέπει όπως έχουμε πει να επιλέγουμε διάφορα αντικείμενα της σελίδας και να τα τροποποιούμε. Εάν έχουμε το παρακάτω αντικείμενο:

```
<div id="example"> </div>
```

Για να το επιλέξουμε αρκεί να γράψουμε \$ ("#example"). Από αυτό το σημείο και πέρα μπορούμε να το τροποποιήσουμε όπως εμείς επιθυμούμε. Τι σημαίνει όμως \$ (); Πώς λειτουργεί; Είναι κάτι μαγικό; Ας θυμηθούμε ότι το \$ () είναι ένας επιλογέας του JQuery ο οποίος μπορεί να πάρει διάφορα ορίσματα χωρισμένα μεταξύ τους με κόμμα. Ένα δηλαδή θέλαμε να πάρουμε το πρώτο DOM αντικείμενο που ταιριάζει με το παραπάνω ID (example), θα έπρεπε να γράψουμε σε JQueryτων εξής κώδικα \$ ('#example')[0]. Αυτό με JavaScript θα

το πετυχαίναμε εάν γράφαμε `document.getElementById ("example")`. Πειραματισμοί που γίνανε σχετικά με τις δύο αυτές μεθόδους έδειξαν ότι η `getElementById`, είναι γρηγορότερη σε σχέση με `$ ()` αλλά μόνο σε ότι αφορά στην επιλογή συγκεκριμένου στοιχείου με συγκεκριμένη ταυτότητα, και όχι ένα σύνολο από το στοιχείο αυτό. Το JQuery μας βοηθάει στην επιλογή ενός μεγάλου αριθμού αντικειμένων με μία πληθώρα τεχνικών, εάν όμως ξέρουμε τι είναι αυτό που θέλουμε να επιλέξουμε καλό είναι να χρησιμοποιούμε JavaScript. Σε τι μεταφράζεται αυτό για εμάς; Με ταχύτητα.

Οι μέθοδοι που μας βοηθούν να μετακινούμε αντικείμενα πάνω στην ιστοσελίδα μας, είναι μία ακόμα λειτουργία που αποτελεί τεράστια δυνατότητα η οποία μας λύνει τα χέρια και μας απαλλάσσει από μεγάλο κόπο εξουθενωτικού προγραμματισμού. Αρκεί να καταλάβουμε ότι για να μετακινήσουμε ένα αντικείμενο 100px δεξιά, το μόνο που πρέπει να γράψουμε είναι:

```
$ ("#element").animate ({ left: 100px; });
```

Εάν την κίνηση αυτήν θα θέλαμε να την υλοποιήσουμε με JavaScript θα γράφαμε τον ακόλουθο κώδικα:

1. `var example = null; // object`
2. `function doMove () {`
3. `example.style.left = parseInt (example.style.left)+1+'px';`
4. `setTimeout (doMove,20); // call doMove in 20msec`
5. `}`
6. `function init () {`
7. `example = document.getElementById ('example');`
8. `example.style.left = '0px'; // set its initial position to 0px`
9. `doMove (); // start animating`
10. `}`
11. `window.onload = init;`

Παρατηρούμε ότι αρχικά πρέπει να γράψουμε πολύ μεγαλύτερο κώδικα και πολύ πιο πολύπλοκο. Τι θα γινόταν εάν θέλαμε να υλοποιήσουμε σε JavaScript μια πιο πολύπλοκη διαδικασία της JQuery την `$ ("example").hide ()`; Με τον τρόπο αυτό μπορούμε να κρύψουμε αντικείμενα τα οποία δεν μας είναι χρήσιμα. Η ίδια εντολή θα μπορούσε να γίνει και με τον επόμενο τρόπο και να δώσει το ίδιο αποτέλεσμα

```
$ ("#example").css ("display","none");
```

Η CSS είναι μια πολύ δυνατή μέθοδος και μας δίνει την δυνατότητα να τροποποιούμε τα χαρακτηριστικά αντικειμένων.

Είναι άραγε εύκολο να πούμε με σιγουριά ότι το ένα η το άλλο είναι καλύτερο; Η θα πρέπει να λειτουργούμε συνδυαστικά και να εκμεταλλευόμαστε τα θετικά του ενός έναντι του άλλου; Για παράδειγμα να κάνουμε χρήση του JQuery για τα στοιχεία του όσον αφορά την επιλογή αντικειμένων και την τροποποίηση τους. αντί δηλαδή να γράφουμε ολόκληρο μακρινάρι `document.getElementById ("element")`, μπορούμε να χρησιμοποιούμε το `$ ("#element")`. Όχι ότι τα δύο αυτά δεν έχουν σχέση, γιατί το JQuery, έχει έναν ενσωματωμένο ελεγκτή, ο οποίος μεταφράζει το `$ ("#element")` σε `document.getElementById ("element")`.

Μέχρι εδώ όλα καλά, αποτελούν όμως την κορυφή του παγόβουνου. Μιλώντας επιστημονικά το JQuery αποτελείται από μια συλλογή η οποία βοηθά στην τεχνική της μεθόδου του μη επαναλαμβανόμενου κώδικα (D.R.Y. Don't Repeat Yourself). Είναι επιτρεπτό και επιβάλλεται η χρήση του JQuery, αλλά σε καμία περίπτωση η εξάρτηση από αυτό. Θα έλεγα ότι το JQuery είναι ένα εργαλείο, η JavaScript όμως είναι ο κήπος. Η γνώση του JQuery στην σύγχρονη εποχή που ζούμε αποτελεί πλεονέκτημα στην δημιουργία όμορφων ιστοσελίδων, καθώς επίσης και στην δημιουργία ευανάγνωστου και ταχύτατου κώδικα. Όλα τα εργαλεία όμως είναι καλά μόνο στα χέρια ενός καλού τεχνίτη.

3.7 Εργαλεία Ανάπτυξης

Σε όλες τις εφαρμογές προηγουμένως, θεωρούμε ότι η τεκμηρίωση θα ήταν χρήσιμη στο να κατανοήσουμε τον τρόπο λειτουργίας του, έτσι ώστε να βοηθούμε να τις εντάξουμε στις δικές μας εφαρμογές. Χρήσιμη βέβαια αυτή η τεκμηρίωση, τίποτα όμως δεν θα μπορούσε να υποκαταστήσει ένα καλό σύνολο εργαλείων ανάπτυξης λογισμικού. Ευτυχώς, υπάρχουν πολλά πακέτα λογισμικού που διατίθενται για τον έλεγχο και τον εντοπισμό σφαλμάτων JavaScript κώδικα, και τα περισσότερα από αυτά είναι διαθέσιμα δωρεάν. Τα πιο βασικά αυτά μπορείτε να τα βρείτε στους συνδέσμους που σας παραθέτουμε, τα οποία για τον Firefox είναι:

Firebug

Το Firebug^[36] αποτελεί μια προέκταση για τον Firefox και είναι κατάλληλο για ανάπτυξη jQuery εφαρμογών.

Web developer toolbar^[37]

Το συγκεκριμένο όχι μόνο αλληλεπικαλύπτει το Firebug στον τομέα της DOM επιθεώρησης, αλλά περιέχει επίσης εργαλεία για κοινές εργασίες όπως η χειραγώγηση των

cookies, την επιθεώρηση μορφή, και αλλαγή μεγέθους σελίδας. Μπορείτε επίσης να χρησιμοποιήσετε αυτήν τη γραμμή εργαλείων για την γρήγορη και εύκολη απενεργοποίηση της Javascript για ένα site για να διασφαλιστεί ότι υποβαθμίζεται η λειτουργικότητα με χάρη όταν το πρόγραμμα περιήγησης του χρήστη είναι λιγότερο ικανό.

Venkman

Το Venkman [38] είναι το επίσημο πρόγραμμα εντοπισμού σφαλμάτων JavaScript για το Mozilla. Παρέχει ένα Compiler προβλημάτων, περιβάλλον που θυμίζει το σύστημα GDB για την αποσφαλμάτωση προγραμμάτων που είναι γραμμένα σε άλλες γλώσσες.

Όπως είναι φυσικό παρόμοια εργαλεία αποσφαλμάτωσης και ελέγχου, υπάρχουν και για τον Explorer:

Microsoft Internet Explorer Developer Toolbar [39]

Η Γραμμή Εργαλείων αυτή, παρέχει κυρίως ενόψει δέντρου όλα τα DOM αντικείμενα για μια ιστοσελίδα. Τα στοιχεία μπορούν να εντοπίζονται οπτικά, να τροποποιούνται on the fly σύμφωνα με τους νέους κανόνες των CSS3.

Microsoft Visual Web Developer

Το πακέτο Microsoft Visual Web Developer [40] μπορεί να χρησιμοποιηθεί για τον έλεγχο javascript εφαρμογών.

Για να τρέξουμε την ελεύθερη έκδοση του εργαλείου αυτού μπορούμε να το επιτύχουμε ακολουθώντας οδηγίες που βρίσκονται στον παρακάτω ιστοχώρο:

<http://www.berniecode.com/blog/2007/03/08/how-to-debug-javascript-with-visual-web-developer-express/>

DebugBar

Το εργαλείο DebugBar [41] παρέχει την δυνατότητα τόσο του ελέγχου των DOM αντικειμένων όσο και της αποσφαλμάτωσης Javascript κώδικα.

δεν θα μπορούσαμε να παραλείψουμε παρόμοια εργαλεία για το Safari και την Opera τα οποία για το μεν πρώτο είναι τα ακόλουθα:

Develop Menu

Από το Safari 3.1, μια επιλογή στην καρτέλα Για προχωρημένους του μενού Προτιμήσεις προσφέρει ένα ειδικό μενού που ονομάζεται Ανάπτυξη. Με αυτό το μενού ενεργοποιείται, ένας Web Inspector και μία JavaScript Console είναι διαθέσιμη.

Όσον αφορά την Opera η οποία αν και έχει περιορισμένο μερίδιο της αγοράς ως ένας desktop browser, είναι ένας σημαντικός παίκτης στα ολοκληρωμένα συστήματα και στις συσκευές κινητών τηλεφώνων. Οι δυνατότητες του, πρέπει να εξεταστούν προσεκτικά κατά τη διάρκεια της ανάπτυξης ιστοσελίδων.

Το Dragonfly, αν και βρίσκεται ακόμα στα αρχικά του στάδια, είναι ένα πολλά υποσχόμενο περιβάλλον εντοπισμού σφαλμάτων για το προγράμματα περιήγησης Opera σε υπολογιστές ή φορητές συσκευές.

3.8 Συμπέρασμα

Φτάνοντας στο τέλος αυτού του κεφαλαίου, το αντικείμενο του οποίου δεν είναι δυνατόν να εξαντληθεί σε λίγες μόνο σελίδες. Σκοπός μας βέβαια δεν ήταν να γράψουμε ένα εγχειρίδιο αλλά περισσότερο να σας δώσουμε ένα κίνητρο να ασχοληθείτε με κάτι καινούριο και αρκετά ευέλικτο όπως το JQuery. Στην συνέχεια στην πορεία αυτού του συγγράμματος θα ξανά αναφερθούμε στ ο εργαλείο αυτό στο κεφάλαιο που θα έχει ως αντικείμενο το AJAX.

4

CSS3

Σκοπός

Σε αυτό το Κεφάλαιο θα εξετάσουμε:

- τι είναι το CSS3,
- τι προσφέρει,
- την HTML5, και
- το FontFace.

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να κατανοήσει τη χρησιμότητα του CSS3,
- να κατανοήσει την HTML5,
- να αντιληφθεί τη δύναμη του FontFace, και
- να καταλάβει ποιο το κέρδος εάν κάνει χρήση όλων αυτών μαζί.

4.1 Παρουσίαση

Στο κεφάλαιο αυτό θα ασχοληθούμε με κάτι εξαιρετικά ενδιαφέρον για την σχεδίαση μιας ιστοσελίδας και κυρίως για τα γραφικά που μπορεί να έχει μια ιστοσελίδα απλά και μόνο με την χρήση του CSS3 [20]. Με τον τρόπο αυτό έχουμε στην διάθεση μας υψηλής ποιότητας γραφικά χωρίς να κάνουμε εφαρμογή γνωστών προγραμμάτων γραφιστικής, τα οποία μας παρέχουν ικανοποιητικά αποτελέσματα μεν, όμως μας φορτώνουν με βαριά πολλές φορές γραφικά τα οποία κάνουν την φόρτωση της εφαρμογής αργή. Επίσης θα κάνουμε ανάπτυξη με την βοήθεια μιας εφαρμογής του HTML5 [13-14] το οποίο αποτελεί την εξέλιξη και μία επανάσταση της HTML [22]. Κλείνοντας θα αναφερθούμε στο Font Face μία τεχνική για την χρήση υψηλής ποιότητας γραμματοσειρών στην εφαρμογή μας. Ο λόγος που ονομάσαμε το κεφάλαιο CSS3, είναι επειδή στην εφαρμογή που συνοδεύει αυτήν την εργασία, έχει γίνει εκτενής χρήση του. Σε καμία περίπτωση δεν υποτιμάμε την HTML5 η οποία αποτελεί την επανάσταση και για τη οποία θα χρειάζονταν αποκλειστικά βιβλία για την περιγραφή της.

4.2 CSS3

Από μόνο του αποτελεί μια επανάσταση στο πεδίο της δημιουργικής σχεδίασης μιας ιστοσελίδας. Αφού δίνει ατελείωτες δυνατότητες στους σχεδιαστές, που ο μόνος περιορισμός που έχουν τώρα πια είναι η ίδια η φαντασία τους. Ας πούμε λίγα λόγια για όσους δεν γνωρίζουν τι είναι το CSS. Η ονομασία αυτή προέρχεται από τα αρχικά των λέξεων ‘Cascading Style Sheets’ η οποία είναι μια τεχνολογία της W3C, η οποία επιτρέπει στους σχεδιαστές ιστοσελίδων να καθορίζουν την παρουσίαση των αντικειμένων μέσα στο χώρο της σελίδας. Και όταν λέμε παρουσίαση εννοούμε την διάταξη στον χώρο και την εμφάνιση τους.

Το CSS από μόνο του αποτελεί ένα μεγάλο πεδίο, και καλό θα ήταν πριν προχωρήσετε στη ανάγνωση αυτού του συγγράμματος να έχετε κάνει μια επανάληψη [32], γιατί εδώ θα παρουσιάσουμε τι νέο υπάρχει σχετικά με αυτό και ονομάζεται CSS3. Το CSS3 είναι ένα εργαλείο το οποίο ακόμα δεν υποστηρίζεται από όλους του Browsers ωστόσο όμως αποτελεί το μέλλον, και καλό θα ήταν εάν θέλετε να παίξετε με τα παραδείγματα που θα αναπτυχθούν παρακάτω, να το κάνετε σε διακομιστή που τα υποστηρίζει όπως το Firefox.

4.2.1 Γιατί να ασχοληθούμε με το CSS3 αφού ακόμα δεν υποστηρίζεται πλήρως;

Γιατί είναι ωραίο να μαθαίνουμε καινούρια πράγματα και κυρίως όταν αυτά είναι ακόμα σε εξέλιξη. Είναι πάρα πολλοί αυτοί που θα ρωτούσαν, πως είναι δυνατό να δημιουργήσουμε στρογγυλές γωνίες στα παράθυρα ή μέσα σε DIVS, άλλοι θα ρωτούσαν σχετικά με την διαφάνεια και τις σκιάσεις, άλλοι σχετικά με την χρωματική διαβάθμιση (gradient), πάρα πολλές τέτοιου είδους ερωτήσεις. Πολλές από αυτές θα απαντηθούν σε αυτό το κεφάλαιο και θα έχουν σχέση με τα πιο γνωστά προβλήματα σχεδίασης που μπορεί να συναντήσουμε, για μια πιο αναλυτική παρουσίαση καλό θα ήταν να καταφύγετε σε κάποιο βιβλίο αποκλειστικά για σχεδίαση με CSS3, η και μέσα από πάρα πολλές σελίδας του διαδικτύου. Ας ξεκινήσουμε λοιπόν με έναν αναλυτικό πίνακα σχετικά με τους διακομιστές και τις τεχνολογίες που αυτοί υποστηρίζουν, τουλάχιστον μέχρι την στιγμή που γράφουμε αυτό το κεφάλαιο, αφού όπως είπαμε τα πράγματα συνεχώς εξελίσσονται. Ο πίνακας στο σχήμα 4.1, θα πρέπει να είναι κάτι που θα πρέπει να το συμβουλευέστε συνεχώς κατά την χρήση των τεχνολογιών που θα παρουσιάσουμε στην συνέχεια.

											
	4	5	3	3.6	6	7	8	9	10	10.5	4
<u>Border Colour</u>	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
<u>Border Image</u>	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓
<u>Border Radius</u>	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓
<u>Box Shadow</u>	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓
<u>@Font-face</u>	✓	✓	✗	✓	?	?	?	?	✓	✓	✓
<u>Gradients</u>	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
<u>HSL / HSLA</u>	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
<u>Multiple Backgrounds</u>	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓
<u>Multiple Columns</u>	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓
<u>Opacity</u>	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
<u>RGB / RGBA</u>	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
<u>Text Shadow</u>	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
<u>Transform</u>	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓

Σχήμα 4.1

4.2.2 Selectors

Οι CSS Selectors είναι ένα εξαιρετικά ισχυρό εργαλείο: μας επιτρέπουν να στοχεύουν συγκεκριμένα στοιχεία της HTML χωρίς να χρειάζεται να στηριζόμαστε σε άσκοπες τάξεις, ταυτότητες και JavaScripts. Οι περισσότεροι από αυτούς δεν είναι κάτι νέο για το CSS3, αλλά δεν είναι τόσο ευρέως διαδεδομένοι όσο θα έπρεπε. Οι Προηγμένοι αυτοί επιλογείς μπορεί να είναι χρήσιμοι εάν προσπαθούμε να επιτύχουμε ένα καθαρό, ελαφρύ κώδικα και το διαχωρισμό μεταξύ της δομής και της παρουσίασης. Μπορούν να μειώσουν τον αριθμό των τάξεων και ταυτοτήτων στα tags και να καταστήσουν ευκολότερο για τους σχεδιαστές να διατηρήσουν ένα εύκολο στην ανάγνωση στυλ.

Τρία νέα ήδη επιλογέων είναι πλέον διαθέσιμα στο CSS3:

- `[att^="value"]`
επιλέγει αντικείμενα τα οποία έχουν μια συγκεκριμένη τιμή σε ένα χαρακτηριστικό τους
- `[att$="value"]`
επιλέγει αντικείμενα τα οποία τελειώνουν με μια συγκεκριμένη τιμή σε ένα χαρακτηριστικό τους
- `[att*="value"]`
επιλέγει αντικείμενα τα οποία εμπεριέχουν μια συγκεκριμένη τιμή σε ένα χαρακτηριστικό τους.

Παράδειγμα των παραπάνω είναι ο κώδικας που ακολουθεί και ο οποίος δίνει κάποια χαρακτηριστικά, σε όλα τα `<a>` στα οποία ο τίτλος τους εμπεριέχει την λέξη παοκ.

```
a[title$="παοκ"] { position: absolute;
                    top: 0; right: 0;
                    display: block;
                    width: 140px;
                    height: 140px;
                    text-indent: -9999px;
                    }
```

4.2.3 Multi-Column Layout

Αυτός ο νέος επιλογέας της CSS3 σας επιτρέπει να επιτύχετε πολλαπλές στήλες μέσα σε ένα έγγραφο χωρίς να χρειάζεται να χρησιμοποιήσετε πολλαπλές divs. Το πρόγραμμα περιήγησης ερμηνεύει τις ιδιότητες αυτές και δημιουργεί τις στήλες, δίνοντας στο κείμενο μιας μορφή που μοιάζει με εφημερίδα.



Σχήμα 4.2

Ο καλύτερος τρόπος για να σας το δείξουμε είναι μέσα από το παρακάτω παράδειγμα με την βοήθεια του οποίου μπορούμε να πετύχουμε εμφάνιση παρόμοια με αυτήν που απεικονίζεται στο πιο πάνω σχήμα.

```
.index #content div {  
  
    -webkit-column-count : 4;  
  
    -webkit-column-gap : 20px;  
  
    -moz-column-count : 4;  
  
    -moz-column-gap : 20px; }  
  
}
```

Με τον συγκεκριμένο επιλογέα καθορίζουμε τον αριθμό των στηλών σε 4, στην πρώτη γραμμή, το κενό μεταξύ των στηλών στην δεύτερη γραμμή και το πλάτος της στήλης με την ιδιότητα `-column-width`. Εάν θελήσουμε να χωρίσουμε τις στήλες με μία γραμμή μεταξύ τους, αυτό είναι απλό με την βοήθεια του παρακάτω κώδικα.

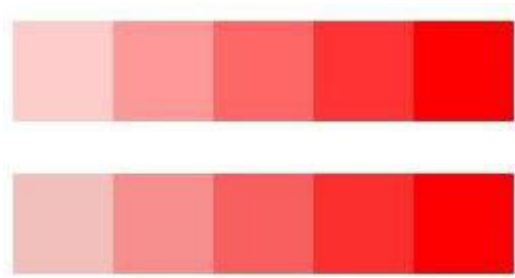
```

div {
    column-rule: 1px
        solid #000000;
}

```

4.2.4 CSS3 HSL & HSLA

Στην ενότητα αυτήν θα μάθουμε όχι μόνο πώς να χρησιμοποιούμε HSLA χρώματα, αλλά και πώς να παίζουμε με την διαφάνεια των αντικειμένων. Το HSLA επιτρέπει τέσσερις τιμές οι οποίες καθορίζουν την διαφάνεια ενός αντικειμένου. Αυτήν την στιγμή θα μπορούσαμε να πούμε με ευκολία ότι μόνο το firefox και το safari υποστηρίζουν αυτήν την δυνατότητα. Επειδή όλα τα πράγματα γίνονται πιο κατανοητά με παραδείγματα δείτε το παρακάτω σχήμα 4.3.



Σχήμα 4.3

Παρατηρώντας το ποιο πάνω σχήμα και κυρίως, εάν έχετε κάνει το κόπο να κατεβάσετε τον firefox 3 beta, θα καταλάβετε ότι η πρώτη γραμμή του σχήματος χρησιμοποιεί HSLA. Ο κώδικας που χρησιμοποιείται φαίνεται πιο κάτω.

```

<div style="background-color: hsla (0,100%,50%,0.2);"></div>
<div style="background-color: hsla (0,100%,50%,0.4);"></div>
<div style="background-color: hsla (0,100%,50%,0.6);"></div>
<div style="background-color: hsla (0,100%,50%,0.8);"></div>
<div style="background-color: hsla (0,100%,50%,1);"></div>

```

Για την δεύτερη γραμμή χρησιμοποιούμε το rgb, ο κώδικας είναι ο εξής.


```
<div style="background-color: rgb (243,191,189);"></div>
<div style="background-color: rgb (246,143,142);"></div>
<div style="background-color: rgb (249,95,94);"></div>
<div style="background-color: rgb (252,47,47);"></div>
<div style="background-color: rgb (255,0,0);"></div>
```

Θα μας πείτε τώρα τι είναι όλα αυτά τα νούμερα και οι ποσοστιαίες διαβαθμίσεις. Τα νούμερα αυτά αποτελούν χρώματα, μια λεπτομερής περιγραφή μπορεί να βρεθεί W3C CSS3 Color Candidate Recommendation [43].

Η alpha που αποτελεί τον τέταρτο χαρακτήρα της hsla και περιγράφεται με ένα νούμερο από το 1.0 μέχρι το 0.0. ένα αντικείμενο με alpha 1.0 είναι συμπαγές και απολύτως ορατό, αντίθετα το ίδιο αντικείμενο με alpha 0.0 είναι μη ορατό. Κάθε οποιαδήποτε τιμή μεταξύ αυτών των δύο καθορίζει πόσο διαφανές είναι το χρώμα του αντικειμένου.

4.2.5 Border-radius: Στρογγυλοποίηση αντικειμένων με CSS!

Ένα επίσης πολύ ενδιαφέρον πεδίο που απασχολεί, και δόθηκε η λύση του μέσα από το CSS3 είναι η καμπυλότητα στις γωνίες που τις βλέπουμε όλο και πιο συχνά στις ιστοσελίδες. Παλαιότερα για να το πετύχουν αυτό έφτιαχναν σχήματα με διάφορα σχεδιαστικά εργαλεία και τα τοποθετούσαν στις γωνίες των αντικειμένων τις οποίες και επιθυμούσαν να στρογγυλοποιήσουν. Αυτό όμως ήταν τότε, τώρα το CSS3 έρχεται να μας γλιτώσει από ώρες σχεδιασμού και μέσα από απλές πια διαδικασίες. Όπως το παρακάτω σχήμα 4.4



This box should have a rounded corners for Firefox, Safari/Chrome, Opera and IE9.

Σχήμα 4.4

Ο κώδικας για να το πετύχουμε είναι σχετικά απλός.

```
#example1 {
border-radius:15px;
}
```

Για την ώρα όμως θα πρέπει να κάνουμε χρήση της `-moz-` για είναι το αποτέλεσμα εμφανές και στον firefox. Έτσι θα πρέπει να γράψουμε

```
#example1 {
-moz-border-radius:15px;
border-radius:15px;
}
```

Σίγουρα θα έχετε την απορία πως αυτό δουλεύει. Η στρογγυλάδα στις γωνίες μπορεί να επιτευχθεί χρησιμοποιώντας για κάθε μία από τις γωνίες των εξής ορισμό:

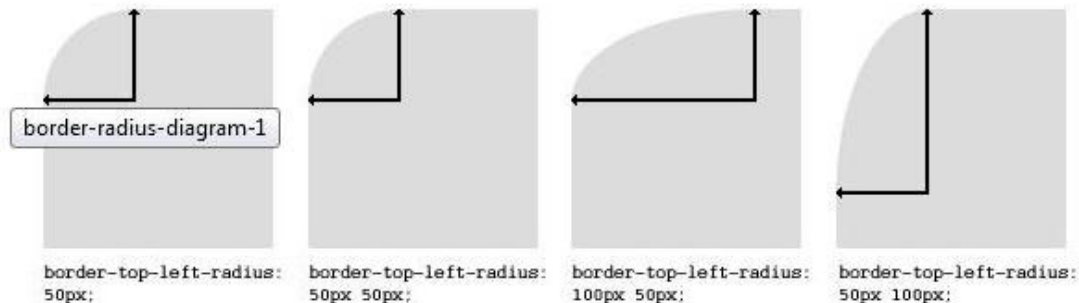
border-*-*-radius: [<length> | <%>] [<length> | <%>]?

Έτσι θα μπορούσαμε να ορίσουμε την καμπυλότητα μιας πάνω αριστερής γωνίας ως εξής:

```
border-top-left-radius: 10px 5px;
```

όταν κάνουμε χρήση δύο τιμών, μπορούν αυτές να έχουν ή τιμές σε px ή ποσοστά σε %. Οι τιμές αυτές καθορίζουν την οριζόντια και κάθετη γωνία σε μια έλλειψη 90°. η οποία καθορίζει την καμπυλότητα της εξωτερικής γωνίας.

Εάν έχουμε μία τιμή τότε αυτήν είναι κοινή τόσο για την οριζόντια γραμμή όσο και για την κάθετη. Στο σχήμα που ακολουθεί δίνονται ορισμένα παραδείγματα για το πώς μπορεί να είναι μια γωνία ανάλογα με τις τιμές που θα πάρουν τα ορίσματα της.



Σχήμα 4.5

Το μόνο σχόλιο το οποίο θα μπορούσαμε να κάνουμε στο παραπάνω σχήμα είναι, ότι εάν και οι δύο τιμές είναι μηδέν το αποτέλεσμα που θα πάρουμε είναι η ίδια η γωνία.

Μία άλλη ιδιότητα είναι η **border-radius**, η οποία μας δίνει την δυνατότητα να προσδιορίσουμε όλες τις γωνίες ταυτόχρονα. Η ιδιότητα αυτή χρησιμοποιεί ένα ή δύο σετ από τιμές, το κάθε ένα από τα οποία παίρνει από μία ως τέσσερις τιμές. Η σύνταξη της μεθόδου είναι ανάλογη της προηγούμενης και ορίζεται ως εξής :

[<length> | <percentage>]{1,4} [/ [<length> | <percentage>]{1,4}]?

Ως παραδείγματα θα μπορούσαμε να γράψουμε :

```
border-radius: 5px 10px 5px 10px / 10px 5px 10px 5px;
```

```
border-radius: 5px;
```

```
border-radius: 5px 10px / 10px;
```

Στο πρώτο παράδειγμα το πρώτο σετ με τις τιμές προσδιορίζει τις οριζόντιες τιμές και το δεύτερο σετ, που διαφοροποιείται από το πρώτο με '/', προσδιορίζει τις κάθετες τιμές.

Οι ιδιότητες αυτές έχουν συμβατότητα με τους περισσότερους διακομιστές. Ένα σχετικά πρόβλημα είναι ο mozilla ο οποίος κάνει χρήση του `-moz-` προθέματος το οποίο δέχεται ως ορίσματα δύο τιμές μέχρι την έκδοση 3.5. Από την έκδοση αυτήν και μετά δέχεται μία τιμή ανά γωνία. Μετά την έκδοση 3.5 ο mozilla ακολουθεί τα πρότυπα του W3C τοποθετώντας το πρόθεμα `-moz-` μπροστά όπως φαίνεται παρακάτω.

W3C Specification

`border-radius`

`border-top-left-radius`

`border-top-right-radius`

`border-bottom-right-radius`

`border-bottom-left-radius`

Mozilla Implementation

`-moz-border-radius`

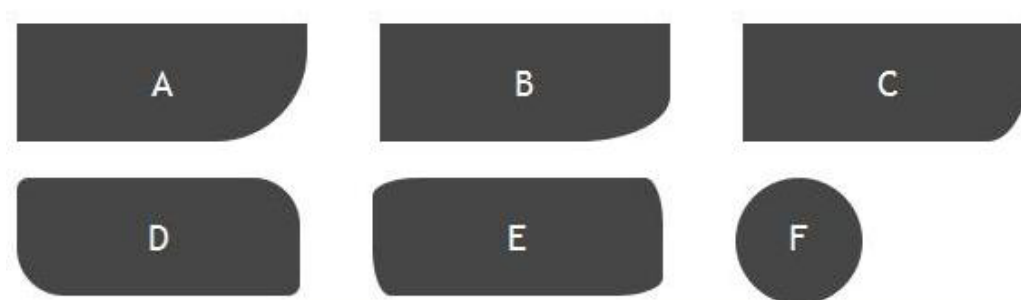
`-moz-border-radius-topleft`

`-moz-border-radius-topright`

`-moz-border-radius-bottomright`

`-moz-border-radius-bottomleft`

Αποτέλεσμα της χρήσης σε διαφορετικούς browsers φαίνεται πιο κάτω. Όλα αυτά πρέπει να δουλεύουν στον IE9, Firefox, Safari/Chrome και Opera.



Σχήμα 4.6

```
#Example_A {  
height: 65px;  
width: 160px;
```

```
-moz-border-radius-bottomright: 50px;
border-bottom-right-radius: 50px;
}
#Example_B {
height: 65px;
width:160px;
-moz-border-radius-bottomright: 50px 25px;
border-bottom-right-radius: 50px 25px;
}
#Example_C {
height: 65px;
width:160px;
-moz-border-radius-bottomright: 25px 50px;
border-bottom-right-radius: 25px 50px;
}
#Example_D {
height: 5em;
width: 12em;
-moz-border-radius: 1em 4em 1em 4em;
border-radius: 1em 4em 1em 4em;
}
#Example_E {
height: 65px;
width:160px;
-moz-border-radius: 25px 10px / 10px 25px;
border-radius: 25px 10px / 10px 25px;
}
#Example_F {
height: 70px;
width: 70px;
-moz-border-radius: 35px;
border-radius: 35px;
}
```

4.2.6 Σκίαση, ένα από τα καλύτερα νέα χαρακτηριστικά του CSS3

Η ιδιότητα **box-shadow** είναι από τις πιο συναρπαστικές ιδιότητες που μας προσφέρει το CSS3, με την οποία έχουμε την δυνατότητα να εφαρμόσουμε πολλαπλές σκιάσεις σε αντικείμενα, τόσο προς το εσωτερικό του όσο και προς το εξωτερικό του σχήματος. Επίσης οι σκιάσεις αυτές μπορούν να έχουν πολλαπλούς χρωματισμούς, δίνοντας στο αντικείμενο που θα εφαρμοστούν ένα εκπληκτικό εφέ. Οι διακομιστές που την υποστηρίζουν είναι η τελευταία έκδοση του Mozilla το Webkit (Safari/Chrome/Konqueror), Opera και ο IE9. Ο Mozilla και το Webkit για την εφαρμογή της σκίασης απαιτούν τα προθέματα `-moz-` και `-webkit-` μπροστά από το όνομα της ιδιότητας (να σημειωθεί ότι από την έκδοση 4+ του Mozilla δεν θα χρειάζεται το συγκεκριμένο πρόθεμα).

Ένα χαρακτηριστικό παράδειγμα του τι εννοούμε σκίαση φαίνεται στο πιο κάτω σχήμα.



Σχήμα 4.7

Ο κώδικας για να πετύχουμε το αποτέλεσμα αυτό είναι:

```
#example1 {  
    box-shadow: 10px 10px 5px #888;  
}
```

Για να μπορεί να είναι το εφέ συμβατό με τον Mozilla και το webkit θα πρέπει να τροποποιήσουμε τον πιο πάνω κώδικα ως εξής.

```
#example1 {
  -moz-box-shadow: 10px 10px 5px #888;
  -webkit-box-shadow: 10px 10px 5px #888;
  box-shadow: 10px 10px 5px #888;
}
```

Πώς δουλεύει η σκιά;

Η ιδιότητα `box-shadow` δέχεται τιμές σκίασης που χωρίζουν μεταξύ τους με το κόμμα. Συνήθως οι τιμές είναι 2-4, (προσδιορίζοντας τιμές για την οριζόντια μετατόπιση, κάθετη μετατόπιση, προαιρετικά θάμπωμα απόσταση και προαιρετικά απόσταση εξάπλωση της σκιάς), και μια προαιρετική τιμή χρώμα και η λέξη-«ένθετο (`inset`)» (για να δημιουργήσει μια εσωτερική σκιά, και όχι το προεπιλεγμένο που είναι η εξωτερική σκιά).

Ίσως όλα αυτά να ακούγονται πολύπλοκα πιστέψτε με όμως δεν είναι, αξίζει να ασχοληθείτε λίγο για να τα καταλάβετε και το αποτέλεσμα θα σας καταπλήξει. Η σύνταξη που ακολουθεί η ιδιότητα αυτή είναι η παρακάτω:

box-shadow: none|<shadow> [, <shadow>]*

<shadow> = inset? && [<length>{2,4} && <color>?]

Παραδείγματα της εφαρμογής της έχω:

`box-shadow: 10px 10px;`

`box-shadow: 10px 10px 5px #888;`

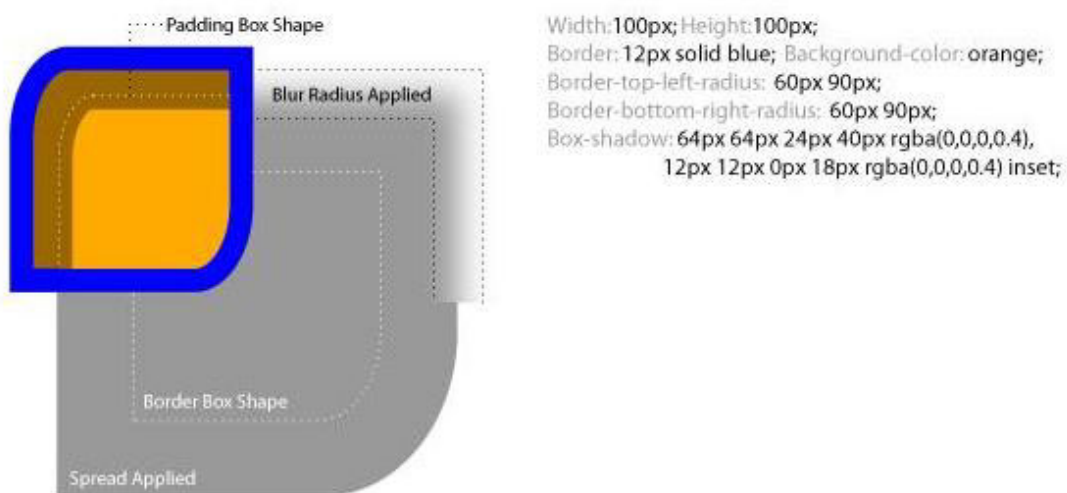
`box-shadow: inset 2px 2px 2px 2px black;`

`box-shadow: 10px 10px #888, -10px -10px #f4f4f4, 0px 0px 5px 5px #cc6600;`

Για να προσδιορίσουμε τη βασική σκιά ενός αντικειμένου το πρώτο βήμα είναι να προσδιοριστεί το σχήμα της σκιάς, επιλέγοντας 2-4 τιμές. Η πρώτη τιμή καθορίζει την οριζόντια μετατόπιση της σκιάς, με θετική αξία αντισταθμίζει τη σκιά στα δεξιά του στοιχείου, και με αρνητική τιμή προς τα αριστερά. Η δεύτερη τιμή προσδιορίζει την κάθετη τοποθέτηση της σκιάς, με θετική αξία συμψηφισμό της σκιάς από το κάτω μέρος του στοιχείου, και με αρνητική από την πάνω κορυφή. Δηλαδή από πού να ξεκινάει η σκιά και τι κατεύθυνση να έχει ως προς τον κατακόρυφο άξονα. Μία προαιρετική τρίτη τιμή καθορίζει την απόσταση θολώματος της σκιάς. Μόνο θετικές τιμές επιτρέπονται, και όσο μεγαλύτερη

είναι η τιμή, τόσο πιο άκρη της σκιάς είναι θολή. Στις προδιαγραφές δεν περιλαμβάνεται ένας ακριβής αλγόριθμος για το πώς το θόλωμα απόσταση θα πρέπει να υπολογίζεται.

Μια προαιρετική τέταρτη τιμή, η οποία παρέχει την δυνατότητα να καθορίζουμε την απόσταση εξάπλωση της σκιάς. Μια θετική τιμή θα προκαλέσει το σχήμα σκιά να επεκταθεί προς όλες τις κατευθύνσεις, ενώ μια αρνητική τιμή θα προκαλέσει το σχήμα σκιά να συσταλεί. Το παρακάτω διάγραμμα (που λαμβάνονται από το W3C, “W3C Backgrounds and Borders Candidate Recommendation [43]”) προσφέρει ένα καλό παράδειγμα του εφέ της διάδοσης (spread) και εξασθένισης (blur) της σκιάς.



Σχήμα 4.8

Στην συνέχεια επιλέξαμε παραδείγματα της διάδοσης (spread) και εξασθένισης (blur) της σκιάς σε διαφορετικούς διακομιστές.



Σχήμα 4.9

Για την υλοποίηση των παραπάνω εφέ έχουμε τους εξής κώδικες.

Το παράδειγμα A, χρησιμοποιεί σκίαση προς τα αριστερά και προς τα πάνω 5px:

```
#Example_A {  
-moz-box-shadow: -5px -5px #888;  
-webkit-box-shadow: -5px -5px #888;  
box-shadow: -5px -5px #888;  
}
```

Το B είναι το ίδιο με την διαφορά ότι έχει εξασθένιση της τάξης 5px:

```
#Example_B {  
-moz-box-shadow: -5px -5px 5px #888;  
-webkit-box-shadow: -5px -5px 5px #888;  
box-shadow: -5px -5px 5px #888;  
}
```

Στο C έχουμε το ίδιο παράδειγμα μόνο που έχει εξάπλωση σκιάς στα 5px:

```
#Example_C {  
-moz-box-shadow: -5px -5px 0 5px #888;  
-webkit-box-shadow: -5px -5px 0 5px#888;  
box-shadow: -5px -5px 0 5px #888;  
}
```

Το D έχει και εξάπλωση και θόλωση της τάξης του 5px:

```
#Example_D {  
-moz-box-shadow: -5px -5px 5px 5px #888;  
-webkit-box-shadow: -5px -5px 5px 5px#888;  
box-shadow: -5px -5px 5px 5px #888;  
}
```

Το E δείχνει σκίαση με καθόλου εξάπλωση παρά μόνο με θόλωση 5px:

```
#Example_E {  
-moz-box-shadow: 0 0 5px #888;  
-webkit-box-shadow: 0 0 5px#888;  
box-shadow: 0 0 5px #888;  
}
```

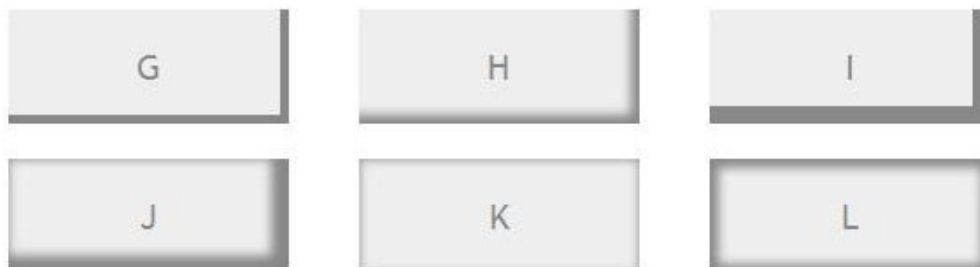
Το F δεν έχει καμία κατεύθυνση παρά μόνο θόλωση και εξάπλωση 5px:

```
#Example_F {  
-moz-box-shadow: 0 0 5px 5px #888;  
-webkit-box-shadow: 0 0 5px 5px#888;  
box-shadow: 0 0 5px 5px #888;  
}
```

Μετά τα βασικά σχετικά με την σκίαση θα μπορούσαμε να πούμε ότι είμαστε στην θέση να εφαρμόσουμε το συγκεκριμένο εφέ στις ιστοσελίδες μας. Το επόμενο στάδιο της ανάλυσης μας θα είναι η περιγραφή της εσωτερικής σκίασης και κλειδί του εφέ είναι η λέξη `inset`.

Inset

Η συγκεκριμένη λέξη πριν από τις τιμές που καθορίζουν το μήκος και το χρώμα της σκιάς. Η εύρεση αυτής της λέξης δίνει την διαταγή στον διακομιστή ότι η σκιά θα πρέπει να συνταχθεί προς το εσωτερικό του σχήματος. Τα παραδείγματα που περιγράψαμε προηγουμένως θα τα δούμε ξανά, αυτήν την φορά με την χρήση της `inset`.



Σχήμα 4.10

Το G δείχνει εσωτερική σκίαση με κατεύθυνση στα αριστερά και προς τα πάνω κατά 5px:

```
#Example_G {  
-moz-box-shadow: inset -5px -5px #888;  
-webkit-box-shadow: inset -5px -5px #888;  
box-shadow: inset -5px -5px #888;  
}
```

Το H είναι το ίδιο με το προηγούμενο με την διαφορά ότι έχει θόλωση της τάξης των 5px:

```
#Example_H {  
-moz-box-shadow: inset -5px -5px 5px #888;  
-webkit-box-shadow: inset -5px -5px 5px #888;  
box-shadow: inset -5px -5px 5px #888;  
}
```

Στο I έχουμε εξάπλωση 5px:

```
#Example_I {  
-moz-box-shadow: inset -5px -5px 0 5px #888;  
-webkit-box-shadow: inset -5px -5px 0 5px #888;  
box-shadow: inset -5px -5px 0 5px #888;  
}
```

Στο J έχουμε εξάπλωση και θόλωση 5px:

```
#Example_J {  
-moz-box-shadow: inset -5px -5px 5px 5px #888;  
-webkit-box-shadow: inset -5px -5px 5px 5px #888;  
box-shadow: inset -5px -5px 5px 5px #888;  
}
```

Το K έχει μόνο εσωτερική θόλωση 5px:

```
#Example_K {  
-moz-box-shadow: inset 0 0 5px #888;
```

```
-webkit-box-shadow: inset 0 0 5px#888;  
box-shadow: inner 0 0 5px #888;  
}
```

Στο L έχουμε θόλωση και εξάπλωση 5px:

```
#Example_L {  
-moz-box-shadow: inset 0 0 5px 5px #888;  
-webkit-box-shadow: inset 0 0 5px 5px#888;  
box-shadow: inset 0 0 5px 5px #888;  
}
```

Πολύ ωραία μέχρι εδώ πώς όμως μπορούμε να εφαρμόσουμε την σκίαση και την στρογγυλοποίηση των γωνιών που μελετήσαμε προηγουμένως; Μελετήστε το παράδειγμα που ακολουθεί και προσπαθήστε να το κάνετε και μόνοι σας.



Σχήμα 4.11

Στο σχήμα Q εφαρμόζουμε σκίαση προς τα κάτω και δεξιά 5px, και σφριγηλότητα επίσης 5px.

```
#Example_Q {  
-moz-border-radius: 5px;  
border-radius: 5px;  
-moz-box-shadow: 5px 5px black;  
-webkit-box-shadow: 5px 5px black;  
box-shadow: 5px 5px black;  
}
```

Το R είναι το ίδιο μόνο που εφαρμόζει και θόλωση.

```
#Example_R {
-moz-border-radius: 5px;
border-radius: 5px;
-moz-box-shadow: 5px 5px 5px black;
-webkit-box-shadow: 5px 5px 5px black;
box-shadow: 5px 5px 5px black;
}
```

Για να εφαρμόσουμε όλα αυτά τα εφέ θα πρέπει να υπάρχει και η κατάλληλη υποστήριξη από του διακομιστές όπως φαίνεται στο πιο κάτω σχήμα [44-45].

Browser	Basic support	Multiple shadows	inset keyword	Spread radius
Internet Explorer	9.0 box-shadow	9.0	9.0	9.0
Firefox (Gecko)	4.0 (2.0) box-shadow	4.0 (2.0)	4.0 (2.0)	4.0 (2.0)
	3.5 (1.9.1) -moz-box-shadow	3.5 (1.9.1)	3.5 (1.9.1)	3.5 (1.9.1)
Opera	10.5 box-shadow	10.5	10.5	10.5
Safari/Chrome (WebKit)	3.0/1.0 (522) -webkit-box-shadow	4.0/1.0 (528)	5.0/4.0 (533)	5.0/4.0 (533)

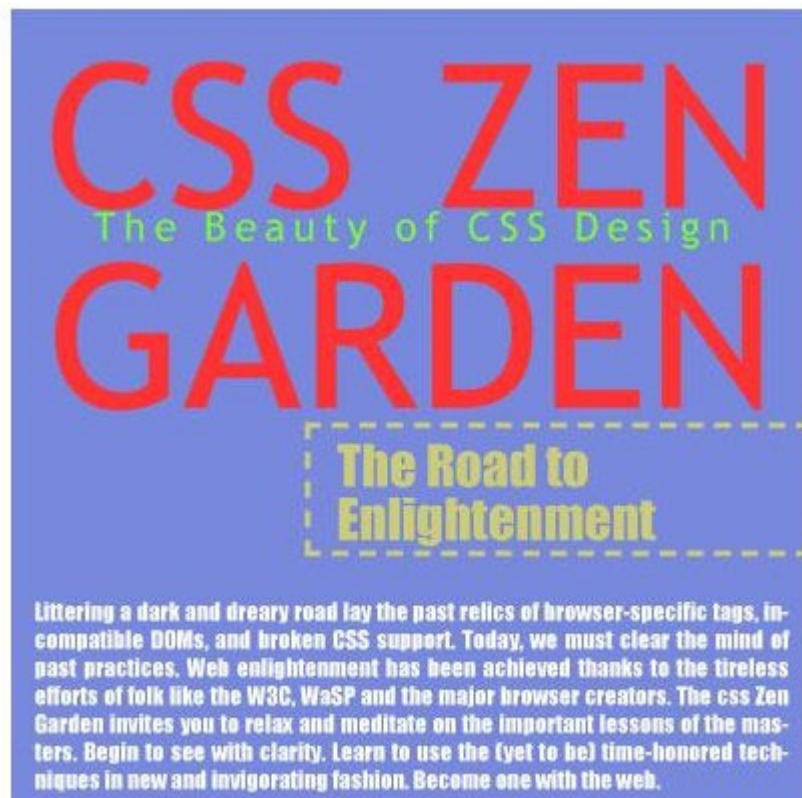
Browser support data courtesy of [Mozilla Developer Center](#).

Σχήμα 4.12

4.3 @font-face

Όπως και το προλόγισαμε ως μέρος του κεφαλαίου θα εντάξουμε και τα fontface [46], και αυτό όχι γιατί πρόκειται για κάτι καινούριου, αφού πρώτο παρουσιάστηκαν από την εποχή του CSS2.0, αλλά γιατί μας δόθηκε η ευκαιρία να ασχοληθούμε λίγο παραπάνω και μάθαμε πράγματα τα οποία θεωρούμε ενδιαφέρον να τα μοιραστούμε.

Έχετε άραγε σκεφτεί οι περισσότεροι από εσάς, για ποιόν λόγο εικόνες φόντου δεν θα έπρεπε να χρησιμοποιηθούν για την παρουσίαση κειμένου στην ιστόθεση που κατασκευάζετε; Θα πει κάποιος πως θα μπορούσαμε να φτιάξουμε υψηλής ποιότητας γραφικούς χαρακτήρες χωρίς να χρησιμοποιήσουμε γραφικά; Δεν θα σας απαντήσουμε ακόμα για να σας κρατήσουμε λίγο σε αγωνία, αντί αυτού λοιπόν θα σας θέσουμε μια ερώτηση πόσες ώρες ξοδέψατε για να κάνετε ένα γραφικό χαρακτήρα, ένα τίτλο με ένα σχεδιαστικό πρόγραμμα; Επίσης εάν δεν μπορείτε να το κάνετε μόνοι σας πόσα λεφτά θα πρέπει να δαπανήσετε σε κάποιον γραφίστα, προκειμένου να πετύχετε το γράφημα που επιθυμείτε; Σας αγχώσαμε έτσι δεν είναι; Γιατί έχουμε δίκιο. Σκεφτείτε τώρα να έχετε πολλά περισσότερα γραφικά. Ο χρόνος και το χρήμα που πρέπει να υπενδύσετε θα είναι τεράστιος. Πόσο μάλλον που οι περισσότερες ιστοσελίδες έχουν την ανάγκη να μεταφραστούν σε περισσότερες γλώσσες. Μέχρι το σημείο αυτό είναι σίγουρο που ότι σας αγχώσαμε, και αυτό για να σας δείξουμε την δύναμη και την επανάσταση που φέρνουν τα Fontface. Γεγονός βέβαια είναι ότι η χρήση εικόνων background είναι ο μόνος τρόπος οι σχεδιαστές να μεταφέρουν τις αγαπημένες τους γραμματοσειρές στις ιστοσελίδες τους. Δεν θα ήταν όμως πιο δίκαιο οι σχεδιαστές να έχουν πρόσβαση σε μια μεγάλη ποικιλία fonts, χωρίς να είναι απαραίτητο για να τους υιοθετήσουν να κάνουν γραφικά.



Σχήμα 4.13

Ο τρόπος είναι το fontface, αντί να κάνουν γραφικά μπορούν να επισυνάψουν χαρακτήρες στις ιστοσελίδες τους, έτσι μπορούν να χρησιμοποιούν True type Fonts χωρίς να τα απεικονίζουν με εικόνες.

Web fonts: ο κώδικας που θα λατρέψετε!

Ένα παράδειγμα του τρόπου που χρησιμοποιούνται τα font-face φαίνεται παρακάτω :

```
@font-face {  
  font-family: "Kimberley";  
  src: url (http://www.princexml.com/fonts/larabie/ »  
  kimberle.ttf) format ("truetype");  
}  
h1 { font-family: "Kimberley", sans-serif }
```

Αυτό που σας είναι σίγουρα γνωστό είναι η τελευταία γραμμή, αλλά και το @font-face μπορεί να μην είναι γνωστό, όμως δεν είναι τόσο δύσκολο να κατανοηθεί, αφού θέλουμε να επισυνάψουμε την γραμματοσειρά "Kimberley", αρκεί να ορίσουμε την ακριβή url του. Τα τελευταία χρόνια υπάρχουν κάποιες διαφωνίες σχετικά με την χρήση της τεχνολογίας. Τα επιχειρήματα κατά των γραμματοσειρών ιστοσελίδων είναι κυρίως νομικά. Ότι δηλαδή οι γραμματοσειρές έχουν κάποια πνευματική ιδιοκτησία και ως εκ τούτου δεν μπορούν να δημοσιευτούν στο διαδίκτυο. Παρά το γεγονός ότι το νομικό καθεστώς των σχημάτων γραμματοσειράς είναι αβέβαιο, τα ονόματα των γραμματοσειρών καλύπτονται από το νόμο περί πνευματικών δικαιωμάτων. Έτσι οι γραμματοσειρές έχουν τα ίδια πνευματικά δικαιώματα με τις εικόνες και το κείμενο. Η χρήση των γραμματοσειρών θα πρέπει να γίνεται ύστερα από την χορήγηση άδειας από τον κάτοχο των πνευματικών δικαιωμάτων τους. Ωστόσο υπάρχουν πολλές γραμματοσειρές η χρήση των οποίων είναι ελεύθερη. Ομοίως, πολλοί σχεδιαστές γραμματοσειράς (συμπεριλαμβανομένων Ray Larabie και Dieter Steffmann) επιτρέπουν γραμματοσειρές τους να χρησιμοποιούνται ελεύθερα στο διαδίκτυο.

Ένα πολύ καλό επιχείρημα υπέρ της χρήσης των γραμματοσειρών είναι το αισθητικό αποτέλεσμα που αυτές προσφέρουν στις ιστοσελίδες που χρησιμοποιούνται. Υπάρχουν πολλές όμορφες και ενδιαφέρουσες γραμματοσειρές που μπορούν να βελτιώσουν την αισθητική και την αύξηση της οπτικής ποικιλίας στο διαδίκτυο. Ακριβώς όπως είμαι μερικές φορές έκπληκτοι από όμορφα σχέδια βιβλίων, θέλουμε να κατακλύζομαστε και από όμορφες ιστοσελίδες. Και, δεν θέλουμε αυτές οι όμορφες σελίδες να πρέπει να αποτελούνται από εικόνες φόντου.

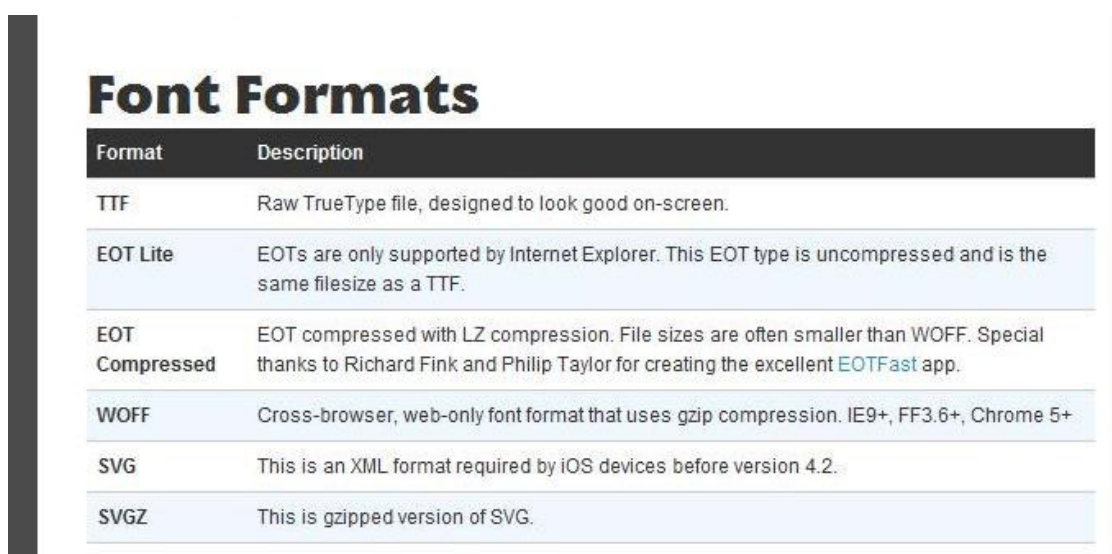
Δύο πράγματα πρέπει να συμβούν πριν οι γραμματοσειρές οδηγήσουν το Web designing στο επόμενο επίπεδο. Πρώτον, πρέπει να εξασφαλιστεί η εγκυρότητα της προσέγγισης. Θα

πρέπει να μπούμε σε μια διαδικασία αξιολόγηση του κώδικα που χρησιμοποιούν τα Fontface, εάν ο κώδικας μας εξυπηρετεί και τι πρέπει να γίνει για να βελτιωθεί. Ο καλύτερος τρόπος για να βοηθήσουμε είναι να παίξουμε με την υπάρχουσα εφαρμογή και να προτείνουμε συνεχώς βελτιώσεις. Δεύτερον, πρέπει να πείσουμε τους ιθύνοντες των προγραμμάτων περιήγησης να προσθέσουν υποστήριξη για γραμματοσειρές Web.

Μία ενδιαφέρουσα ιστοσελίδα η οποία έχει πληθώρα από fonts τα οποία είναι ελεύθερα στο να τα κατεβάσετε και να τα χρησιμοποιήσετε είναι το <http://www.fontsquirrel.com/fontface>. Μέσα σε αυτήν μπορείτε να διαλέξετε από μια πληθώρα γραμματοσειρών και το καλύτερο μπορείτε να κατεβάσετε τον κώδικα για τα fontface καινούριας γενιάς web2.0. Ο κώδικας θα είναι της μορφής που φαίνεται πιο κάτω

```
@font-face {  
  font-family: 'KomikaTitleWideRegular';  
  src: url ('KomikaTitle-Wide-webfont.eot');  
  src: url ('../fonts/KomikaTitle-Wide-webfont.eot?iefix') format ('eot'),  
       url ('../fonts/KomikaTitle-Wide-webfont.woff') format ('woff'),  
       url ('../fonts/KomikaTitle-Wide-webfont.ttf') format ('truetype'),  
       url ('../fonts/KomikaTitle-Wide-webfont.svg#webfontCJDxlCCZ') format ('svg');  
  font-weight: normal;  
  font-style: normal;
```

Όπως παρατηρείτε τον κώδικα βλέπετε ότι έχουμε διάφορα αρχεία με διαφορετικές καταλήψεις, επεξήγηση των οποίων φαίνεται στον πίνακα που ακολουθεί.



Format	Description
TTF	Raw TrueType file, designed to look good on-screen.
EOT Lite	EOTs are only supported by Internet Explorer. This EOT type is uncompressed and is the same filesize as a TTF.
EOT Compressed	EOT compressed with LZ compression. File sizes are often smaller than WOFF. Special thanks to Richard Fink and Philip Taylor for creating the excellent EOTFast app.
WOFF	Cross-browser, web-only font format that uses gzip compression. IE9+, FF3.6+, Chrome 5+
SVG	This is an XML format required by iOS devices before version 4.2.
SVGZ	This is gzipped version of SVG.

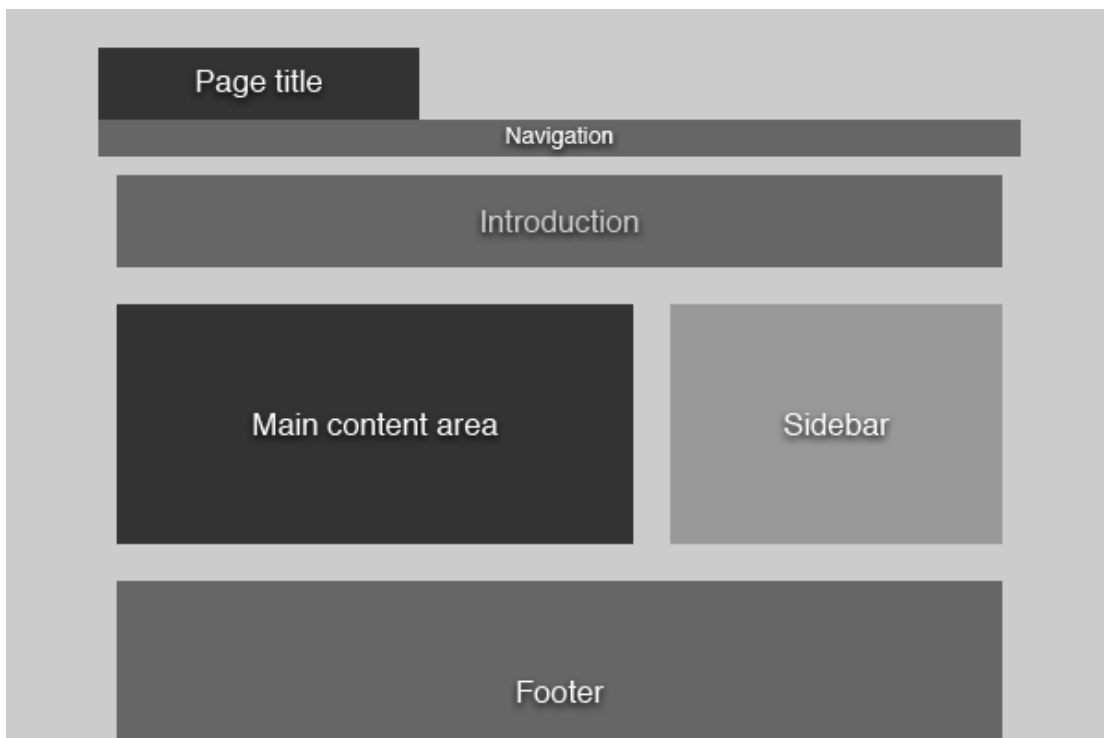
Σχ.4.14 Πίνακας καταλήξεων αρχείων που υποστηρίζει το fontface^[46]

Κατά την εγκατάσταση του κώδικα αυτού στην ιστοσελίδα σας θα συναντήσετε προβλήματα με τον explorer, αυτό οφείλετε ότι η τεχνολογία δεν έχει ακόμα υιοθετηθεί από όλους τους Browser.

4.4 HTML 5

Η HTML 5 είναι η επόμενη σημαντική έκδοση της HTML. Εισάγει μια ποικιλία νέων στοιχείων που θα κάνουν τις σελίδες μας πιο ενδιαφέρουσες. Ως αποτέλεσμα όλων αυτών, είναι πως οι μηχανές αναζήτησης καθώς επίσης και τα προγράμματα ανάγνωσης θα μπορούν πιο εύκολα να περιηγηθούν στις ιστοσελίδες μας. Επιπλέον, η HTML 5 θα περιλαμβάνει φανταχτερά APIs για την κατάρτιση γραφικών στην οθόνη, την αποθήκευση δεδομένων χωρίς σύνδεση, μεταφορά και απόθεση, και πολλά άλλα. Ας ξεκινήσουμε τη σήμανση μέχρι το ιστολόγιο σελίδα.

Για την περιγραφή όλων αυτών θα δημιουργήσουμε ένα παράδειγμα με το οποίο θα γίνουν κατανοητές όλες οι καινούριες τεχνολογίες που φέρνει η HTML 5. Καταρχήν θα πρέπει να ορίσουμε το πώς θα φαίνεται η ιστοσελίδα μας.



Σχήμα 4.15

Η HTML 5 έχει συγκεκριμένα tags για τον ορισμό αντικειμένων όπως header, navigation, sidebar and footer. Αρχικά θα παρουσιάσουμε των κώδικα που πραγματοποιεί το παραπάνω αποτέλεσμα. Θα σας προτείνουμε πρώτου συνεχίσετε την ανάγνωση να μελετήσετε το παρακάτω κώδικα και με την βοήθεια του σχήματος 4.15, προσπαθήστε να βρείτε όλα τα νέα στοιχεία που φέρει η HTML 5.

```
1. <!doctype html>
2. <html>
3. <head>
4.   <title>Page title</title>
5. </head>
6. <body>
7.   <header>
8.     <h1>Page title</h1>
9.   </header>
10.  <nav>
11.    <!-- Navigation -->
12.  </nav>
13.  <section id="intro">
14.    <!-- Introduction -->
15.  </section>
16.  <section>
17.    <!-- Main content area -->
18.  </section>
19.  <aside>
20.    <!-- Sidebar -->
21.  </aside>
22.  <footer>
23.    <!-- Footer -->
24.  </footer>
25.
26. </body>
27. </html>
```

Νομίζουμε ότι σε όλους ο παραπάνω κώδικας φαίνεται ως HTML, δεν είναι ακριβώς όμως έτσι γιατί υπάρχουν αρκετά νέα πράγματα.

- Η HTML 5 έχει ένα συγκεκριμένο DOCTYPE. Αυτό ορίζεται στην πρώτη γραμμή του κώδικα, και λέει στον Browser ότι το έγγραφο που ακολουθεί είναι ένα HTML.
- Η νέα ετικέτα στην γραμμή 7 είναι η <header> βρίσκεται πάντα στην αρχή του εγγράφου και μπορεί να περικλείει στοιχεία όπως είναι ο τίτλος της σελίδας ή ένα λογότυπο. Θα μπορούσε να περιλαμβάνει επίσης έναν πίνακα περιεχομένων ή μια φόρμα αναζήτησης.
- Η ετικέτα nav- χρησιμοποιείται για να περιλάβει τα στοιχεία πλοήγησης, όπως είναι η κύρια πλοήγηση σε μια τοποθεσία ή περισσότερες ειδικευμένες πλοηγήσεις όπως το επόμενο / προηγούμενο.
- Η ετικέτα section, χρησιμοποιείται για να υποδηλώσει μια ενότητα του εγγράφου. Μπορεί να περιέχει όλα τα είδη των εννοιών, όπως και να έχει ένθετα section.
- Το <aside> χρησιμοποιείται για να περιλάβει μέρη του εγγράφου τα οποία αποτελούν μεν στοιχεία του, όμως μπορούν να σταθούν και μόνα τους. Στο συγκεκριμένο παράδειγμα το χρησιμοποιούμε για να περιγράψουμε το sidebar.
- Το υποσέλιδο περιέχει πρόσθετες πληροφορίες σχετικά με το βασικό περιεχόμενο, όπως πληροφορίες σχετικά με το ποιος το έγραψε, πληροφορίες πνευματικών δικαιωμάτων, συνδέσεις με σχετικά έγγραφα και ούτω καθεξής.

Ήδη είναι προφανές, ότι η HTML 5 δεν χρησιμοποιεί <div> για να ορίσει διάφορα τμήματα του εγγράφου. Αντί αυτού χρησιμοποιεί ποιο εξελιγμένα tags τα οποία περιγράφουν πιο περιεκτικά τα διάφορα μέρη του εγγράφου. Ένα επίσης νέο tag που χρησιμοποιεί η HTML 5 είναι το <article>. Η ετικέτα του άρθρου χρησιμοποιείται για να υποδηλώσει ένα ανεξάρτητο τμήμα για παράδειγμα σε ένα blog, συζήτηση, εγκυκλοπαίδεια, κ.λπ.

Όταν η HTML 5 θα εφαρμοστεί σε όλους τους browsers, θα γίνει πολύ πιο εύκολη η κατασκευή ιστοσελίδων. Θα είμαστε σε θέση μην χρησιμοποιούμε floats για τα layout, και θα περνάμε πολύ λιγότερο χρόνο στο να γράφουμε javascript για να τοποθετήσουμε τις εικόνων φόντου ή να κατασκευάσουμε πίνακες με ζέβρα γραμμές. Ας ελπίσουμε ότι θα χρησιμοποιήσουμε όλον αυτόν τον επιπλέον χρόνο για να μελετήσουμε ορισμένες μακροπρόθεσμες παραμελημένες περιοχές του web design, όπως το front end design και την καλή αρχιτεκτονική της πληροφορίας.

4.5 Συμπέρασμα

Οι ιδιότητες του CSS3 μπορούν να βελτιώσουν σημαντικά τη ροή της εργασίας σας, κάνοντας μερικά από τα πιο χρονοβόρα στοιχεία του CSS ελαφρύτερα, επιτρέποντας καλύτερο και καθαρότερο τρόπο κωδικοποίησης. Μερικές ιδιότητες δεν είναι ακόμη ευρέως υποστηριζόμενες, ακόμη και από τους πιο πρόσφατους φυλλομετρητές (browsers), αυτό βέβαια δεν σημαίνει ότι δεν θα πρέπει να πειραματιζόμαστε ώστε να δώσουν στους επισκέπτες μας, σύγχρονα προγράμματα περιήγησης, με προηγμένα χαρακτηριστικά του CSS styling.

5

Ajax

Σκοπός

Σε αυτό το Κεφάλαιο:

- θα γνωρίσουμε την τεχνική Ajax, και
- θα μελετήσουμε διάφορους τρόπους εφαρμογής της.

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να αντιληφθεί τη δυναμική της μεθοδολογίας αυτής,
- να καταφέρει μέσα από την τεχνολογία αυτή να δώσει ζωντάνια στα έργα του, και
- να θεωρήσει το Ajax ως την αρχή και το τέλος κάθε σύγχρονου έργου στο διαδίκτυο.

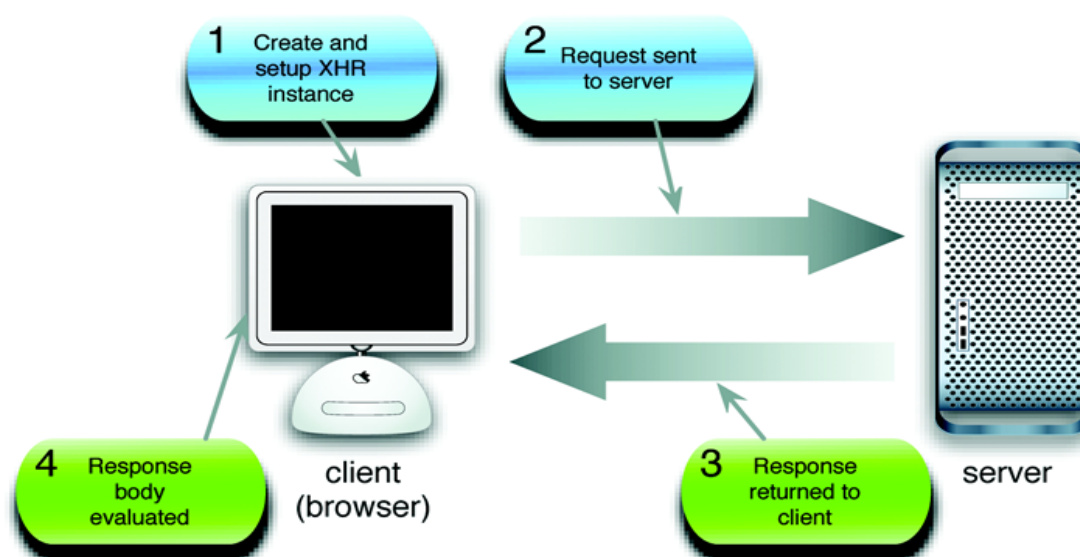
5.1 Παρουσίαση

Το Ajax [9-11] είναι ένα πιασάρικο όνομα για έναν τύπο του προγραμματισμού που έγινε δημοφιλής το 2005 από την Google και άλλους μεγάλους κατασκευαστές ιστοσελίδων. Ajax είναι τα αρχικά για Asynchronous JavaScript και XML [47]. Είναι ένα πραγματικά ευέλικτο εργαλείο το οποίο δίνει την δυνατότητα στον προγραμματιστή να συνδυάζει με τον καλύτερο δυνατό τρόπο όλες τις τεχνικές που παρουσιάσαμε στα προηγούμενα κεφάλαια. Πώς όμως θα μπορούσαμε να περιγράψουμε αυτό το εργαλείο καλύτερα παρά μόνο με ένα παράδειγμα.

5.2 Προετοιμασία

Το Ajax συγχωνεύει μαζί υπάρχουσες τεχνολογίες για να βοηθήσει τους προγραμματιστές να δώσουν στους χρήστες του διαδικτύου μία πιο προηγμένη εμπειρία περιήγησης. Με τη χρησιμοποίηση XHTML [48], CSS3 [19-20], JavaScript [8] και XML, μαζί με το XMLHttpRequest [49], μπορείτε να μετατρέψετε τα προγράμματα περιήγησης σε πλατφόρμες εφαρμογών που αναπαραγάγουν πιστά εφαρμογές γραφείου. Αυτή η δυνατότητα επιτρέπει σε υπάρχουσες ιστοσελίδες να μετατραπούν σε τοποθεσίες Web 2.0, αυξάνοντας ταυτόχρονα τον αριθμό των νέων διαδικτυακών εφαρμογών που μπορείτε να βρείτε στο Διαδίκτυο σήμερα.

Μπορούμε να υποστηρίξουμε επιτυχώς, ότι καμιά τεχνολογία δεν έχει αλλάξει το τοπίο, Ιστού περισσότερο τα τελευταία χρόνια από την υιοθέτηση του Ajax. Η ικανότητα να κάνει ασύγχρονα αιτήματα προς τον διακομιστή χωρίς την ανάγκη να φορτώσετε τις σελίδες, επέτρεψε μια ολόκληρη νέα σειρά αλληλεπίδρασης χρήστη. Στο κεφάλαιο αυτό θα ρίξουμε μια γρήγορη ματιά στην τεχνολογία αυτή, σκοπός μας δεν είναι να αναπτύξουμε ένα πλήρες εγχειρίδιο, αλλά να σας δώσουμε μία πρώτη γεύση.

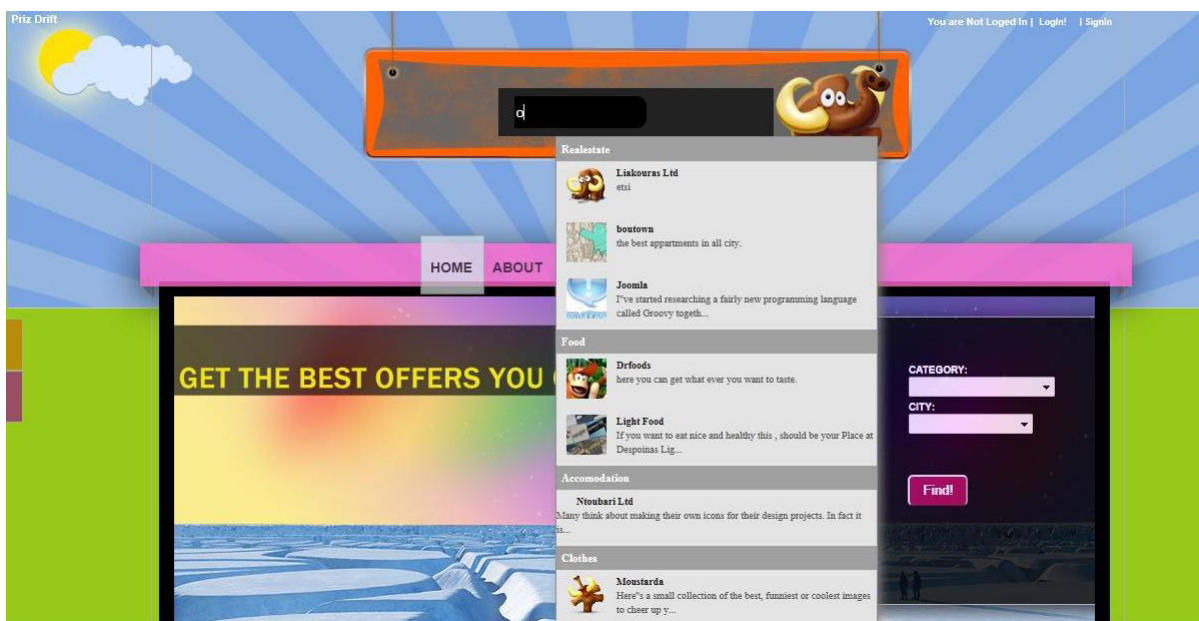


Σχήμα 5.1

Το AJAX είναι απλώς ένα μέσο για τη φόρτωση δεδομένων από τον server στον web browser, ή τον Client, χωρίς ορατή ανανεώσετε τη σελίδα. Αυτά τα δεδομένα μπορούν να πάρουν πολλές μορφές, και έχουμε πολλές επιλογές για το τι πρέπει να κάνουμε με αυτά. Θα κατασκευάσουμε ένα παράδειγμα το οποίο θα παίρνει πληροφορίες και θα τις τοποθετεί σε ομάδες σε σχέση με την κατηγορία που ανήκουν. Το παράδειγμα αυτό είναι μέρος του κώδικα που αναπτύξαμε στα πλαίσια της εργασίας αυτής.

5.2.1 Ποιό είναι το απίστευτο

Το απίστευτο με την τεχνολογία αυτήν, είναι ότι ο μόνος κώδικας που θα βλέπουμε μέσα στην ιστοσελίδα μας θα είναι μια άδεια div, την οποία θα την γεμίζουμε με τις εταιρείες ανά κατηγορία. Το αποτέλεσμα που θα πάρουμε μετά το πέρας τις περιγραφής, θα είναι το εξής.



Σχήμα 5.2

Φαίνεται ξεκάθαρο αυτό που περιγράψαμε, δηλαδή οι εταιρείες κατηγοριοποιημένες ανά υπηρεσία που αυτές προσφέρουν. Πώς όμως θα πετύχουμε το αποτέλεσμα αυτό; Απαντάμε γράφοντας μόνο μια γραμμή κώδικα `<div id="suggestions"></div>`. Βέβαια πίσω από αυτό κρύβεται ένα απίθανο εργαλείο το AJAX.

5.3 Δεδομένα

Πρώτου μπούμε στα βαθιά θα θέλαμε να αναφέρουμε τους τρόπους με τους οποίους η πληροφορία που θέλουμε ανά πάσα στιγμή επικολλάται στο έγγραφο μας, χωρίς να είναι απαραίτητη η επαναφόρτωση της σελίδας. Οι περισσότερες εφαρμογές που κάνουν χρήση της τεχνολογίας αυτής, δεν κάνουν τίποτα άλλο από το να επικολλάν μετά από μια αίτηση δική μας ένα μέγεθος πληροφοριών στο αρχείο μας. Αυτήν η τεχνική συνήθως ονομάζεται ΔΗΑΗ ^[50] (Asynchronous HTTP and HTML). Το πρώτο που μας χρειάζεται είναι λίγος κώδικας HTML, τον οποίο και θα εισάγουμε μέσα στο έγγραφο μας. Ο κώδικας αυτός για το συγκεκριμένο παράδειγμα θα τοποθετηθεί σε ένα αρχείο με το όνομα a.html. ο κώδικας που θα έχει το αρχείο αυτό για τον σκοπό του παραδείγματος θα είναι:

```
<div class="entry">
<h3 class="term">ABDICATION</h3>
<div class="part">n.</div>
<div class="definition">
An act whereby a sovereign attests his sense of the high temperature of the throne.
<div class="quote">
<div class="quote-line">Poor Isabella's Dead, whose abdication</div>
<div class="quote-line">Set all tongues wagging in the Spanish nation.</div>
<div class="quote-line">For that performance 'tweunfair to scold her:</div>
<div class="quote-line">She wisely left a throne too hot to hold her.</div>
<div class="quote-line">To History she'll be no royal riddle &mdash;</div>
<div class="quote-line">Merely a plain parched pea that jumped thegriddle.</div>
<div class="quote-author">G.J.</div>
</div>
</div>
</div>
<div class="entry">
<h3 class="term">ABSOLUTE</h3>
<div class="part">adj.</div>
<div class="definition">
Independent, irresponsible. An absolute monarchy is one
in which the sovereign does as he pleases so long as he
pleases the assassins. Not many absolute monarchies are
```

left, most of them having been replaced by limited monarchies, where the sovereign's power for evil (and for good) is greatly curtailed, and by republics, which are governed by chance.

</div>

</div>

Εάν παρατηρήσετε καλά τον κώδικα θα διαπιστώσετε ότι δεν μιλάμε για αληθινό κώδικα HTML, αφού δεν περιλαμβάνει tags όπως <html>, <head>, και <body> τα οποία είναι απαραίτητα συστατικά κάθε αρχείου HTML. Ο λόγος που χρησιμοποιούμε τέτοιου είδους αρχείων είναι για να τα επικολλήσουμε σε ένα σημείο του κώδικα μας, τον εξής τρόπο:

```
$ (document).ready (function () {  
    $('#letter-a a').click (function ()  
    {  
        $('#dictionary').load ('a.html');  
        return false;  
    });  
});
```

Τι μας λέει ο κώδικας αυτός; Καταρχάς μέσα στην ready ορίζουμε ένα συμβάν, με το οποίο όταν ο χρήστης κάνει επιλογή του a με ταυτότητα #letter-a, εκτελείται μία συνάρτηση η οποία μέσα στο <div id='dictionary'> φορτώνει το περιεχόμενο του a.html. Όλη αυτή την δουλειά την κάνει η μέθοδος load ('a.html').

Βλέπουμε πόσο εύκολο είναι να παίρνουμε και να τοποθετούμε όπου εμείς επιθυμούμε κώδικα HTML. Πώς όμως θα μπορούσαμε τα δεδομένα πριν τα τοποθετήσουμε να τα επεξεργαστούμε;

5.3.1 Επεξεργασία πριν την προβολή

JSON

Ο όρος JavaScript Object Notation [51] (JSON) επινοήθηκε από τον Douglas Crockford, προκειμένου να χαρακτηρίσει την απλή αυτή μορφή σύνταξης που ακολουθεί. Μπορούμε να τοποθετήσουμε τις πληροφορίες μας μέσα σε ένα αρχείο τύπου b.json, με διαφορετικούς

τρόπους. Ενδεικτικά θα τοποθετήσουμε κάποιες πληροφορίες οι οποίες θα έχουν την εξής μορφή.

```
[
{
"term": "BACCHUS",
"part": "n.",
"definition": "A convenient deity invented by the...",
"quote": [
"Is public worship, then, a sin,",
"That for devotions paid to Bacchus",
"The lictors dare to run us in,",
"And resolutely thump and whack us?"
],
"author": "Jorace"
},
{
"term": "BACKBITE",
"part": "v.t.",
"definition": "To speak of a man as you find him when..."
},
{
"term": "BEARD",
"part": "n.",
"definition": "The hair that is commonly cut off by..."
},
}
```

Για να μπορέσουμε να πάρουμε την πληροφορία αυτήν μέσα από το αρχείο θα κάνουμε χρήση της μεθόδου `$.getJSON()`. Η μέθοδος αυτή είναι λίγο διαφορετική από τις μεθόδους `jQuery`, τις οποίες αναπτύξαμε εκτεταμένα σε προηγούμενα κεφάλαια. Η `$.getJSON()` δεν υπάρχει ένα λογικό αντικείμενο `DOM`, πάνω στο οποίο εφαρμόζεται. Τα αποτελέσματα της θα τοποθετηθούν μέσα στο `script` το ίδιο, και όχι σε κάποιο σημείο της σελίδας μας.

Για να κάνουμε χρήση της μεθόδου αυτής γράφουμε όπως και προηγουμένως τον εξής κώδικα:

```

$(document).ready(function () {
    $('#letter-b a').click(function () {
        $.getJSON('b.json');
        return false;
    });
});

```

Ο κώδικας αυτός σε αντίθεση με προηγουμένως δεν θα δημιουργήσει κάποιο συμβάν κάνοντας κλικ στο αντίστοιχο δεσμό. Η συνάρτηση φορτώνει τα δεδομένα, όμως δεν της έχουμε πει τι να κάνει με τα δεδομένα αυτά. Για τον λόγο αυτό και για να υπάρχει τρόπος να πούμε στην `$.getJSON`, τι να κάνει με τα δεδομένα αυτά της δίνουμε ως δεύτερο όρισμα μια συνάρτηση, η οποία εκτελείται όταν τα δεδομένα έχουν φορτωθεί επιτυχώς. Όπως έχει αναφερθεί όλες οι λειτουργίες που κάνει ο AJAX είναι ασύγχρονες, έτσι παρέχεται ένας τρόπος να περιμένει η συνάρτηση μέχρι να φορτωθούν όλα τα δεδομένα, αντί να αρχίσει να εκτελείτε κατευθείαν. Επιπλέον η συνάρτηση που χρησιμοποιείται ως δεύτερο όρισμα παίρνει ως όρισμα μια μεταβλητή στην οποία αποθηκεύει την πληροφορία.

```

$(document).ready(function () {
    $('#letter-b a').click(function () {
        $.getJSON('b.json', function (data) { });
        return false;
    });
});

```

Για να μπορέσουμε να επεξεργαστούμε την πληροφορία που θα αποθηκευτεί στην μεταβλητή `data`, χρειαζόμαστε έναν τρόπο να την διαβάσουμε. Στο σημείο αυτό μπαίνει η μέθοδος `$.each()`, η οποία παίρνει ως πρώτο όρισμα έναν πίνακα δεδομένων και σαν δεύτερο όρισμα μία συνάρτηση. Η `$.each()` εκτελεί ένα loop και κάθε φορά στον δεύτερο όρισμα της περνάει, ως πρώτο όρισμα τον δείκτη και σαν δεύτερο όρισμα περνάει την τιμή που απεικονίζει ο δείκτης αυτός στον πίνακα δεδομένων.

```

$(document).ready(function () {
    $('#letter-b a').click(function () {
        $.getJSON('b.json', function (data) {
            $('#dictionary').empty ();
            $.each (data, function (entryIndex, entry) {
                var html = '<div class="entry">';
                html += '<h3 class="term">' + entry['term'] + '</h3>';
                html += '<div class="part">' + entry['part'] + '</div>';
                html += '<div class="definition">';
                html += entry['definition'];
                html += '</div>';
                html += '</div>';
            });
            $('#dictionary').append (html);
        });
    });
});
return false;
});

```

Εκτέλεση script

Ένας άλλος τρόπος για να τοποθετούμε την πληροφορία μέσα στην σελίδα μας είναι να εκτελούμε κάποιο script. Πολλές φορές δεν είμαστε σε θέση να γνωρίζουμε τι θέλουμε να τοποθετήσουμε μέσα στην σελίδα πριν γίνει κάποια επιλογή από τον χρήστη. Για το λόγο αυτό δημιουργούμε scripts τα οποία τα εκτελούμε κάθε φορά που τα έχουμε ανάγκη.

Η εκτέλεση ενός script είναι τόσο εύκολο όσο και η φόρτωση ενός αρχείου HTML, μέσα στην ιστοσελίδα μας. Απλά η διαφορά είναι ότι στην περίπτωση αυτήν χρησιμοποιούμε την καθολική συνάρτηση \$.getScript ().

```

$(document).ready(function () {
    $('#letter-c a').click(function () {
        $.getScript ('c.js');
        return false;
    });
});

```

Το αρχείο πλέον που θα επεξεργαστούμε θα έχει την μορφή

```
var entries = [  
  {  
    "term": "CALAMITY",  
    "part": "n.",  
    "definition": "A more than commonly plain and..."  
  },  
  {  
    "term": "CANNIBAL",  
    "part": "n.",  
    "definition": "A gastronome of the old school who..."  
  },  
  {  
    "term": "CHILDHOOD",  
    "part": "n.",  
    "definition": "The period of human life intermediate..."  
  },  
  {  
    "term": "CLARIONET",  
    "part": "n.",  
    "definition": "An instrument of torture operated by..."  
  },  
  {  
    "term": "COMFORT",  
    "part": "n.",  
    "definition": "A state of mind produced by..."  
  },  
  {  
    "term": "CORSAIR",  
    "part": "n.",  
    "definition": "A politician of the seas."  
  }  
];
```

```
var html = "";
$.each (entries, function () {
html += '<div class="entry">';
html += '<h3 class="term">' + this['term'] + '</h3>';
html += '<div class="part">' + this['part'] + '</div>';
html += '<div class="definition">' + this['definition'] + '</div>';
html += '</div>';
});
$('#dictionary').html (html);
```

Το όνομα του θα είναι c.js.

XML

Το XML είναι μέρος των αρχικών του Ajax, και αποτελεί άλλη μια μέθοδο αποθήκευσης πληροφοριών, η οποία μιμείται των JSON. Έστω ότι έχουμε το αρχείο με δεδομένα της μορφής xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<entries>
<entry term="DEFAME" part="v.t.">
<definition>
To lie about another. To tell the truth about another.
</definition>
</entry>
<entry term="DEFENCELESS" part="adj.">
<definition>
Unable to attack.
</definition>
</entry>
<entry term="DELUSION" part="n.">
<definition>
The father of a most respectable family, comprising
Enthusiasm, Affection, Self-denial, Faith, Hope,
Charity and many other goodly sons and daughters.
```

```

</definition>
<quote author="Mumfrey Mappel">
<line>All hail, Delusion! Were it not for thee</line>
<line>The world turned topsy-turvy we should see;
</line>
<line>For Vice, respectable with cleanly fancies,
</line>
<line>Would fly abandoned Virtue's gross advances.
</line>
</quote>
</entry>

```

```

<entry term="DIE" part="n.">
<definition>
The singular of "dice." We seldom hear the word,
because there is a prohibitory proverb, "Never say
die." At long intervals, however, some one says: "The
die is cast," which is not true, for it is cut. The
word is found in an immortal couplet by that eminent
poet and domestic economist, Senator Depew:

```

```

</definition>
<quote>
<line>A cube of cheese no larger than a die</line>
<line>May bait the trap to catch a nibbling mie.</line>
</quote>
</entry>
</entries>

```

Για να μπορέσουμε όπως και προηγουμένως να πάρουμε και να επεξεργαστούμε τις πληροφορίες του αρχείου, χρησιμοποιούμε τον παρακάτω κώδικα.

```

$(document).ready (function () {
    $('#letter-d a').click (function () {
        $.get ('d.xml', function (data) {
            });
        return false;
    });

```

});

});

Αυτήν την φορά η μέθοδος που κάνει όλη την δουλειά είναι η `$.get`. Με λίγα λόγια η συνάρτηση αρπάζει τον κώδικα που βρίσκετε στην συγκεκριμένη τοποθεσία και τα δεδομένα του τα περνάει στη συνάρτηση που φέρει ως δεύτερο όρισμα.

5.4 Επιλογή Δεδομένων

Παρουσιάσαμε τέσσερις τρόπους παρουσίασης των δεδομένων προς επεξεργασία, ποιος όμως από αυτούς είναι και ο καλύτερος;

HTML απαιτεί πολύ μικρό κόπο για να κατασκευαστεί. Τα δεδομένα μπορούν να φορτωθούν σε όποιο σημείο της ιστοσελίδας μας με μία απλή μόνο μέθοδο. Από τη άλλη τα δεδομένα δεν είναι έτσι δομημένα ώστε να είναι χρήσιμα και για άλλες εφαρμογές.

JSON files είναι με τέτοιο τρόπο δομημένα που να επιτρέπουν την απλή επαναχρησιμοποίηση τους. Οι πληροφορίες που εσωκλείουν είναι πολύ απλά δομημένες και η ανάγνωση τους μπορεί να γίνει με πολύ μεγάλη ταχύτητα. Ωστόσο σφάλματα μέσα στα αρχεία αυτά μπορούν να προκαλέσουν σφάλματα και μέσα στην ιστοσελίδα.

JavaScript files είναι υπερβολικά ευέλικτα, όμως δεν αποτελούν μηχανισμούς αποθήκευσης δεδομένων. Επειδή είναι γραμμένα σε μια συγκεκριμένη γλώσσα δεν μπορούν να χρησιμοποιηθούν για να παρέχουν πληροφορίες σε διαφορετικές εφαρμογές. Η δυνατότητα ωστόσο που προσφέρουν να μπορούμε να φορτώνουμε κατεπιλογή κώδικα στην σελίδα μου τα κάνει χρήσιμα.

XML documents είναι οι Βασιλιάδες της συμβατότητας. Τα XML για το διαδίκτυο αποτελούν την *lingua franca*, απεικονίζοντας τα δεδομένα με τον τρόπο αυτό είναι πιθανόν να επαναχρησιμοποιηθούν. Το μοναδικό μειονέκτημα είναι ότι είναι πιο αργή εάν θελήσουμε να τροποποιήσουμε τα δεδομένα ή να τα ανακτήσουμε, από άλλες μεθόδους.

Με βάση όλα αυτά τα χαρακτηριστικά μπορούμε εύκολα να συμπεράνουμε ότι είναι καλύτερα εάν δεν θα χρησιμοποιηθούν τα δεδομένα πουθενά αλλού να χρησιμοποιούμε HTML. Σε περίπτωση που τα δεδομένα θα επαναχρησιμοποιηθούν και μπορεί επίσης να επηρεαστούν από την εφαρμογή που θα τα χρησιμοποιήσει καλό είναι να κάνουμε χρήση του

JSON. Εάν η απομακρυσμένη εφαρμογή που θα χρησιμοποιήσει τα δεδομένα δεν είναι γνωστή, για μεγαλύτερη ασφάλεια καλό είναι να κάνουμε χρήση της XML.

5.4 Ajax στην Πράξη.

Δεν θα μπορούσαμε να βρούμε καλύτερο τρόπο για να σας πείσουμε για την δυναμικότητα αυτού του συγκεκριμένου τρόπου προγραμματισμού, από το να περιγράψουμε μία εφαρμογή πραγματικού κόσμου. Όπως και προλογήθηκε στην αρχή αυτού του κεφαλαίου, σκοπός μας είναι η δημιουργία ενός πεδίου αναζήτησης μέσα στην ιστοσελίδα το οποίο με κάθε κλικ του χρήστη θα φιλτράρει την βάση και θα παρουσιάζει κάποια προτεινόμενα αποτελέσματα σχ.5.2. παρόμοια προγράμματα συναντάμε σε πολλούς σύγχρονους ιστότοπους όπως το Google, facebook, yahoo και πάρα πολλούς άλλους.

Μέσα από την διαδικασία αυτήν θα διαπιστώσετε κάτι στο οποίο αναφερόμαστε από την αρχή των σημειώσεων αυτών και δεν είναι τίποτε άλλο από την δυναμική σύντηξη, όλων των τεχνολογιών που περιγράψαμε κάτω από την εποπτεία του Ajax.

HTML

Καταρχάς θα ξεκινήσουμε με τον κώδικα που θα βάλουμε στην ιστοσελίδα μας για να δημιουργήσουμε ένα πεδίο μέσα στο οποίο θα πληκτρολογούμε αυτό που θέλουμε να αναζητήσουμε. Αυτό το πετυχαίνουμε με απλό κώδικα HTML που φαίνεται παρακάτω:

```
<form id="searchforma">
<div> What are you looking for? <input type="text" size="30" value="" id="inputString"
onkeyup="lookup (this.value);"/></div>
<div id="suggestions"></div><!--it will be the divider in which we will place the actual found
results-->
</form>
```

Δεν νομίζουμε να υπάρχει κανένας που δεν καταλαβαίνει τι κάνει αυτός ο κώδικας, ωστόσο οφείλουμε να εξηγήσουμε ότι το πεδίο μέσα στο οποίο πληκτρολογούμε αυτό που αναζητάμε, απεικονίζεται ως:

```
<input type="text" size="30" value="" id="inputString" onkeyup="lookup (this.value);">
```

Το σημείο που κάνει τη διαφορά είναι ότι κάθε φορά που αλλάζει η κατάσταση του καλείτε η συνάρτηση `onkeyup="lookup (this.value)`. Η `lookup (this.value)` παίρνει την τιμή του πεδίου της φόρμας και κάνει μια λειτουργία στην οποία θα αναφερθούμε στην συνέχεια. Όπως αναφερθήκαμε, τα αποτελέσματα θα εκτυπωθούν μέσα στην :

```
<div id="suggestions"></div><!--it will be the divider in which we will place the actual found results-->
```

Θα μπορούσε κάποιος να πει, πως θα γίνει δεν θα πρέπει να δημιουργηθεί κάποιος συγκεκριμένος τρόπος κατάταξης των αποτελεσμάτων και κάποια συγκεκριμένη μορφοποίηση;

Η απάντηση σε ατά τα ερωτήματα είναι πως ο κώδικας γενάτε κατά την δημιουργία του όσο για την μορφοποίηση αυτήν θα γίνει με την τεχνική που έχει ήδη περιγραφεί και είναι το CSS3.

CSS3

Είναι η επόμενη τεχνική που θα χρησιμοποιήσουμε προκειμένου να μορφοποιήσουμε τα αποτελέσματα της αναζήτησης.

```
#suggestions {
    position: relative;
    left: 195px;
    width: 320px;
    display: none;
    z-index: 9997;
}
/* SEARCHRESULTS */
#searchresults {
    border-width: 1px;
    border-color: #919191;
    border-style: solid;
    width: 320px;
    background-color: #a0a0a0;
    font-size: 10px;
    line-height: 14px;
    z-index: 9997;
}
#searchresults a { display: block; background-color: #e4e4e4; clear: left; height: 56px; text-decoration: none; }
#searchresults a: hover { background-color: #b7b7b7; color: #ffffff; }
#searchresults a img {
    float: left;
    padding: 5px 10px;
```

```

border-top-style: none;
border-right-style: none;
border-bottom-style: none;
border-left-style: none;
max-height:40px;
}
#searchresults a span.searchheading { display:block; font-weight:bold; padding-top:5px;
color:#191919; }
#searchresults a:hover span.searchheading { color:#ffffff; }
#searchresults a span { color:#555555; }
#searchresults a:hover span { color:#f1f1f1; }
#searchresults span.categorysea { font-size:11px; margin:5px; display:block; color:#ffffff; }
#searchresults span.seperator {
float:right;
padding-right:15px;
margin-right:5px;
background-image:url (../images/shortcuts_arrow.gif);
background-repeat:no-repeat;
background-position:right;
}
#searchresults span.seperator a {
background-color:transparent;
display:block;
margin:1px;
height:auto;
color:#ffffff;
cursor: crosshair;
}

```

Ο παραπάνω κώδικας πιστεύουμε ότι δεν παρουσιάζει κάποια ιδιαίτερη δυσκολία στην κατανόηση του, και ενδείκνυται για μελέτη και εφαρμογή σε παρόμοιες περιπτώσεις. Θα σας προτείναμε να προσπαθήσετε να τον εφαρμόσετε σε δικές σας ιστοσελίδες και να προσπαθήσετε να τον τροποποιήσετε, γιατί νομίζουμε έτσι θα μπορέσετε να τον καταλάβετε καλύτερα.

Ajax

Είναι το μαγικό σημείο το οποίο περιμέναμε σε όλο την εργασία και αυτό που θα δώσει ζωή στην στατική φόρμα, ο κώδικας της οποίας δόθηκε παραπάνω. Το σημείο όμως που θα αφυπνίσει την φόρμα και θα την ζωντανέψει βρίσκεται στην πρόταση `onkeyup="lookup (this.value)`. Τι σημαίνει αυτό όμως; Με απλά Ελληνικά μας λέει ότι κάθε φορά που μεταβάλλετε η κατάσταση του πλαισίου μέσα στο οποίο κάνω την πληκτρολόγηση, κατά την διαδικασία της αναζήτησης,

καλείτε η συνάρτηση `lookup ()`, με τιμή ως όρισμα την τρέχουσα τιμή που έχει το πλαίσιο την ώρα που την καλούμε. Ο κώδικας της συνάρτησης `lookup ()` φαίνεται παρακάτω.

```
function lookup (inputString) {
    if (inputString.length == 0) {
        $('#suggestions').fadeOut (); // Hide the suggestions box
    } else {
        $.post ("do-thesearch.php", {queryString: ""+inputString+""}, function (data)
{ // Do an AJAX call
            $('#suggestions').fadeIn (); // Show the suggestions box
            $('#suggestions').html (data); // Fill the suggestions box
        });
    }
}
```

Η πρώτη γραμμή του κώδικα δείχνει τον τρόπο που ορίζεται μια συνάρτηση, ο οποίος είναι σε όλους μας γνωστός. Μέσα στο σώμα της συνάρτησης ξεκινάμε με μία συνθήκη, η οποία ελέγχει εάν τα δεδομένα τα οποία αποστέλλονται κατά την κλήση της, μέσω της μεταβλητής «`inputString`», τα οποία να μην ξεχνάμε ότι είναι της μορφής κειμένου, έχουν κάποια τιμή. Με ποιόν τρόπο γίνεται ο έλεγχος αυτός μέσω της μεθόδου `length`, η οποία στην μεταβλητή στην οποία εφαρμόζεται υπολογίζει το μήκος της και επιστρέφει μία τιμή που δεν είναι άλλη από το μήκος των χαρακτήρων από τους οποίους αποτελείται. Με τον τρόπο αυτό γίνεται έλεγχος εάν το πλαίσιο αναζήτησης έχει κάποια τιμή, εάν δεν έχει το `<div id="suggestions"></div>` το κάνει `fadeOut`, δηλαδή το κρύβει. Σε περίπτωση που υπάρχει κάποια τιμή καλείται η μέθοδος `$.post ()` που αποτελείται από τρία ορίσματα, η οποία είναι μια κλήση μεθόδου Ajax που αποστέλλει το περιεχόμενο της μεταβλητής `inputString` (όρισμα 2), σε ένα αρχείο `do-thesearch.php` (όρισμα 1). Το συγκεκριμένο αρχείο εκτελεί όλες τις διαδικασίες επικοινωνίας με τον server και θα παρουσιαστεί στην συνέχεια.

Το μεγαλύτερο ενδιαφέρον είναι όμως αφού γίνει η μεταφορά των δεδομένων στην `do-thesearch.php`, η συνάρτηση `$.post`, εκτελεί μια συνάρτηση (όρισμα 3) την

```
function (data) { // Do an AJAX call
    $('#suggestions').fadeIn (); // Show the suggestions box
    $('#suggestions').html (data); // Fill the suggestions box
}
```

Τι κάνει η συνάρτηση αυτήν όμως; Όπως παρατηρούμε μέσα στον ορισμό της περιλαμβάνει και μια μεταβλητή την `data`, γιατί υπάρχει η μεταβλητή όμως αυτήν, στο σημείο αυτό είναι η δυναμικότητα που προσφέρει το Ajax. Μέσω της συνάρτησης αυτής θα πάρουμε όλα τα δεδομένα τα οποία θα παράγει η κλήση της `do-thesearch.php` και θα τα αποθηκεύσουμε στην μεταβλητή `data`. Σκεφτείτε για λίγο τι σας είπαμε μόλις τώρα. Ναι καλά καταλάβατε, μέσω της `do-thesearch.php` μπορούμε να δημιουργήσουμε κώδικα και μέσω της μεταβλητής `data`, να τοποθετήσουμε τον κώδικα που μόλις παραγάγαμε σε όποιο σημείο της σελίδας μου θέλουμε. Στην συγκεκριμένη περίπτωση το τοποθετούμε μέσα στην `#suggestions` και πως το κάνουμε αυτό μέσω της εντολής σε JQuery `$('#suggestions').html (data)`. Η μέθοδος `html`, αντικαθιστά τα περιεχόμενα της `suggestions` με αυτά που φέρει η `data`. Προσοχή χρειάζεται στον χειρισμό της μεθόδου αυτής, γιατί εάν υπάρχει οτιδήποτε άλλο μέσα στο χώρο που αυτή εφαρμόζεται αντικαθιστάται από το περιεχόμενο της `data`. Βέβαια όσοι από εσάς μελετήσατε τον CSS κώδικα που συνοδεύει την εφαρμογή αυτήν και δεν τον ξεπεράσατε έτσι απλά, θα παρατηρήσατε ότι η `#suggestions`, είναι αρχικά από ορισμού της εξαφανισμένη, και για να την δούμε θα πρέπει να την εμφανίσουμε. Αυτό το πετυχαίνουμε με την εντολή `$('#suggestions').fadeIn ()`. Η μέθοδος `fadeIn ()` εμφανίζει το αντικείμενο στο οποίο εφαρμόζεται, στην προκειμένη περίπτωση το `suggestions`.

Ήδη από τα παραπάνω διαπιστώνουμε τον συνδυασμό κάτω από ένα δυναμικό εργαλείο τριών μεθοδολογιών προγραμματισμού, όπως το JQuery, CSS, και PHP με αποτέλεσμα την παραγωγή κώδικα που προηγουμένως δεν υπήρχε και την επεξεργασία του κώδικα αυτού στην μεριά του Client.

PHP

Στο παράδειγμα μας γίνεται αναφορά σε ένα αρχείο με το όνομα `do-thesearch.php`, το οποίο θα τολμούσαμε να το περιγράψουμε ως καταδρομέα του κώδικα που δημιουργούμε γιατί κάνει όλη την επίπονη δουλειά χωρίς να είναι ο πρωταγωνιστής. Και τι σημαίνει αυτό, παρατηρήστε τον κώδικα που ακολουθεί.

```
<p id="searchresults">
<?php
// Initialize session
session_id ();
session_start ();
header ('Cache-control: private'); // IE 6 FIX
$hostname="localhost";
$databse="final_thesis";
```

```

$username="ilias";
$password="iliast24";
    // PHP5 Implementation - uses MySQLi.
    // mysqli ('localhost', 'yourUsername', 'yourPassword', 'yourDatabase');
    $db = new mysqli ('localhost', 'ilias', 'iliast24', 'final_thesis');

    if (!$db) {
        // Show error if we cannot connect.
        echo 'ERROR: Could not connect to the database.';
    } else {
        // Is there a posted query string?
        if (isset ($_POST['queryString'])) {
            $queryString = $db->real_escape_string ($_POST['queryString']);

            // Is the string length greater than 0?
            if (strlen ($queryString)>0) {
                $query = $db->query ("SELECT * FROM company INNER JOIN
categories ON CAT_ID = CID WHERE COMPNAME LIKE '%". $queryString. "' ORDER BY
CAT_ID LIMIT 8");

                if ($query) {
                    // While there are results loop through them - fetching an Object.
                    // Store the category id
                    $catid = 0;
                    while ($result = $query ->fetch_object ()) {
                        if ($result->CAT_ID!= $catid) { // check if the category changed
                            echo '<span class="categorysea">'. $result->CAT_NAME. '</span>';
                            $catid = $result->CAT_ID;
                        }
                        $email = $result->EMAIL;
                        $logo = $result->LOGO;
                        echo "<a href=\"#\" onclick=\"lookup1 ('$email');\>";
                        echo '';
                        //'. $result->LOGO.'
                        $name = $result->COMPNAME;
                        if (strlen ($name) > 35) {

```

```

        $name = substr ($name, 0, 35). "..."; //EAN EINAI
MEGALYTERO TO KANW FETCH K BAZW TELEIES
    }
    echo '<span class="searchheading">'.$name.'</span>';

    $description = $result->DESCRIPT;
    if (strlen ($description) > 80) {
        $description = substr ($description, 0, 80). "...";
    }

    echo '<span>'.$description.'</span></a>';

}

echo '<span class="seperator"><a href="http://www.google.com" title="Sitemap">Nothing
interesting here? Try the Google.</a></span><br class="break" />';
    } else {
        echo 'ERROR: There was a problem with the query.';
    }
} else {
    // Dont do anything.
} // There is a queryString.
} else {
    echo 'There should be no direct access to this script!';
}
}
?>
</p>

```

Θα προσπαθήσουμε με λίγα λόγια να αναλύσουμε τον κώδικα αυτό για να διαπιστώσετε την δυναμικότητα που τον συνοδεύει.

Όλος ο κώδικας που θα παραχθεί περικλείεται μέσα σε μια παράγραφο με id searchresults. Στην αρχή ξεκινάμε δημιουργώντας μία session και αρχικοποιούμε ορισμένες μεταβλητές οι οποίες θα χρησιμοποιηθούν για να συνδεθούμε στην βάση μας.

```

// Initialize session
session_id ();

```

```

session_start ();
header ('Cache-control: private'); // IE 6 FIX
$hostname="localhost";
$dbname="final_thesis";
$username="ilias";
$password="iliast24";
    
```

Αφού αρχικοποιήσαμε τις μεταβλητές μου θα δημιουργήσουμε ένα αντικείμενο της βάσης δεδομένων μας

```
$db = new mysqli ('localhost', 'ilias', 'iliast24', 'final_thesis');
```

Το αντικείμενο αυτό θα το ονομάσουμε \$db. Πάνω σε αυτό θα μπορέσουμε να εκτελέσουμε τα διάφορα Queries για να πάρουμε τα δεδομένα που θέλουμε.

Η εντολή isset (\$ _POST['queryString']) ελέγχει εάν η php έχει δεχθεί ένα post με τιμή queryString. Θα εκτελεστεί εάν είναι αληθές διαφορετικά δεν θα εκτελείτε.

Στην περίπτωση μας θέλουμε να πάρουμε τα στοιχεία των εταιρειών που κάνουν τις προσφορές για το λόγο αυτόν θα γράψουμε

```
$query = $db->query ("SELECT * FROM company INNER JOIN categories ON CAT_ID = CID WHERE COMPNAME LIKE '%'. $queryString. "' ORDER BY CAT_ID LIMIT 8");
```

Το \$db->query είναι το σημείο μέσα στο οποίο εκτελείτε το ζητούμενο query. Στην συγκεκριμένη περίπτωση ζητάμε να γίνει συνένωση δύο πινάκων του company και του categories. Ένα άλλο σημείο το οποίο παρουσιάζει ενδιαφέρον είναι \$result = \$query ->fetch_object (). Στην περιγραφή της προηγούμενης εντολής η εκτέλεση του query είχε ως αποτέλεσμα την αποθήκευση των αποτελεσμάτων στην \$query, Αυτή η γραμμή δεν είναι πίνακας αλλά resource! Έτσι, δεν μπορούμε να διαχειριστούμε τη μεταβλητή \$ query ως πίνακα, αφού είναι τύπου resource. Με την εντολή fetch_object (), παίρνει την πρώτη γραμμή του resource και την μετατρέπει σε πίνακα. Στην συνέχεια αποθηκεύουμε τον πίνακα αυτόν στην μεταβλητή \$result.

Η εντολή επίσης if (strlen (\$name) > 35) { \$name = substr (\$name, 0, 35). "...";} ελέγχει εάν το όνομα είναι μεγαλύτερο του 35, σε περίπτωση που αυτό συμβαίνει εκτυπώνει μόνο τους 35 πρώτους χαρακτήρες και στην συνέχεια βάζει τρεις τελείες.

Όλος αυτός ο κώδικας αναλύθηκε για να μας δείξει ότι η κύρια σελίδα της εφαρμογής είναι στατική, και μέσω του αρχείου php, αποκτά δυναμισμό και παράγει κώδικα ο οποίος προηγουμένως δεν υπήρχε.

Από τα παραπάνω διαπιστώνουμε την ευελιξία που μας προσφέρει το εργαλείο αυτό στο να συνδυάζουμε διάφορες τεχνολογίες μεταξύ τους. Θα μπορούσαμε να κλείσουμε το κεφάλαιο στο σημείο αυτό, σκεφτήκαμε όμως να διαθέσουμε λίγες ακόμα σελίδες για την περιγραφή μιας τεχνικής η οποία ονομάζεται σελιδοποίηση ή Pagination.

5.5 Σελιδοποίηση

Όλοι μας έχουμε δει σε μεγάλα site όπως είναι το Google, το yahoo, το eBay οπου ζητάμε μεγάλες ποσότητες πληροφοριών οι οποίες συνήθως μας παρουσιάζονται με την μορφή ενός πίνακα. Τι θα συμβεί εάν κατά την αναζήτηση δεν επιλέξουμε κάτι συγκεκριμένο και απλά πιέσουμε το κουμπί αναζήτησης; Στην προκειμένη περίπτωση θα μεταφερθούν όλες οι πληροφορίες από το server στην οθόνη του υπολογιστή μας. Και θα απαντούσε κανείς και που είναι το παράξενο. Δεν υπάρχει τίποτα παράξενο αρκεί η διαχείριση τους να γίνει με τέτοιο τρόπο ώστε αφενός μεν να μην έχουμε πτώση του συστήματος λόγω υπερφόρτωσης, και αφετέρου η σελίδα στην οποία θα προβληθούν οι πληροφορίες να τις παρουσιάσει με τρόπο φιλικό προς τον χρήστη.

Οι τεχνικές που μπορούν να εφαρμοστούν για να πάρουμε τις πληροφορίες είναι δύο ειδών. Η μία είναι να πάρουμε όλη την πληροφορία από τον server ανεξαρτήτως μεγέθους και η άλλη είναι να χρησιμοποιήσουμε τον Ajax για να πάρουμε ένα μέρος του πίνακα πληροφοριών. Η δεύτερη η οποία είναι και η πιο δυναμική, είναι ανεξάρτητη του Browser που χρησιμοποιούμε και από τις πιο διαδεδομένες στον χειρισμό μεγάλου όγκου πληροφοριών. Για να κάνουμε ακόμα πιο ενδιαφέρον αυτά που θα περιγράψουμε παρακάτω, και να προσθέσουμε λίγο αλατοπίπερο θα συνδυάσουμε την σελιδοποίηση με την τεχνική της διάταξης ή όπως είναι γνωστή ως Sorting. Αυτό γιατί είναι γνωστό όπου έχουμε διαχείριση μεγάλου όγκου πληροφοριών οι δύο αυτές τεχνικές μας βοηθάν στον αποτελεσματικότερο τρόπο εύρεσης της ζητούμενης πληροφορίας. Ωστόσο στην περιγραφή που ακολουθεί δεν θα περιγράψουμε αναλυτικά την μέθοδο sorting απλά θα την αναφέρουμε μόνο σχηματικά, αφού στόχος της εργασίας αυτής είναι να παρουσιάσει την δυναμικότητα των εργαλείων που έχουν αναφερθεί ως τώρα και όχι να δημιουργήσουμε άλλο ένα εγχειρίδιο.

5.5.1 Διάταξη και Σελιδοποίηση

Αφού και οι δύο τεχνικές επηρεάζουν τον τρόπο που θα εμφανιστεί η πληροφορία, θα πρέπει να δούμε με ποιόν τρόπο επηρεάζει η μία την άλλη. Ας υποθέσουμε για παράδειγμα

ότι έχουμε έναν πίνακα με οκτώ γραμμές και 2 στήλες. Ο πίνακας αυτός είναι διατεταγμένος με βάση την πρώτη στήλη όπως φαίνεται στο σχήμα 5.3.

A	4
B	5
C	2
D	7
E	1
F	8
G	3
H	6

Σχ.5.3

Εάν γίνει αναδιάταξη του πίνακα με βάση πλέον την δεύτερη στήλη, τότε πολλές από τις γραμμές του θα αλλάξουν θέση Σχ.5.4.

E	1
C	2
G	3
A	4
B	5
H	6
D	7
F	8

Σχ.5.4

Ας δούμε τώρα τι θα γίνει εάν προσθέσουμε και την σελιδοποίηση των δεδομένων σε όλα τα παραπάνω. Υποθέτουμε ότι από τον server παίρνουμε μόνο τις τέσσερις πρώτες γραμμές και ότι η διάταξη των δεδομένων γίνεται στον browser του χρήστη. Το αποτέλεσμα που θα προκύψει θα είναι αυτό που φαίνεται στο σχήμα 5.5.

A	4
B	5
C	2
D	7

C	2
A	4
B	5
D	7

Σχ.5.5

Στην περίπτωση αυτή που το paging έγινε από τον server και το sorting από τον Browser, τα αποτελέσματα που προκύπτουν δεν είναι σωστά αφού το sorting αφορά μόνο τα δεδομένα τα οποία εμπεριέχονται στη πρώτη αυτή σελίδα του πίνακα. Εμείς όμως όταν επιλέγουμε sorting θέλουμε να η διαδικασία να αφορά το σύνολο των δεδομένων. Για να αποτρέψουμε να γίνει αυτό θα πρέπει και τις δύο διεργασίες να τις εκτελέσουμε είτε στον server είτε στον Browser. Με τον τρόπο αυτό η πρώτη γραμμή του πίνακα μετά την διάταξη του, θα είναι πραγματικά και το πρώτο στοιχείο των δεδομένων του, όπως φαίνεται και στο σχήμα 5.6.

A	4
B	5
C	2
D	7

E	1
C	2
G	3
A	4

Σχ.5.6

5.5.2 Υλοποίηση της Σελιδοποίησης

Ας δούμε λοιπόν πως θα προσθέσουμε την σελιδοποίηση σε έναν πίνακα. Θα πρέπει να συμφωνήσουμε πόσες γραμμές από τον πίνακα των δεδομένων θέλουμε να προβάλουμε ανά σελίδα.

```
$(document).ready(function () {
  $('table.paginated').each(function () {
    var currentPage = 0;
    var numPerPage = 10;
```

```

var $table = $(this);
$table.find('tbody tr').hide ()
.slice (currentPage * numPerPage,
(currentPage + 1) * numPerPage)
.show ();
});
});

```

Στο συγκεκριμένο παράδειγμα κάθε σελίδα προβάλλει δέκα γραμμές δεδομένων. Στο σημείο αυτό θέλαμε να τονίσουμε ότι βασιζόμαστε στην ύπαρξη του στοιχείου <tbody>, γιατί θέλουμε να χωρίσουμε τα δεδομένα του πίνακα από τις επικεφαλίδες κάθε στήλης. Με τον τρόπο αυτό όταν θα μεταβούμε στην επόμενη σελίδα οι επικεφαλίδες θα παραμείνουν και δεν θα χαθούν.

Για να επιλέξουμε τις δέκα πρώτες γραμμές του πίνακα, αρχικά θα κρύψουμε τον πίνακα. Στην συνέχεια θα επιλέξουμε ποια σελίδα επιθυμούμε να προβληθεί και τα δεδομένα του πίνακα που αντιστοιχούν στην σελίδα αυτήν θα προβληθούν στην οθόνη του χρήστη. Για την υλοποίηση της διαδικασίας αυτής χρησιμοποιούμε την μέθοδο slice (), η οποία μειώνει τον μέγεθος των δεδομένων. Για να υλοποιήσουμε την μέθοδο αυτή χωρίς απρόοπτα αποτελέσματα, θα πρέπει να εντοπίσουμε τον αριθμό του πρώτου και του τελευταίου δεδομένου σε αυτήν την σελίδα. Για να βρούμε το πρώτο δεδομένο απλά πολλαπλασιάζουμε τον αριθμό της συγκεκριμένης σελίδας με τον αριθμό των δεδομένων που έχουμε αποφασίσει να αποτελούν την κάθε σελίδα. Για παράδειγμα εάν θέλουμε την Τρίτη σελίδα θα πρέπει για να βρούμε το πρώτο στοιχείο της να πολλαπλασιάσουμε $3 \cdot 10 = 30$ άρα το στοιχείο τριάντα είναι και το πρώτο στοιχείο της σελίδας μας. Με τον ίδιο τρόπο για να βρούμε το τελευταίο στοιχείο της σελίδας, θα πρέπει να υπολογίσουμε το πρώτο στοιχείο της επόμενης σελίδας. Η μέθοδος slice () παίρνει όλες τις γραμμές μέχρι το πρώτο στοιχείο χωρίς να συμπεριλαμβάνει την δεύτερη παράμετρο της, και τις εμφανίζει με την Show (). Προσπαθήστε να το ξανασκεφτείτε αυτό που μόλις διαβάσατε και θα δείτε ότι είναι πραγματικά εκπληκτικό. Ο παραπάνω κώδικας εύκολα με βάση τις γνώσεις σας θα μπορούσε να ενσωματωθεί στο σε μια από τις μεθόδους Ajax που περιγράφηκαν σε προηγούμενες παραγράφους του κεφαλαίου αυτού.

Ας προχωρήσουμε λίγο παρακάτω, και ας απεικονίσουμε τον τρόπο με τον οποίο ο χρήστης θα μπορεί να καθορίζει ποια σελίδα επιθυμεί να προβληθεί.

Για να δημιουργήσουμε τον μηχανισμό αυτό θα πρέπει να υπολογίσουμε τον αριθμό των σελίδων που θα περιλάβουν τα δεδομένα μας.

```

var numRows = $table.find('tbody tr').length;
var numPages = Math.ceil (numRows / numPerPage);

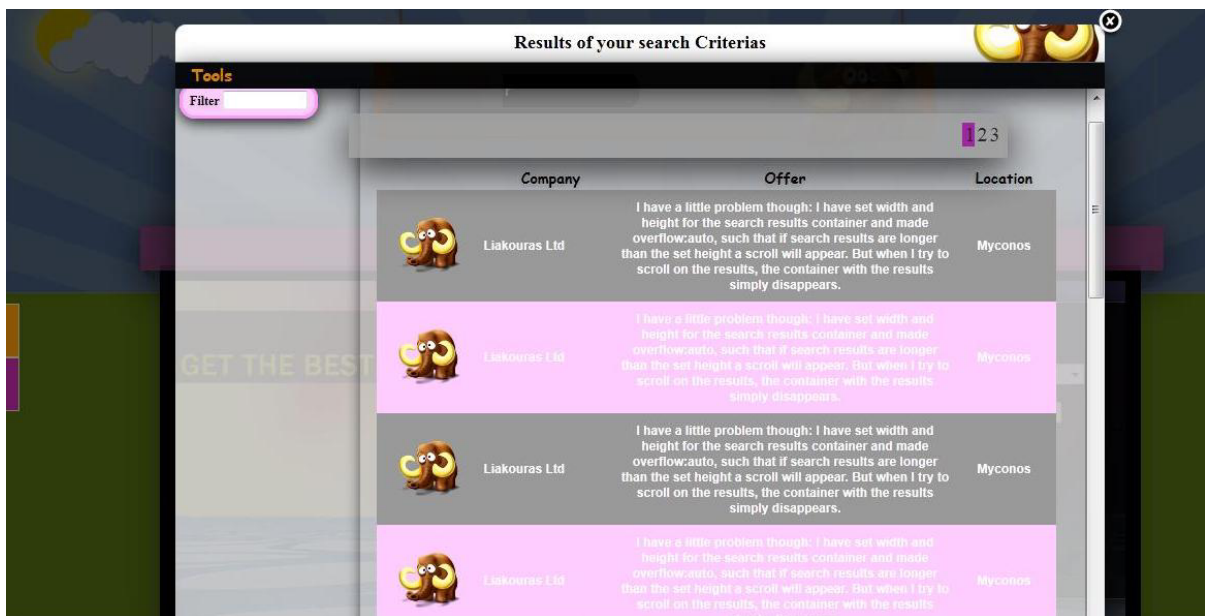
```

```

var $pager = $ ('<div class="pager"></div>');
for (var page = 0; page < numPages; page++) {
    $ ('<span class="page-number">' + (page + 1) + '</span>')
    .appendTo ($pager).addClass ('clickable');
}
$ pager.insertBefore ($table);

```

Δεν νομίζουμε να υπάρχει κανένας λόγος που κάποιος από εσάς δεν καταλαβαίνει το παραπάνω κομμάτι κώδικα. Θα μπορούσαμε να δώσουμε μια εξήγηση για την `Math.ceil (numRows / numPerPage)`, η οποία μέθοδος στρογγυλοποιεί το αποτέλεσμα της διαίρεσης, αφού ο αριθμός των σελίδων δεν θα μπορούσε να είναι άλλος από ακέραιος. Στην συνέχεια δημιουργούμε μία σειρά από κουμπιά ένα για κάθε σελίδα, το αποτέλεσμα φαίνεται στο παρακάτω σχήμα.



Σχ.5.7

5.5 Συμπέρασμα

Μετά την ολοκλήρωση του κεφαλαίου αυτού Θα πρέπει να έχετε αντιληφθεί τις δυνατότητες που προσφέρει το εργαλείο που ονομάζεται Ajax. Θα πρέπει να είστε σε θέση να απαντάτε στο ερώτημα ότι το Ajax δεν είναι μία γλώσσα προγραμματισμού, αλλά μια τεχνολογία η οποία συνδυάζει διάφορα εργαλεία προγραμματισμού κάτω από μια ομπρέλα, με τέτοιο τρόπο ώστε να λειτουργούν όλες μαζί. Και το κυριότερο που θα πρέπει να έχετε κατανοήσει, το γεγονός ότι σας δίνεται η δυνατότητα να επικοινωνείτε με τον server και να

ζητάτε δεδομένα τα οποία μπορείτε να τα παρουσιάζετε χωρίς να ανανεώνεται συνεχώς την σελίδα της εφαρμογής σας.

Αυτό βέβαια που θα συνιστούσαμε είναι να ανατρέξετε σε πηγές τόσο βιβλιογραφικές όσο και στο διαδίκτυο και να μελετήσετε παραδείγματα. Μην ξεχνάτε ότι η πληροφορική δεν είναι ιατρική, δεν θα πρέπει να αποστηθίσετε οτιδήποτε, αλλά θα πρέπει να μελετήσετε διάφορα παραδείγματα από τα οποία θα πάρετε τα καλύτερα χαρακτηριστικά τους και να τα αφομοιώστε ή ακόμα καλύτερα να τα εξελίξετε.

6

Διακομιστής και Βάση Δεδομένων

Σκοπός

Σε αυτό το Κεφάλαιο θα εξετάσουμε τα:

- Apache server,
- PHP 5.0, και
- MySql

Προσδοκώμενα αποτελέσματα

Ο χρήστης:

- να μπορεί να επιλέγει την καλύτερη τεχνολογία,
- να συγκρίνει τις τεχνολογίες μεταξύ τους,
- να μεταφέρει την εφαρμογή του σε διαφορετικούς υπολογιστές, και
- να παραμετροποιεί τον υπολογιστή του με σκοπό να μπορεί να συνδυάζει όλες

6.1 Παρουσίαση

Ο λόγος για τον οποίο αποφασίσαμε να γράψουμε αυτό το κεφάλαιο είναι γιατί όπως έχει αναφερθεί σε πολλά σημεία αυτού του συγγράμματος, συνοδεύεται από μία εφαρμογή η οποία αναπτύχθηκε για καθαρά εκπαιδευτικούς σκοπούς, στόχος της οποίας είναι να συνδυάσει όλα αυτά που αναφέρθηκαν προηγουμένως. Δεν θα μπορούσαμε όμως να κλείσουμε την αναφορά μας σε όλες αυτές τις τεχνολογίες χωρίς να αναφέρουμε τα εργαλεία που χρησιμοποιήσαμε τόσο για την αποθήκευση της εφαρμογής σε έναν Server, όσο και την βάση στην οποία είναι τοποθετημένα όλα τα δεδομένα με τα οποία συνδιαλέγεται καθώς και τον τρόπο επικοινωνίας της εφαρμογής με την βάση αυτή.

6.2 PHP

Τι είναι η PHP [1-2] τελικά; Που την έχουμε συναντήσει; Σε τι χρησιμεύει; Γιατί την επιλέγουμε; Είναι πολλά από τα ερωτήματα που δημιουργούνται διαβάζοντας τον τίτλο αυτής της ενότητας. Ένα είναι σίγουρο ότι την έχουμε συναντήσει στην πορεία αυτής της εργασίας, όχι ως κώδικα αλλά ως αναφορά σε αρχεία με κατάληξη «.php». Και να το πρώτο πράγμα το οποίο μάθατε, όλα τα αρχεία της php έχουν την κατάληξη που αναφέραμε. Επειδή είναι ένα αρκετά χρήσιμο εργαλείο και για να την περιγράψουμε θα πρέπει να αφιερώσουμε βιβλία, εμείς το μόνο που θα αναλύσουμε παρακάτω είναι μόνο ορισμένα χαρακτηριστικά της.

Απάντηση στο πρώτο ερώτημα έρχεται να δώσει ο παρακάτω ορισμός. Σύμφωνα με τον οποίο η PHP είναι μία server-side scripting γλώσσα σχεδιασμένη ειδικά για το Διαδίκτυο. Μέσα σε μία απλή HTML σελίδα μπορεί να παρεμβληθεί κώδικας PHP ο οποίος θα εκτελείται κάθε φορά που ένας χρήστης επισκέπτεται τη σελίδα. Ο PHP κώδικας μιας σελίδας μεταφράζεται στον Web server σε HTML ή σε περιεχόμενο που μπορεί να δει ο χρήστης. Με άλλα λόγια ο κώδικας της php μπορεί να παρεμβάλλεται ανάμεσα στο κώδικα της ιστοσελίδας μας περικλειόμενος ανάμεσα από τα χαρακτηριστικά αρχής (<?php) και τέλους (?>) της γλώσσας.

Με μία σύντομη αναδρομή στην ιστορία της γλώσσας αυτής, θα μπορούσαμε να αναφέρουμε ότι επινοήθηκε αρχικά το 1994 από έναν άνθρωπο, τον Rasmus Lerdorf. Αργότερα υιοθετήθηκε από πολλούς ταλαντούχους ανθρώπους και υπέστη τρεις μεγάλες αναβαθμίσεις ώστε να γίνει το προϊόν που βλέπουμε σήμερα. Η εξάπλωση της PHP στο Διαδίκτυο είναι εντυπωσιακή. Έρευνα τον Οκτώβριο του 2005 έδειξε ότι περισσότερα από 23 εκατομμύρια domains παγκοσμίως χρησιμοποιούσαν την PHP και αυτός ο αριθμός αυξανόταν ραγδαία. Η σημερινή εξάπλωση της PHP παρουσιάζεται στη διεύθυνση <http://www.php.net/usage.php>.

Ένα από τα χαρακτηριστικά της γλώσσας αυτής το οποίο βοήθησε και στην ραγδαία εξάπλωση της είναι ότι πρόκειται για προϊόν ανοικτού κώδικα (open-source). Τι σημαίνει

αυτό; Αυτό σημαίνει ότι ο σχεδιαστής που την χρησιμοποιεί έχει πρόσβαση στον πηγαίο κώδικα της γλώσσας. Έτσι μπορεί να τη διαμορφώσει να την αλλάξει ή να την επαναδημοσιεύσει χωρίς κόστος.

Το ακρωνύμιο PHP αρχικά σήμαινε **P**ersonal **H**ome **P**age, ωστόσο με τον καιρό ακολούθησε τη μόδα των περιοδικά επαναλαμβανόμενων ακρωνυμίων (π.χ. GNU =Gnu's Not Unix) και πλέον σημαίνει **Ph**p **H**ypertext **P**reprocessor.

6.2.1 Τα πλεονεκτήματα της

Ορισμένα από τα πλεονεκτήματα της PHP, τα οποία την κάνουν τόσο δημοφιλή σε σχέση με άλλες server-side scripting γλώσσες είναι:

- Η υψηλή απόδοση.
- Η πληθώρα βιβλιοθηκών.
- Η πρόσβαση στον πηγαίο κώδικα.
- Η δυνατότητα σύνδεσης με διάφορα συστήματα διαχείρισης βάσης δεδομένων.
- Το χαμηλό κόστος.
- Η μεταφερσιμότητα.
- Η ευκολία μάθησης και χρήσης.

Η επίσημη ιστοσελίδα της PHP είναι η <http://www.php.net>, ωστόσο ο χρήστης μπορεί να βρει πλήθος ιστοσελίδων που υποστηρίζουν την ανάπτυξη εφαρμογών ή βιβλιοθηκών της γλώσσας.

6.3 MySQL

Η MySQL [1-3] εμφανίστηκε πρώτη φορά δημοσίως το 1996, ωστόσο η ανάπτυξή της ξεκινάει από το 1979. Έχει κερδίσει το βραβείο Linux Journal Readers' ChoiceAward πάρα πολλές φορές.

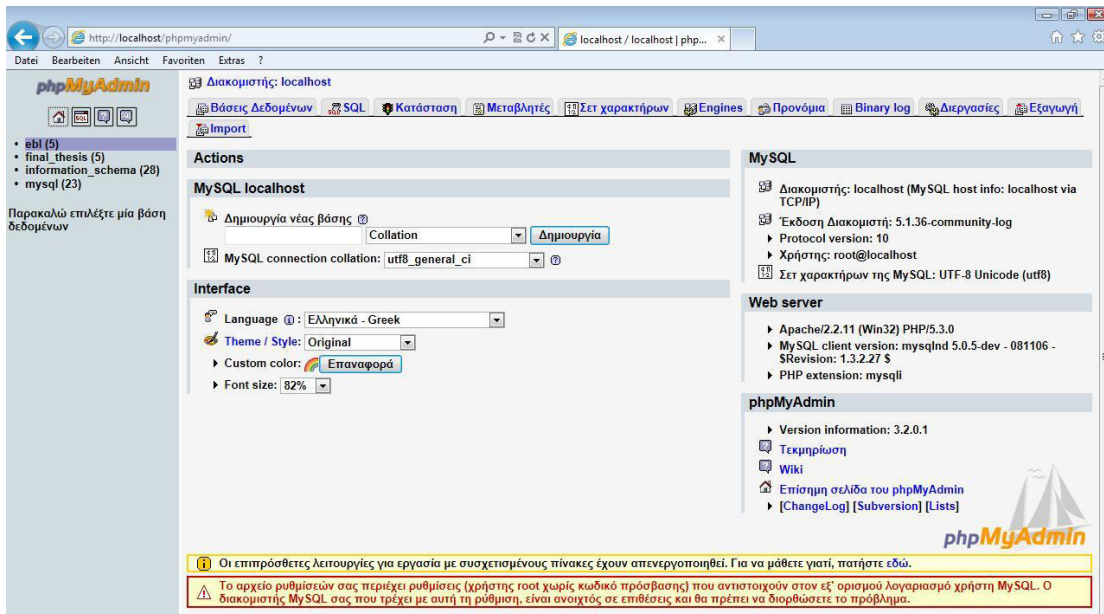
Πρόκειται για ένα ισχυρό πολύ γρήγορο σύστημα διαχείρισης σχεσιακής βάσης δεδομένων, με δυνατότητες αποδοτικής αποθήκευσης, αναζήτησης, ταξινόμησης και ανάκτησης δεδομένων. Ο server της MySQL ελέγχει την πρόσβαση στα δεδομένα έτσι ώστε:

- α) Να μπορούν πολλοί χρήστες ταυτόχρονα να τα χρησιμοποιούν.
- β) Να έχουν γρήγορη πρόσβαση σε αυτά.

γ) Μόνο εξουσιοδοτημένα άτομα να έχουν πρόσβαση σε αυτά.

Η MySQL χρησιμοποιεί τη γλώσσα SQL (Structured Query Language) η οποία έχει καθιερωθεί παγκοσμίως στις βάσεις δεδομένων.

Μία δυναμική λειτουργία της η οποία και την έχει βοηθήσει να εδραιωθεί, είναι το γεγονός πως όπως και η PHP, έτσι και η MySQL είναι ένα προϊόν ανοικτού κώδικα, ωστόσο υπάρχουν και άδειες για εμπορική χρήση της.



Σχ.6.1 Κεντρική σελίδα γραφικού περιβάλλοντος My SQL

6.3.1 Τα πλεονεκτήματα της.

Κάποια από τα πλεονεκτήματα της MySQL είναι:

- Η πρόσβαση στον πηγαίο κώδικα.
- Η υψηλή απόδοση.
- Η ευκολία μάθησης και χρήσης.
- Το χαμηλό κόστος.
- Η συνεχής υποστήριξη και βελτίωση.
- Η ταχύτητα.

Οι σχεδιαστές της MySQL προκειμένου να διατηρήσουν την υψηλή απόδοση δεν ενσωμάτωσαν σε αυτήν στοιχεία που υπάρχουν σε άλλα πακέτα. Η μη ενσωμάτωση των

στοιχείων αυτών αποτέλεσε και την αφορμή για την γέννηση ορισμένων μειονεκτημάτων όπως η απουσία Stored procedures, Subqueries, Unions, Triggers, Referential integrity.

Τα μειονεκτήματα αυτά δεν μπόρεσαν να παραβλέψουν ορισμένα από τα χαρακτηριστικά τα οποία συνετέλεσαν στην διάδοση της όπως, η MySQL είναι «ελαφριά» και πολύ γρήγορη και ιδανική για εφαρμογές στο Διαδίκτυο. Μάλιστα η ταχύτητά της είναι σημαντικά μεγαλύτερη από άλλα εμπορικά πακέτα.

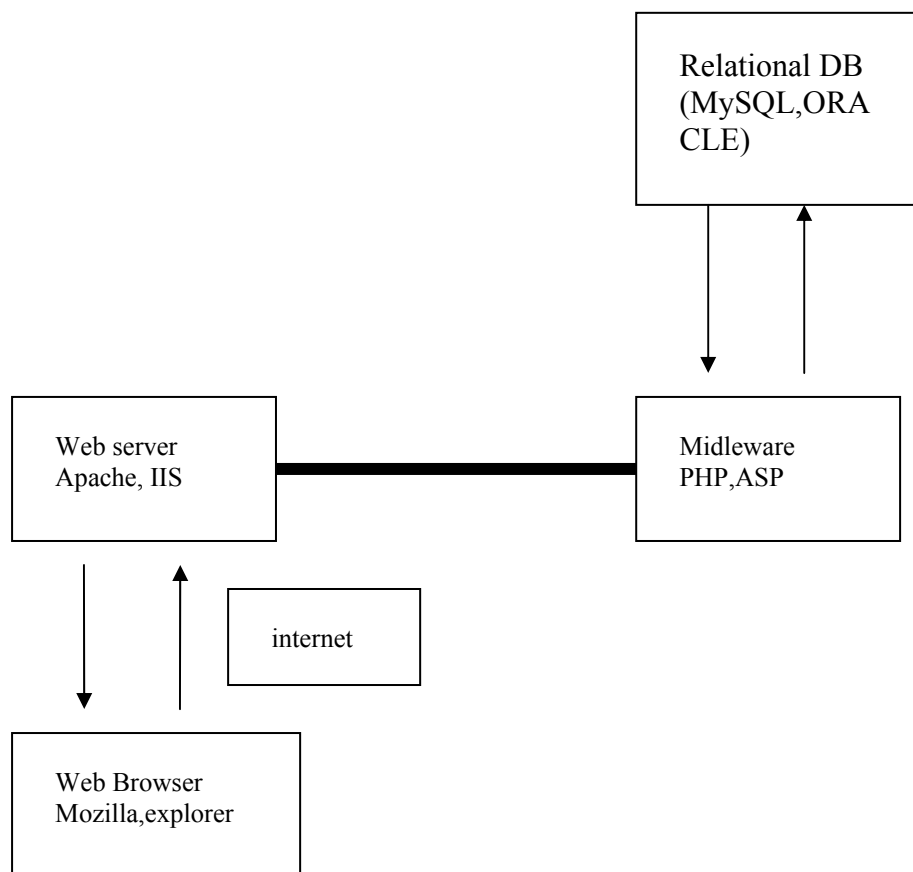
6.3 Apache server

Ένα είναι σίγουρο ότι δεν πρόκειται για μια φυλή ινδιάνων, εάν και το όνομα του μας παραπέμπει. Είναι ένας server για το Διαδίκτυο. Δουλειά του είναι να τρέχει πάνω από το λειτουργικά σύστημα ενός υπολογιστή και να εξυπηρετεί τις αιτήσεις των χρηστών. Οι δύο δημοφιλέστεροι web servers είναι ο Apache [52] και ο Internet Information Server [53] (IIS) της Microsoft. Ο IIS εξαρτάται πλήρως από το περιβάλλον των Windows εφόσον σχεδιάστηκε για να συνεργάζεται με τα υπόλοιπα προϊόντα της Microsoft.

Ο Apache από την άλλη ακολουθεί την φιλοσοφία του λογισμικού ανοικτού κώδικα (open-source). Για αυτόν τον λόγο είναι ο κυρίαρχος στο Διαδίκτυο. Αν και αποδίδει καλύτερα σε περιβάλλοντα Unix, ο Apache μπορεί να τρέξει χωρίς κανένα πρόβλημα και σε περιβάλλοντα Windows. Ο Apache γενικά χαρακτηρίζεται ως ιδιαίτερα γρήγορος και πολύ σταθερός. Το αρνητικό σχόλιο που ακούγεται ωστόσο συχνά για αυτόν είναι ότι επειδή ακολουθεί τη φιλοσοφία του Unix δεν διαθέτει ένα φιλικό γραφικό περιβάλλον διαχείρισης. Οι ρυθμίσεις γίνονται συνήθως με την γραμμή εντολών ή με αλλαγή γραμμών σε αρχεία κειμένου.

6.4 Πώς συνδυάζονται όλα αυτά;

Όλα τα παραπάνω για να μπορέσουμε να τα λειτουργήσουμε θα πρέπει να βρεθεί ένας, μαγικός η μη, τρόπος για να τα βάλουμε να μιλήσουν μεταξύ τους. Στο σχήμα [1] που ακολουθεί παρουσιάζεται ο τρόπος αυτός.



Σχ.6.2

Στην εφαρμογή μας όπως και σε κάθε εφαρμογή στο διαδίκτυο γίνεται χρήση μιας βάσης δεδομένων στην οποία αποθηκεύονται, τα δεδομένα που χρησιμοποιούνται. Όπως για παράδειγμα στοιχεία χρηστών κωδικοί και άλλα. Έτσι λοιπόν ο χρήστης ζητάει να δει την εφαρμογή μας στον web browser (π.χ. Internet Explorer, Netscape, Firefox) εισάγοντας το αντίστοιχο URL. Η αίτηση διασχίζει το Διαδίκτυο και φτάνει στον web server (π.χ. Apache, IIS) που περιέχει την ιστοσελίδα της εφαρμογής μας. Ο web server καλεί το κατάλληλο module (π.χ. PHP, ColdFusion, ASP, JSP) προκειμένου να μεταφράσει την σελίδα σε μορφή που μπορεί να παρουσιάσει ο web browser. Κατά την μετάφραση του κώδικα από το module πραγματοποιείται η σύνδεση με το σύστημα διαχείρισης της βάσης δεδομένων και γίνεται η επεξεργασία (εγγραφή, τροποποίηση, διαγραφή) των δεδομένων. Τελικά η ιστοσελίδα της

εφαρμογής μας παρουσιάζεται στον web browser του χρήστη περιέχοντας όλα τα απαραίτητα δεδομένα.

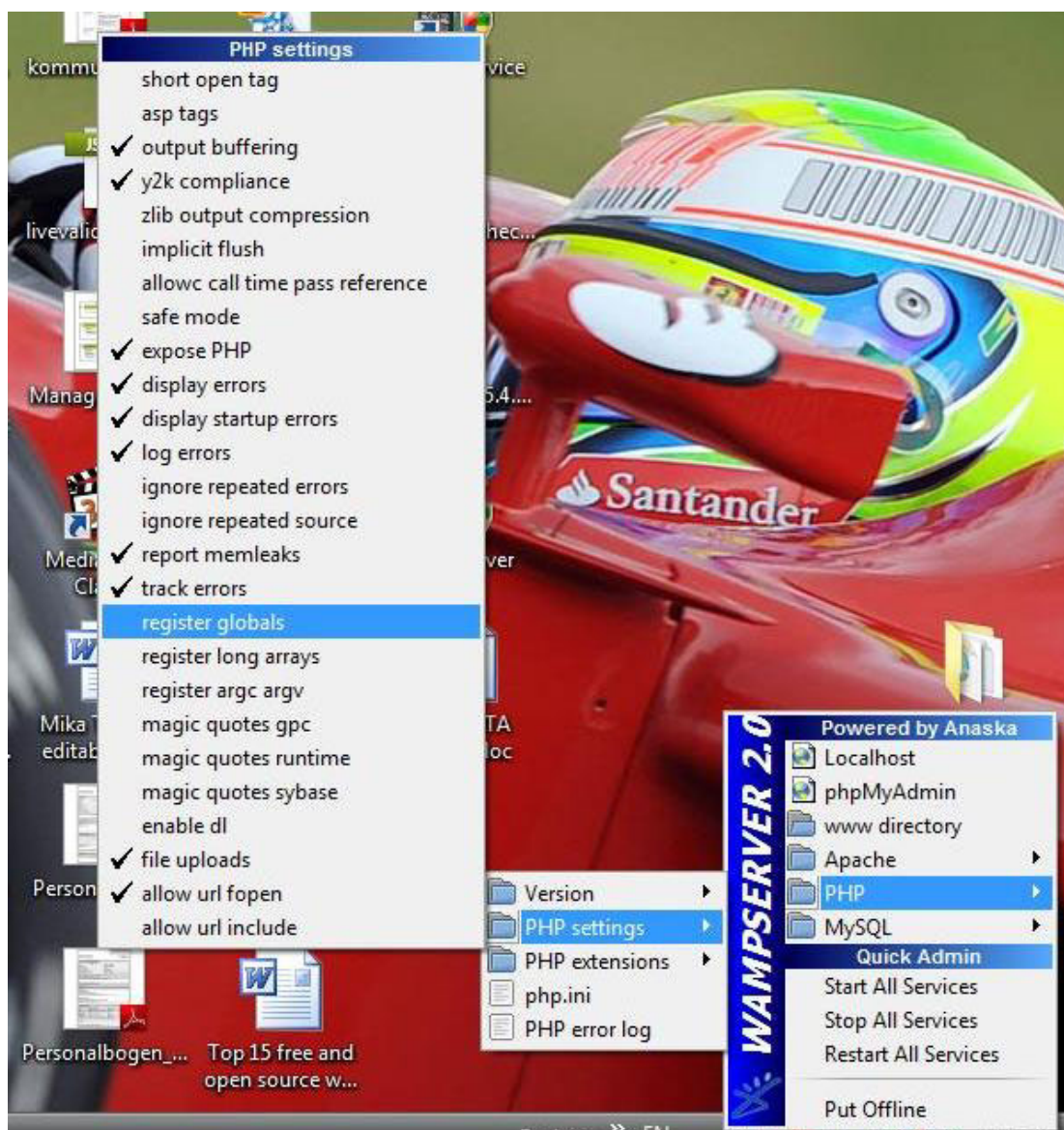
6.5 Εγκατάσταση.

Η PHP υπάρχει διαθέσιμη στην ιστοσελίδα <http://www.php.net>. Ο χρήστης μπορεί να επιλέξει ανάμεσα από δύο αρχεία εγκατάστασης τα αρχεία πηγαίου κώδικα και δυαδικά αρχεία. Ο πιο απλός τρόπος εγκατάστασης τον οποίο και προτείνουμε είναι να επιλέξετε τα δυαδικών αρχείων τα οποία είναι pre-compiled και pre-configured. Σε περίπτωση που ο χρήστης επιλέξει να κάνει την εγκατάσταση με χρήση των αρχείων πηγαίου κώδικα τότε θα πρέπει να καθορίσει μόνος του τις ρυθμίσεις ανάλογα με το λειτουργικό του σύστημα και να κάνει compile τα αρχεία του κώδικα. Ωστόσο υπάρχουν και δωρεάν ολοκληρωμένα πακέτα για την εγκατάσταση και χρήση AMP (Apache – MySQL – PHP), όπως περιγράφεται στην επόμενη παράγραφο.

Για μια εύκολη και γρήγορη εγκατάσταση της PHP που θα τρέχει και θα ενσωματώνεται σε έναν web server μπορούμε να χρησιμοποιήσουμε ένα ολοκληρωμένο πακέτο AMP (Apache – MySQL – PHP). Υπάρχουν διάφορα τέτοια πακέτα στην αγορά τα οποία μπορείτε να κατεβάσετε και να εγκαταστήσετε. Για την εφαρμογή που αναπτύξαμε κάναμε χρήση του WAMP5 [54]. Κατά την εγκατάσταση του εγκαθιστά μαζί και τον Apache, την MySQL, και την PHP. Οι εκδόσεις των λογισμικών δεν αναφέρονται για τον λόγο του ότι, όταν θα κάνετε την εγκατάσταση μπορεί να έχουν τροποποιηθεί.

6.5.1 Παραμετροποίηση

Είναι γεγονός ότι για να βάλουμε όλα αυτά να παίζουν μπάλα, και να συνεργάζονται μαζί θα πρέπει να κάνουμε κάποιες ρυθμίσεις. Θα πρέπει να εξετάσουμε τον τρόπο με τον οποίο μπορούμε να επέμβουμε στην Php, με σκοπό να μην αντιμετωπίσουμε κανένα πρόβλημα. Όλες οι ρυθμίσεις που αφορούν την Php είναι αποθηκευμένες σε ένα αρχείο το όνομα του οποίου είναι php.ini. Το αρχείο αυτό αποθηκεύεται στον υπολογιστή κατά την εγκατάσταση, σε αρχείο η θέση του οποίο αλλάζει ανάλογα με το λειτουργικό σύστημα. Με την WAMP5 μπορούμε να τροποποιήσουμε το αρχείο αυτό το οποίο είναι σε μορφή κειμένου, με τρόπο που φαίνεται στο παρακάτω σχήμα.



Σχ.6.3

Μία σημαντική παρατήρηση είναι ότι σε περίπτωση που κάνουμε οποιαδήποτε αλλαγή στο αρχείο μας, θα πρέπει την αλλαγή αυτή να την αποθηκεύσουμε. Αρκετοί κάνουν το εξής λάθος, επειδή τροποποιούν το αρχείο με το textpad, κατά την αποθήκευση το δεν προσέχουν το όνομα με αποτέλεσμα να το αποθηκεύουν ini.php.txt, που είναι φυσικά λάθος.

Κατά την εγκατάσταση δημιουργείτε ο φάκελος `C:\wamp\www`, είναι ο φάκελος μέσα στον οποίο αποθηκεύονται όλα τα αρχεία τα οποία μπορεί να δει ο χρήστης από τον server. Σε αυτόν τον φάκελο θα πρέπει να αποθηκευτούν όλα τα αρχεία της εφαρμογής που θα κάνετε, με τον τρόπο αυτό θα μπορούν να ερμηνευτούν από την μηχανή της php και να εμφανιστούν από τον Apache. Ας πάρουμε το πιο απλό σενάριο ότι αναπτύσσουμε σελίδες με κώδικα PHP στον προσωπικό μας υπολογιστή και δεν μας ενδιαφέρει να τις ανεβάσουμε στο Διαδίκτυο. Τότε ο Apache server αντιστοιχεί στην διεύθυνση `http://localhost` ή στην ip `127.0.0.1`. Σε περίπτωση που διαθέτουμε σταθερή ip τότε μπορούμε να ανεβάσουμε τις

ιστοσελίδες μας στο Διαδίκτυο δηλώνοντας στον Apache την ip και το domain name. Ένα συχνό λάθος που κάνουν πολλοί είναι ότι αποθηκεύουν το αρχείο της σελίδας τους στον **C:\wamp\www** και στην συνέχεια τον ανοίγουν από τον φάκελο του δίσκου. Αυτό θα έχει ως αποτέλεσμα να βλέπουν τον ίδιο τον κώδικα.

Ο μόνος τρόπος για να δούμε τα αποτελέσματα του κώδικά μας είναι να έχουμε εκκινήσει τον Apache (από τη συντόμευση που δημιουργήθηκε μετά την εγκατάσταση) και να δούμε την ιστοσελίδα μας μέσα από την διεύθυνση `http://localhost`. Για παράδειγμα `http://localhost/myfile.php`.

6.5.1 Μεταφερσιμότητα.

Για να μεταφέρουμε την εφαρμογή που δημιουργήσαμε με τα εργαλεία που περιγράψαμε στην διάρκεια αυτού του κεφαλαίου, θα πρέπει να μεταφέρουμε τόσο τα αρχεία όσο και την βάση δεδομένων στον νέο υπολογιστή. Για την μεταφορά των αρχείων, αρκεί να τα αντιγράψετε στον αντίστοιχο φάκελο C:\wamp\www του νέου υπολογιστή. Για να μεταφέρουμε την βάση δεδομένων υπάρχουν δύο τρόποι, ο ένα όμως είναι ο πιο σωστός. Να πάρουμε δηλαδή την βάση δεδομένων και τα αρχεία σε μορφή SQL, και να τα τρέξουμε στον νέο Server, αυτός ο τρόπος είναι και ο ενδεδειγμένος σε περίπτωση που θα τρέξει η βάση σε κάποια νέα έκδοση της MySQL.

Ο δεύτερος τρόπος ο οποίος είναι και ο πιο εύκολος είναι να αντιγράψουμε τον φάκελο της βάσης στον αντίστοιχο φάκελο του νέου υπολογιστή. Η μέθοδος αυτήν δεν παρουσιάζει κανένα πρόβλημα ειδικά αν μιλάμε για την ίδια έκδοση της MySQL. Τι θα κάνουμε δηλαδή, Κάθε φορά που δημιουργείτε μια βάση, δημιουργείτε και ένας φάκελος της μορφής C:\wamp\bin\mysql\mysql5.0.45\data\ilias. Όπου *ilias* έστω ότι είναι το όνομα της βάσης (θα μπορούσε να είναι όποιο όνομα εμείς θελήσουμε όπως Μανώλης, Θανάσης κ.α).

Θα πρέπει να προσέξετε ότι μαζί με την βάση σας δεν μεταφέρονται και τα προφίλ των χρηστών τα οποία δημιουργήσατε, και αυτό γιατί δεν αποτελούν μέρος των δεδομένων της βάσης σας.

6.6 Συμπέρασμα

Θα μπορούσαμε να πούμε σαν επίλογο του κεφαλαίου αυτού, ότι δεν αναφερθήκαμε εκτεταμένα σε κάθε ένα από τα εργαλεία που περιγράφηκαν. Αυτό έχει να κάνει κυρίως με το γεγονός ότι ως θέμα της διπλωματικής εργασίας είναι οι τεχνολογίες που αναπτύσσονται τον τελευταίο καιρό με αποτέλεσμα να φέρνουν επανάσταση στο χώρο του διαδικτύου και σε αυτό που τείνει να εξελιχθεί και ονομάζεται web3.0. Και θα ρωτούσε κάποιος και γιατί η PHP δεν αποτελεί μέρος των τεχνολογιών αυτών αφού στον Ajax η επικοινωνία με τον server για ανταλλαγή πληροφοριών γίνεται μέσω αυτής. Και απαντάμε ότι βεβαίως και αποτελεί μέρος της, όμως επιλέξαμε για λόγους χώρου και χρόνου να περιγράψουμε αυτές που θεωρούσαμε ότι μας προσφέρουν κάτι το διαφορετικό από άποψη οικονομίας σε γραμμές κώδικα. Το γεγονός ότι δεν κάναμε εκτεταμένη αναφορά δεν υποβαθμίζει τα εργαλεία αυτά για τα οποία έχουν γραφεί κατά καιρούς πάρα πολλά βιβλία.



Συμπεράσματα

Καταρχάς δεν θα ήταν δυνατόν να κλείσουμε την εργασία αυτή, που αφορά στις νέες τεχνολογίες διαδικτυακού προγραμματισμού, χωρίς να αναπτύξουμε ορισμένα συμπεράσματα που προέκυψαν.

7.1 Ο τρόπος που αντιλαμβανόμαστε το Web!

Είναι γενικά παραδεκτό ότι όλες αυτές οι τεχνολογίες, θα μεταμορφώσουν το τοπίο του web αισθητικά, και θα δώσουν τη δυνατότητα σε αυτούς που ασχολούνται με τον χώρο αυτό να δημιουργήσουν πραγματικά έργα τέχνης. Δεν θα λείψουν βέβαια και οι υπερβολές στην χρήση των δυνατοτήτων που προσφέρουν αυτές οι νέες τεχνολογίες στην προσπάθεια επίδοξων προγραμματιστών να δείξουν τις ικανότητες τους, με αποτέλεσμα την δημιουργία αντιαισθητικών και δύσχρηστων εφαρμογών.

Όλες αυτές οι εξελίξεις δεν θα μπορούσαν να αφήσουν ανέπαφες και τις εταιρείες που ασχολούνται με τον σχεδιασμό των σύγχρονων φυλλομετρητών (web Browsers), όπως τους Firefox, Chrome, Opera και Safari, στο να επιδοθούν σε έναν ατελείωτο αγώνα δρόμο για να εκμεταλλευτούν τις δυνατότητες αυτών των εργαλείων. Παρατηρείται γενικά στον χώρο του διαδικτυακού προγραμματισμού μια μετατόπιση του βάρους, από εκεί που ήταν στους σχεδιαστές/προγραμματιστές, να περνάει τώρα στις εταιρείες οι οποίες πρέπει να τρέξουν για να προλάβουν τις απαιτήσεις των προγραμματιστών.

Η πίεση αυτή έφερε αποτέλεσμα με το λανσάρισμα στην αγορά από την Chrome της μηχανή WebKit [55], μιας μηχανής διάταξης που ικανοποιεί μια μεγάλη μερίδα των προδιαγραφών του CSS3 και HTML5 που έχουν ήδη υλοποιηθεί. Έτσι παρατηρείται μια μεγάλη στροφή στην αγορά σε σχέση με τον κυρίαρχο Internet Explorer. Σύμφωνα με ανακοινώσεις [56-57-58-59] στην Γερμανία ο Firefox αρχίζει να ξεπερνάει τον Internet Explorer, ο οποίος ακολουθεί με μεγάλη προσπάθεια τις εξελίξεις που φέρνουν τα νέα εργαλεία που μελετήσαμε.

7.2 Χώροι κοινωνικής Δικτύωσης!

Κανείς δεν μπορεί να αρνηθεί ότι το 2006 [60] ήταν μια σημαντική χρονιά για τα μέσα μαζικής ενημέρωσης του διαδικτύου: το Twitter εμφανίστηκε, και από τότε έχει γίνει σλόγκαν και για πολλούς αποτελεί μια αίθουσα συνεδριάσεων και το γραφείο τους. Είναι προφανές ότι θα συνεχίσει σε ένα μεγάλο μέρος του ιστού στο μέλλον.

Με την αύξηση των αποδεκτών τέτοιων δικτυακών τόπων, γεννήθηκε η απαίτηση για επεξεργασία δεδομένων σε πραγματικό χρόνο, καθώς επίσης και την μεταφορά φόρτου εργασίας από το επίπεδο του διακομιστή (server), στο επίπεδο του χρήστη πλέον. Αυτά όλα θα έχουν ως αποτέλεσμα την δημιουργία ταχύτερων εφαρμογών και την άμεση και χωρίς καθυστέρηση ενημέρωση. Παρατηρούμε ότι οι αλλαγές αυτές στην τεχνολογία σχεδίασης

και ανάπτυξης εφαρμογών στο διαδίκτυο, άρχισαν να εφαρμόζονται στις περισσότερες μηχανές αναζήτησης. Έτσι έχουμε περισσότερους ανθρώπους που συμμετέχουν στη δημιουργία των πληροφοριών στο διαδίκτυο, με αποτέλεσμα ο τρόπος με τον οποίο παίρνουμε πληροφορίες να μετατοπιστεί από το να είναι από μια μοναδική πηγή, σε μια κοινότητα που δημιουργεί την πηγή. Εάν για παράδειγμα ψάχνετε για πληροφορίες σχετικά με ένα κατάστημα επισκευής αυτοκινήτων, θα σας δείξει τα τελευταία tweets και τις ενημερώσεις του Facebook σχετικά με την εταιρεία, αντί των παρωχημένων και στατικών πληροφοριών της.

Ένα πράγμα το οποίο πιστεύουμε θέλει ακόμα προσπάθεια στο να συμβάλει στην αύξηση της αποτελεσματικότητας των εργαλείων που περιγράψαμε, είναι η δομή που ακολουθούν. Σε αντίθεση με τις παραδοσιακές γλώσσες, όπως τη C για παράδειγμα, η οποία έχει μια δομή που συμβάλει στην βελτίωση της πολυπλοκότητας του κώδικα της με αποτέλεσμα να έχουμε βέλτιστα προγράμματα, τα εργαλεία που περιγράψαμε υστερούν κατά πολύ. Αποτέλεσμα αυτού είναι η λειτουργία τους να είναι μη βέλτιστη ως προς την απόδοση, και να διαφέρει, ως προς αυτή στους διάφορους φυλλομετρητές (Browsers).

7.3 Η εφαρμογή μας!

Όπως και τονίσαμε πάρα πολλές φορές στην διάρκεια αυτού του συγγράμματος, για να καλύψουμε τις ανάγκες της εργασίας και να μπορέσουμε μέσα από την πράξη να σας περάσουμε τις εμπειρίες μας από την εφαρμογή όλων των τεχνικών που μελετήσαμε, δημιουργήσαμε μια πιλοτική ιστοσελίδα. Η ιστοσελίδα δημιουργήθηκε αποκλειστικά με τα εργαλεία που περιγράφηκαν, αποτέλεσμα αυτού είναι η δημιουργία μιας client-side εφαρμογής η οποία εκμεταλλεύεται όλες τις δυνατότητες των σύγχρονων μεθόδων προγραμματισμού. Ως αποτέλεσμα της χρήσης όλων των τεχνολογιών σε συνδυασμό μαζί, είναι να έχουμε μια εφαρμογή η οποία κάνει ελάχιστη χρήση γραφικών φτιαγμένων με προγράμματα όπως το photoshop. Αποτέλεσμα αυτού είναι να μην έχουμε κάνει χρήση στατικών γραφικών και όλα μας τα γραφικά να δημιουργούνται με την χρήση του CSS3. Αυτό έχει ως αποτέλεσμα η εφαρμογή να μην περιλαμβάνει μεγάλου όγκου εικόνες, που την καθυστερούν κατά την φόρτωση της. Ένα δεύτερο σημείο που είναι απαραίτητο να επισημάνουμε είναι το γεγονός πως η χρήση του Ajax και του JQuery, έδωσαν φτερά στην εφαρμογή μας. Τι σημαίνει αυτό, σίγουρα δεν αποτελεί διαφήμιση για γνωστό αναψυκτικό, όμως η χρήση των εργαλείων αυτών μας παρέχει την δυνατότητα να δημιουργούμε τον κώδικα κατά την διάρκεια της εκτέλεσης μίας συγκεκριμένης απαίτησης. Για παράδειγμα υπάρχουν κομμάτια του κώδικα τα οποία δεν προϋπάρχουν στατικά κατά την δημιουργία της εφαρμογής, αλλά παράγονται δυναμικά με την χρήση γραμμών κώδικα. Αυτό έχει ως αποτέλεσμα η εφαρμογή μας να εκτελείται σαν να αποτελεί μια client side εφαρμογή, αφού

το μεγαλύτερο μέρος της επεξεργασίας των δεδομένων και των αποτελεσμάτων που αυτά παράγουν γίνεται στο επίπεδο του χρήστη. Θα μπορούσε να παρατηρηθεί επίσης και να συμπεριληφθεί στα πλεονεκτήματα της εφαρμογής, ότι δεν γίνεται καμία ανανέωση της αρχικής σελίδας και όλες οι επιμέρους εργασίες εκτελούνται υπό την μορφή παραθυρικού περιβάλλοντος. Ο τρόπος ανάπτυξης της εφαρμογής περιορίζει την χρήση του διακομιστή μόνο στην εισαγωγή και εξαγωγή δεδομένων. Αφού λοιπόν δεν αποτελεί αυτοσκοπό της εργασίας αυτής η δημιουργία μιας πλήρους εμπορικής εφαρμογής, συμπεριλάβαμε ορισμένα παραδείγματα όπως εισαγωγή στοιχείων για την εγγραφή ενός νέου χρήστη και ταυτόχρονο έλεγχο της ορθότητας τους σε επίπεδο χρήστη, την εγγραφή ενός χρήστη, την διαμόρφωση του profil του, την αναζήτηση προσφορών και πολλές άλλες δυνατότητες οι οποίες εκτελούνται με τρόπο που απασχολείται ο διακομιστής το δυνατό λιγότερο. Μελετώντας τις τεχνικές που περιγράφονται στην εργασία και κατανοώντας στην πράξη την εφαρμογή των τεχνικών αυτών μέσα από το πρόγραμμα που γράφτηκε για τον σκοπό αυτό, αποκτάμε μια εικόνα του τρόπου που θα πρέπει να κατασκευάζονται οι νέες ιστοσελίδες έτσι ώστε να κάνουν το δυνατό λιγότερη χρήση του διακομιστή. Γνωρίζουμε πως έχουμε μία ραγδαία αύξηση της χρήσης του διαδικτύου. Η παραγωγή εφαρμογών που θα απασχολούν λιγότερο των διακομιστή που τις φιλοξενεί θα είχε ως αποτέλεσμα την επίλυση προβλημάτων υπερφόρτωσης που αποτελούν ένα από του κύριους πονοκέφαλους του διαδικτυακού προγραμματισμού.

Στο σημείο αυτό θα αναφέρουμε κάποια τεχνικά χαρακτηριστικά. Το κύριο αρχείο της εφαρμογής με το οποίο καθορίζεται και το Layout της είναι το `index3.php`, το οποίο αποτελείται από 555 γραμμές κώδικα. Δεν τελειώνει εδώ όμως το θέμα. Η εφαρμογή η οποία όπως είπαμε είναι εφαρμογή δυναμική, που σημαίνει αυτό κατά την διάδραση της με τον χρήστη γεννάται ο κώδικας.

Για τον σκοπό αυτό χρειάστηκε να γίνει χρήση της τεχνολογίας Ajax, η οποία όπως και είπαμε είναι συνδυασμός διαφόρων τεχνικών και γλωσσών προγραμματισμού. Στην δικιά μας περίπτωση περιλαμβάνει συνδυασμό JQuery και PHP. Για να καλύψουμε την επικοινωνία της εφαρμογής μας με την βάση μας, όπως και τονίσαμε για της ανάγκες της εφαρμογής στήθηκε ο apache server, πάνω στον οποίο τρέχει η MySQL βάση μας, χρειάστηκαν να δημιουργηθούν 15 αρχεία γραμμένα σε PHP, συνολικού κώδικα 1300 γραμμών. Επίσης για την λειτουργία του Ajax χρειάστηκε να γραφεί κώδικας σε JQuery, ο οποίος περιλαμβάνεται σε 4 αρχεία με ονόματα `iliasterpready.js` `scripiliias.js` `box.js` `ilisidenav.js`, συνολικού κώδικα 1100 γραμμών κώδικα. Δεν θα πρέπει να αγνοήσουμε το γεγονός ότι για την λειτουργία της εφαρμογής χρειάστηκαν και κάποια frameworks τα οποία και απαιτήθηκε να προσαρμοστούν καταλλήλως για να εναρμονιστούν στις απαιτήσεις της εφαρμογής. Ο κώδικας για αυτά τα frameworks δεν συνυπολογίστηκε σε καμία από τις παραπάνω περιπτώσεις. Τέλος δεν θα μπορούσαμε να ξεχάσουμε ότι η διαμόρφωση της εμφάνισης της ιστοσελίδας μας, πραγματοποιήθηκε με CSS3, για το λόγο αυτό

κατασκευάστηκαν 2 αρχεία το index3.css και το slide.css συνολικού αριθμού 2951 γραμμών κώδικα.

Βέβαια όπως και αναφέραμε δεν δημιουργήσαμε μια εμπορική εφαρμογή, ωστόσο όμως κάναμε μια εφαρμογή πολύ κοντά σε αυτές της αγοράς η οποία σε κάποια καινούρια της έκδοση θα μπορούσε πολύ άνετα να θεωρηθεί εφάμιλλη. Θα προτείναμε για τον σκοπό αυτό να προσεχθούν κάποιες λεπτομέρειες όσον αφορά το κομμάτι του Ajax το οποίο εάν εμπλουτιστεί θα μπορούσε να μικρύνει τον κώδικα του βασικού μας αρχείου index3.php, κατά 50%. Επίσης επειδή εφαρμογή δημιουργήθηκε χωρίς καμία προγενέστερη γνώση μας στον τομέα του διαδικτυακού προγραμματισμού, ορισμένα κομμάτια του κώδικα που αναπτύχθηκαν στα αρχικά στάδια της εργασίας αυτής ίσως να πρέπει να τροποποιηθούν κατά τρόπο που θα αυξάνει την αποδοτικότητα και την ταχύτητα της εφαρμογής.

Εν κατακλείδι θα τονίσουμε ότι τα επόμενα χρόνια θα ακολουθήσει μια επανάσταση στον τρόπο που βλέπουμε και αντιλαμβανόμαστε μια εφαρμογή ιστού. Δεν θα διαφέρει καθόλου από τις εφαρμογές που έχει ο καθένας μας εγκατεστημένες στον υπολογιστή του, αφού η όλη φιλοσοφία τους θα είναι πλέον δομημένη στην πλευρά του χρήστη.

Βιβλιογραφία

Σκοπός αυτού της διπλωματικής εργασίας ήταν η παρουσίαση όλων αυτών των νέων τεχνολογιών που κατακλύζουν τον χώρο του διαδικτυακού προγραμματισμού. Η παρουσίαση αυτή ωστόσο δεν καλύπτει σε βάθος όλες τις δυνατότητες αυτών των εργαλείων.

Έχουν εκδοθεί πάρα πολλά βιβλία σχετικά με τα αντικείμενα αυτά. Για τις ανάγκες αυτού του εγχειριδίου εμείς χρησιμοποιήσαμε αρκετά από αυτά τα οποία και σας τα προτείνουμε για περαιτέρω εμβάθυνση, στα αντικείμενα που αναλύσαμε στα προηγούμενα κεφάλαια.

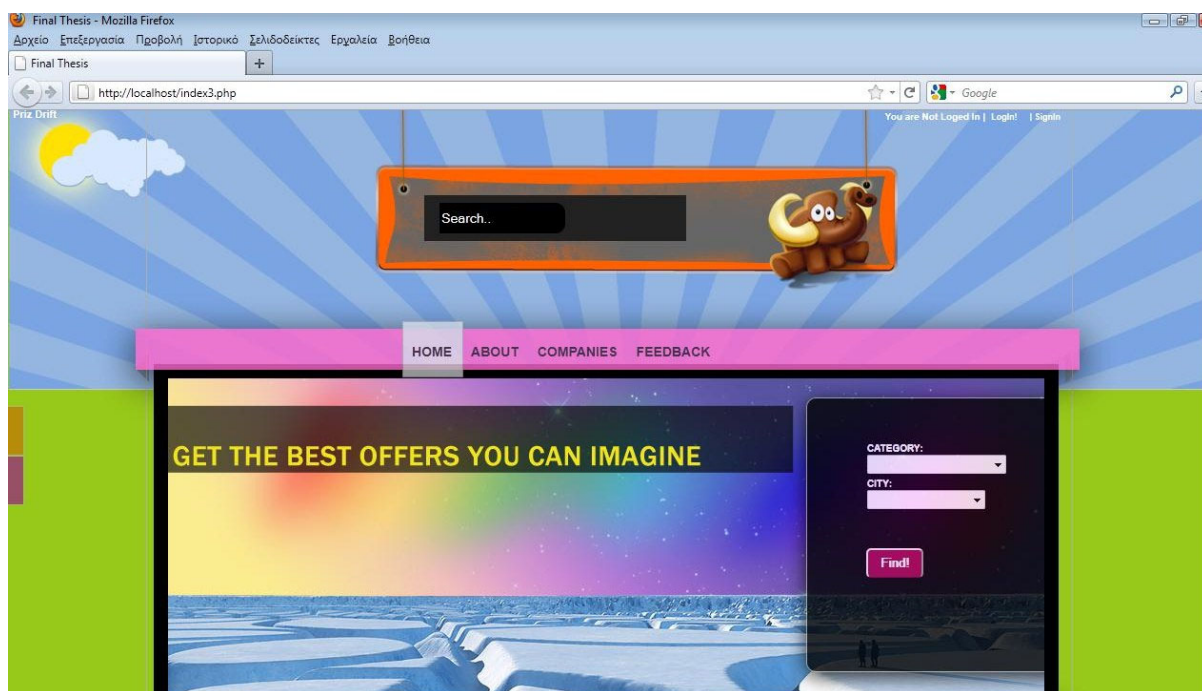
- 1 PHP and MySQL Web Development (3rd Edition) by **Luke Welling and Laura Thomson**. Publisher: Sams; ISBN-10: 0672326728, ISBN-13: 978-0672326721
- 2 Beginning PHP and MySQL 5: From Novice to Professional, Second Edition, by **W. Jason Gilmore**. Publisher: Apress; 2 edition, ISBN-10: 1590595521, ISBN-13:978-1590595527
- 3 MySQL/PHP Database Applications, by **Jay Greenspan and Brad Bulger**. Published by M&T Books, ISBN: 0-7645-3537- 4
- 4 Easy PHP Websites with the Zend Framework, by **W. Jason Gilmore**. Published by W.J. Gilmore, LLC
- 5 jQuery in Action, by **BEAR BIBEAULT and YEHUDA KATZ**. Publisher:Manning Publications Co, ISBN 1-933988-35-5
- 6 jQuery UI 1.6 The User Interface Library for jQuery, by **Dan Wellman**. Publisher: Packt Publishing Ltd. ISBN 978-1-847195-12-8
- 7 Learning jQuery 1.3, by **Jonathan Chaffer Karl Swedberg**. Publisher: Packt Publishing Ltd. ISBN 978-1-847196-70-5
- 8 JavaScript: The Good Parts, by **Douglas Crockford**. Publisher: O'Reilly. ISBN-13: 978-0-596-51774-8
- 9 Ajax: The Definitive Guide, by **Anthony T. Holdener, III**. Publisher: O'Reilly. ISBN-10: 0-596-52838-8, ISBN-13: 978-0-59-652838-6
- 10 Ajax Patterns and Best Practices, by **Christian Gross**. Publisher:APRESS. ISBN-13: 978-1-59059-616-6, ISBN-10 : 1-59059-616-1
- 11 Ajax in Action, by **Dave Crane and Eric Pascarello with Darren James**. Publisher: Manning Publications Co. ISBN 1-932394-61-3
- 12 CSS3 FOR WEB DESIGNERS. **Dan Cederholm**. Publisher: Jeffrey Zeldman. ISBN 978-0-9844425-2-2
- 13 HTML5 and CSS3 Develop with Tomorrow's Standards Today, by **BrianP. Hogan**. Publisher: The Pragmatic Bookshelf. ISBN-10: 1-934356-68-9 ISBN-13: 978-1-934356-68-5
- 14 HTML5 & CSS3 for the Real World, by **Alexis Goldstein, Louis Lazaris, and Estelle Weyl**. Published by SitePoint Pty. Ltd. ISBN 978-0-9808469-0-4
- 15 Ανάπτυξη συστήματος λογισμικού βάσει της μεθοδολογίας ICONIX, **Αλέξανδρος Ν. Χατζηγεωργίου**. Publisher: ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ 2008.

Εκτός βέβαια από τα που αναφέρθηκαν ένα μεγάλο μέρος της προσπάθειας μας στηρίχθηκε σε πηγές του διαδικτύου ορισμένες από τις οποίες σας τις επισυνάπτουμε με σκοπό την περαιτέρω μελέτη τους.

- 16 www.apache.org, σχετικά με τον Apache
- 17 www.php.net, σχετικά με την PHP
- 18 www.mysql.com, σχετικά με την MySQL
- 19 <http://gradients.glrzad.com/>, σχετικά με CSS3
- 20 <http://www.css3.info/>, σχετικά με CSS3
- 21 <http://www.w3.org/>, σχετικά με W3C
- 22 <http://www.w3schools.com/>, σχετικά tutorials με CSS3, HTML5, Ajax, JQuery.
- 23 <http://tutorialzine.com/>, σχετικά tutorials με CSS3, HTML5, Ajax, JQuery.
- 24 <http://net.tutsplus.com/>, σχετικά tutorials με CSS3, HTML5, Ajax, JQuery.
- 25 <http://jquery.com/>, σχετικά με το JQuery.
- 26 <http://www.allbusiness.com/technology/internet-technology-web-development/313058-1.html>
- 27 <http://www.w3schools.com/asp/default.asp>, σχετικά με την ASP
- 28 <http://www.java.com/en/>, σχετικά με την Java
- 29 <http://www.w3schools.com/dhtml/default.asp>, σχετικά με την Dhtml
- 30 <http://www.entheosweb.com/Flash/default.asp>, σχετικά με την Flash
- 31 <http://www.silverlight.net/learn>, σχετικά με την Silverlight
- 32 <http://oreilly.com/web2/archive/what-is-web-20.html>, σχετικά με την Web2.0
- 33 http://en.wikipedia.org/wiki/Unified_Modeling_Language, σχετικά με UML
- 34 <http://www.ambysoft.com/unifiedprocess/agileUP.html>, σχετικά με Agile
- 35 <http://www.w3schools.com/html5/default.asp>, σχετικά με DOM
- 36 <http://www.getfirebug.com/>, σχετικά με firebug
- 37 <http://chrispederick.com/work/web-developer/>
- 38 <http://www.mozilla.org/projects/venkman/>
- 39 <http://www.microsoft.com/downloads/details.aspx?FamilyID=e59c3964-672d-4511-bb3e-2d5e1db91038>
- 40 <http://msdn.microsoft.com/vstudio/express/vwd/>
- 41 <http://www.debugbar.com/>
- 42 <http://www.w3schools.com/css/>, σχετικά με CSS
- 43 <http://www.w3.org/TR/2003/CR-css3-color-20030514/#hsla-color>
- 44 <http://www.css3.info/modules/selector-compat/>
- 45 http://www.deepbluesky.com/blog/-/browser-support-for-css3-and-html5_72/
- 46 http://webfonts.info/wiki/index.php?title=@font-face_browser_support
- 47 <http://www.w3schools.com/xml/>, σχετικά με XML
- 48 <http://www.w3schools.com/xhtml/>, σχετικά με XHTML
- 49 <http://www.w3.org/TR/XMLHttpRequest/>
- 50 <http://www.xfront.com/microformats/AHAH.html>
- 51 <http://www.json.org/>
- 52 <http://www.apache.org/>
- 53 <http://www.iis.net/>
- 54 <http://www.wampserver.com/en/>
- 55 <http://webkitgtk.org/>
- 56 <http://www.onlinemarketing-trends.com/2011/01/browser-wars-chrome-vs-firefox-vs-ie.html>
- 57 http://www.huffingtonpost.com/2011/03/28/google-chrome-10-vs-firefox-4-vs-internet-explorer-9_n_841320.html
- 58 http://news.cnet.com/8301-13554_3-10018838-33.html
- 59 <http://marketshare.hitslink.com/browser-market-share.aspx?spider=1&qprid=0>
- 60 <http://en.wikipedia.org/wiki/Twitter>

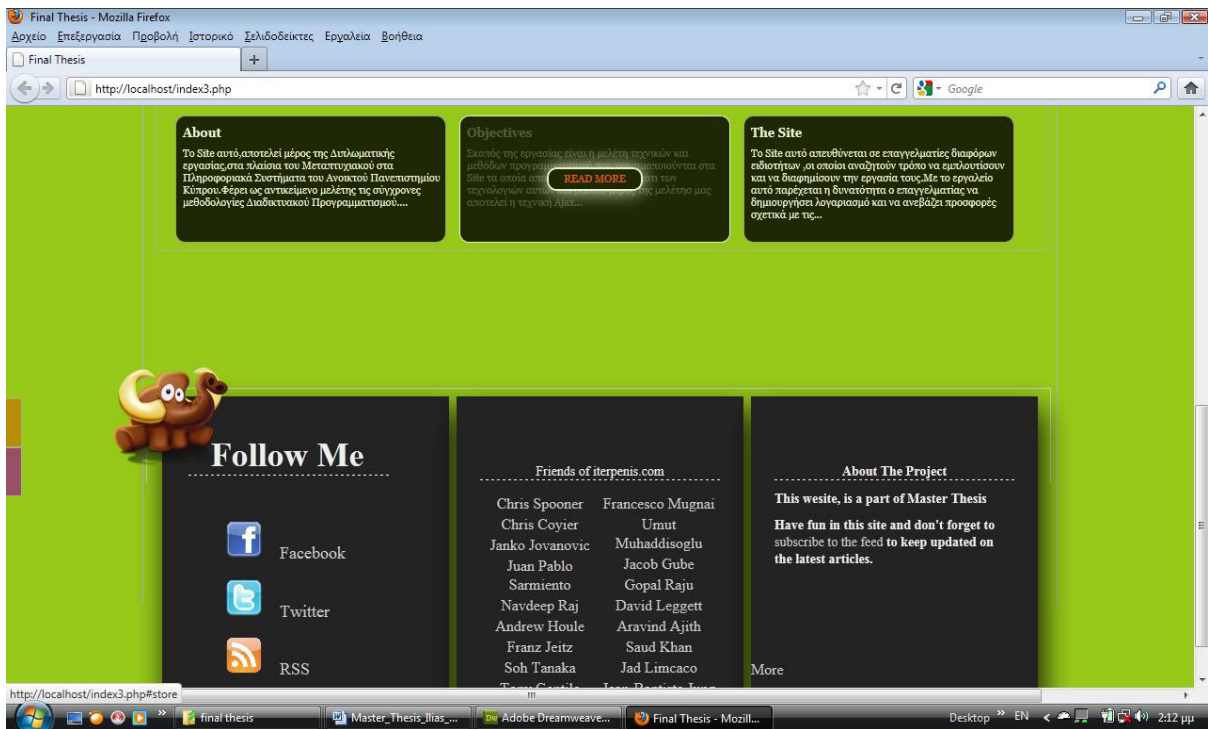
Παράρτημα Α

Στις σελίδες που θα ακολουθήσουν στο συγκεκριμένο παράρτημα θα αποτυπώσουμε μέρος της λειτουργικότητας της εφαρμογής, η οποία κατασκευάστηκε ως κομμάτι της διπλωματικής εργασίας. Θα θέλαμε στο σημείο αυτό να τονίσουμε ότι δεν πρόκειται για μια πλήρη εμπορική εφαρμογή, αφού δεν ήταν αυτός ο σκοπός της διπλωματικής εργασίας. Το αποτέλεσμα της ιστοσελίδας είναι εμφανές στο σχήμα που ακολουθεί.



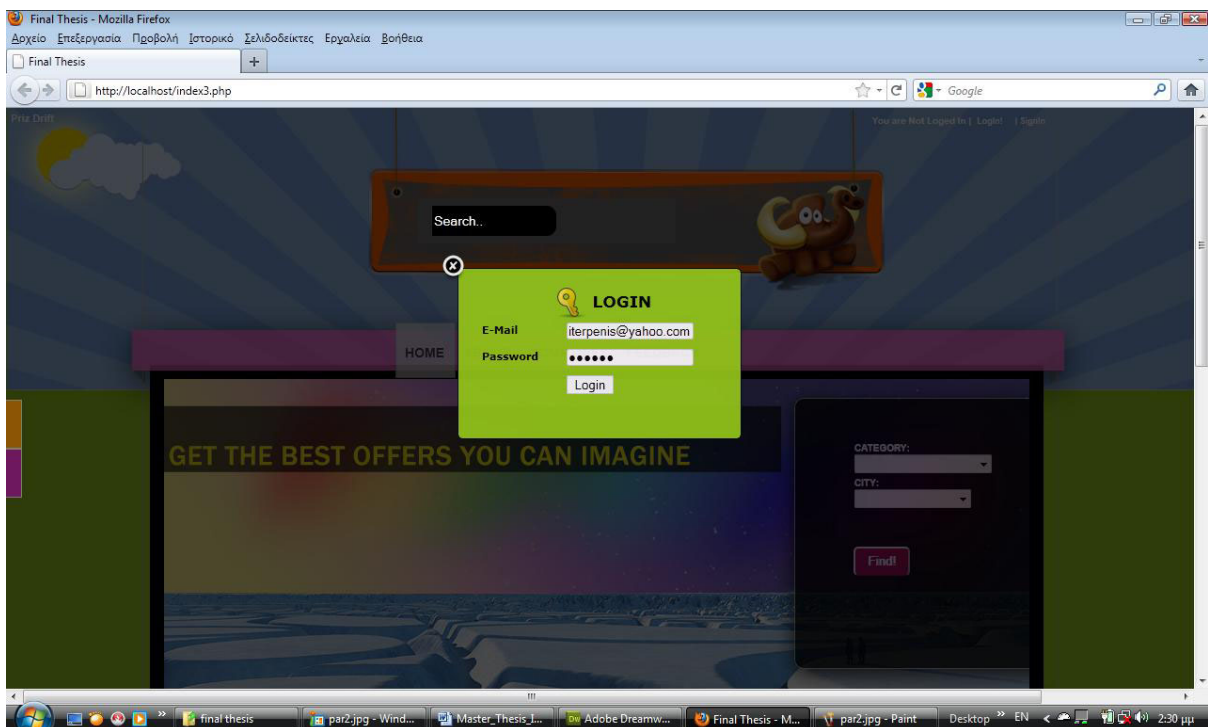
Σχ.1

Είναι φανερό το γεγονός ότι μας δίνεται η δυνατότητα να εκτελέσουμε τις περισσότερες από τις βασικές λειτουργίες που έχει μια σύγχρονη εφαρμογή αυτού του είδους. Όπως για παράδειγμα να κάνουμε αναζήτηση η οποία περιγράφηκε λεπτομερέστατα στο Κεφάλαιο 5 (σελ. 97). Μας δίνεται επίσης η δυνατότητα να κάνουμε εγγραφή ως χρήστες του ιστοχώρου αυτού, ο τρόπος που το πετυχαίνουμε αυτό αναλύθηκε στο Κεφάλαιο 3 (σελ.55). Το κάτω μέρος της ιστοσελίδας φαίνεται στο σχήμα που ακολουθεί. Δεν πρόκειται για κάτι ενδιαφέρον με την πρώτη ματιά, όμως τόσο σχ.1 όσο και στο σχ.2 όλα τα γραφικά κατασκευάστηκαν με την χρήση του CSS3^[13-14] και του FontFace^[46].

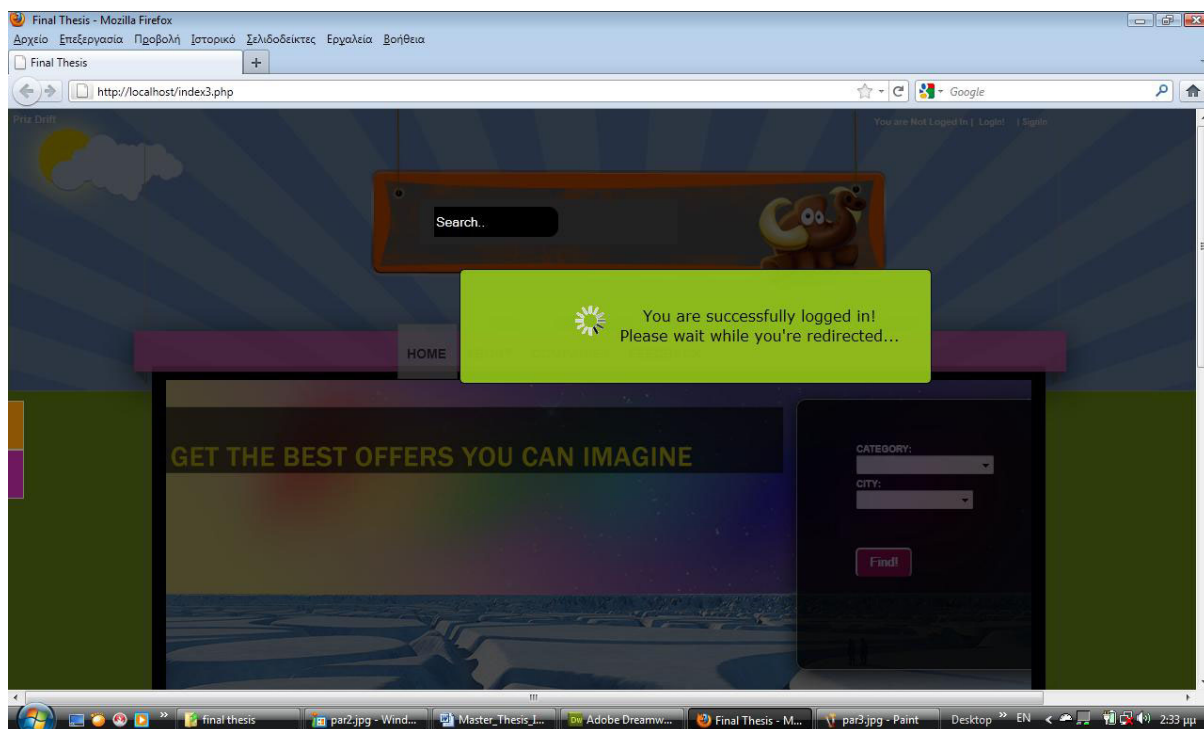


Σχ.2 Σκιάσεις και γραφικά με την χρήση CSS3

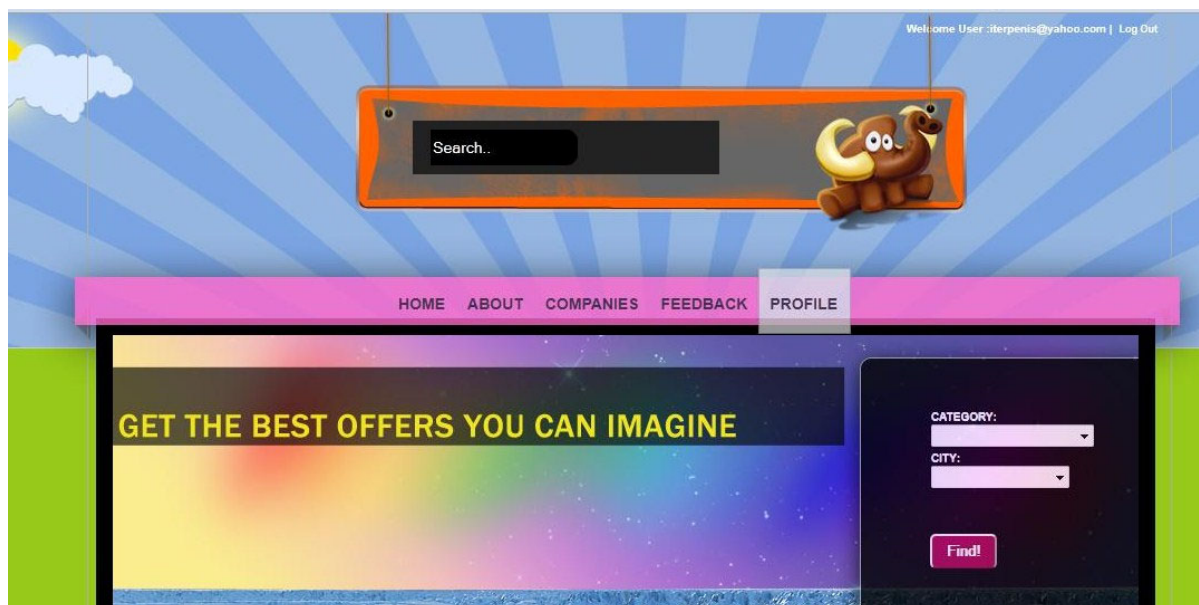
Ο χρήστης σε περίπτωση που έχει κάποιο λογαριασμό θα μπορεί να κάνει εγγραφή και να αποκτήσει πρόσβαση στον χώρο στον οποίο θα διαμορφώσει το προσωπικό του Profil. Με την εγγραφή του, δίνεται η δυνατότητα να δημιουργεί τις προσφορές του. Ακολουθούν στιγμιότυπα οθόνης με τα οποία φαίνεται η διαδικασίες που περιγράφηκαν προηγουμένως.



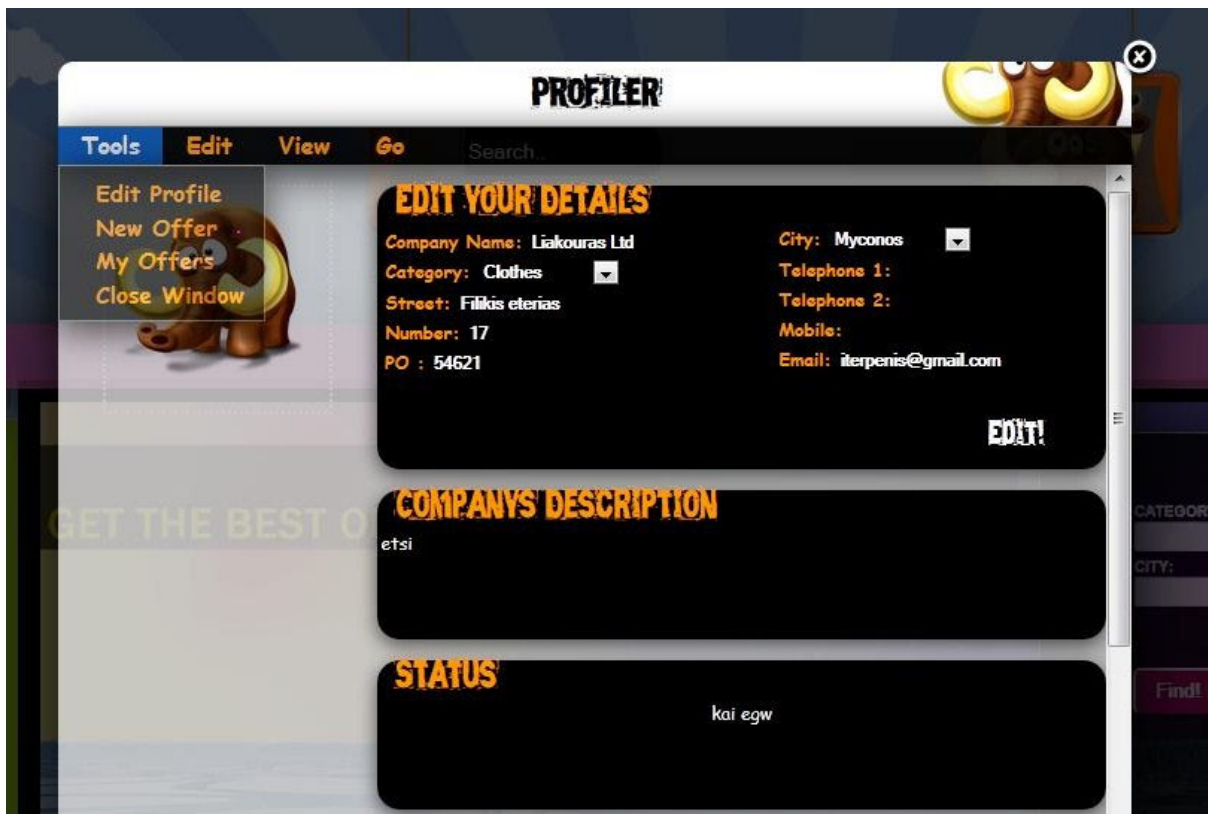
Σχ.3 Εγγραφή



Σχ.4 Επιβεβαίωση εγγραφής



Σχ.5 Ενεργοποίηση επιλογής PROFILE μετά την εγγραφή



Σχ.6 PROFILE με δυνατότητα επεξεργασίας του, και χρήση του Fontface

Με το άνοιγμα του προσωπικού του χώρου, ο χρήστης θα είναι σε θέση να επεξεργάζεται τα στοιχεία του να αλλάζει το Status του, και να περιγράφει την εταιρεία του. Όπως και έχουμε αναφέρει η ιστοσελίδα αυτή είναι μέρος των σημειώσεων οι οποίες κάνουν λόγο για δυναμικές ιστοσελίδες. Στο συγκεκριμένο σημείο μπορούμε να κάνουμε τροποποιήσεις των στοιχείων και επαλήθευση τους χωρίς καμιά ανανέωση της σελίδας μας. Για παράδειγμα με ένα κλικ πάνω στο πεδίο που γίνεται περιγραφή της εταιρείας θα μπορούσαμε να γράψουμε “ilias final thesis”.



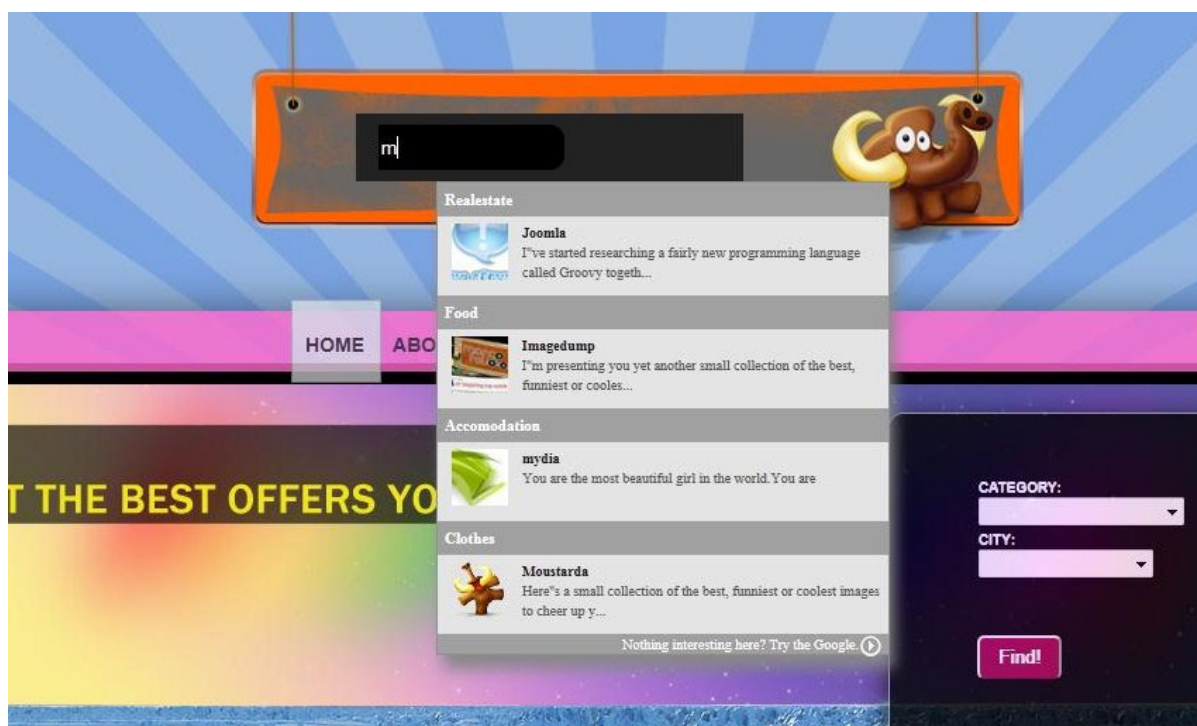
Σχ.7 Κλικ πάνω στο πεδίο που γίνεται περιγραφή της εταιρείας



Σχ.8 Αλλαγή της περιγραφής

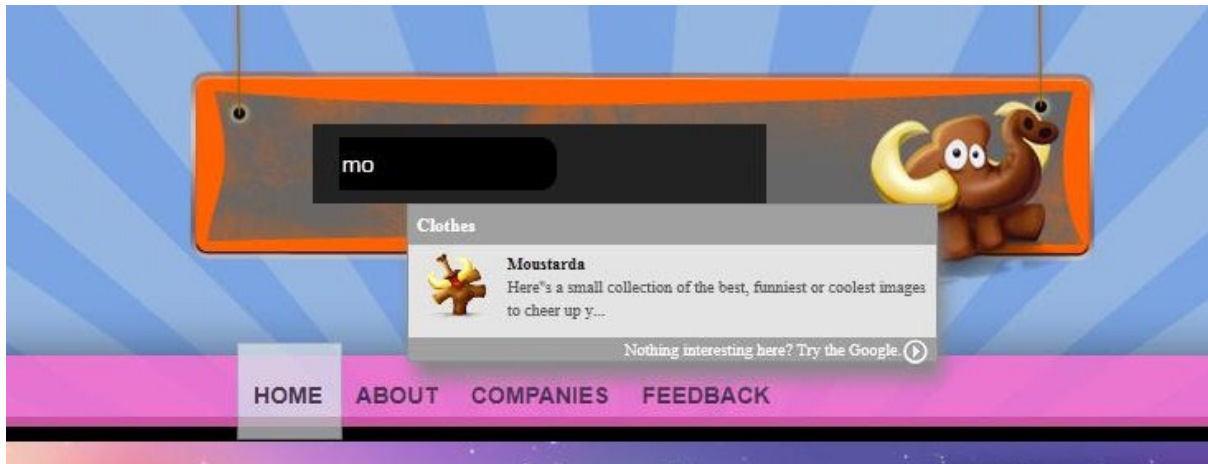
Με τον ίδιο τρόπο μπορούμε να αλλάξουμε και τα άλλα πεδία του Profile, για λόγους οικονομίας χώρου θα σας προτείνουμε να το δοκιμάσετε μόνοι σας.

Σε περίπτωση που δεν είστε κάποια εταιρεία και ενδιαφέρεστε να βρείτε κάποια προσφορά έχετε δύο τρόπους στην διάθεση σας για να το επιτύχετε. Ο πρώτος είναι μέσω του κεντρικού Search γράφοντας την λέξη που θέλετε, αποτέλεσμα αυτού είναι να σας παρουσιάζονται κάποιες προτάσεις από το σύστημα σχετικά με την αναζήτηση σας. Ας υποθέσουμε ότι ψάχνουμε την εταιρεία με την επωνυμία “Moustarda”. Όπως φαίνεται το σύστημα αρχικά παρουσιάζει όλα τα λύματα που φέρουν μέσα στην περιγραφή τους το γράμμα “m”.



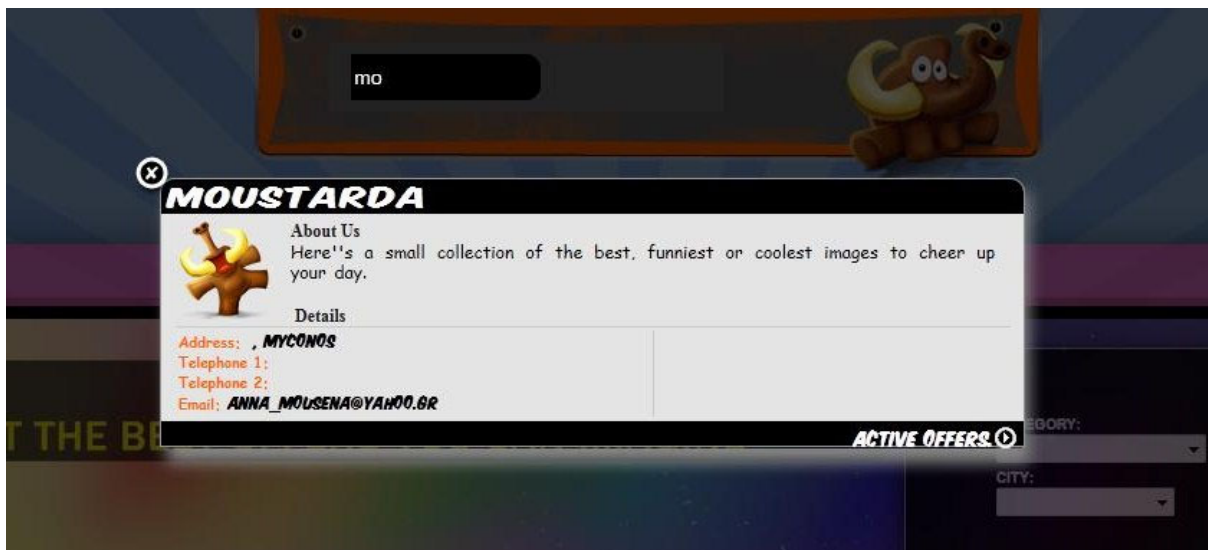
Σχ.9 Αναζήτηση της εταιρείας με την επωνυμία “Moustarda”

Συνεχίζοντας την πληκτρολόγηση επιλέγουμε το γράμμα “ο” το σύστημα φιλτράρει τα προηγούμενα αποτελέσματα ξεχωρίζοντας αυτά που έχουν συνεχόμενα τα γράμματα “mo”, καταλήγοντας στο σχήμα 10.



Σχ.10 Αποτέλεσμα πληκτρολόγησης “mo”

Εάν το αποτέλεσμα είναι το επιθυμητό, ο χρήστης μπορεί να το επιλέξει κάνοντας κλικ επάνω του. Αποτέλεσμα της ενέργειας αυτής είναι να εμφανίζονται τα χαρακτηριστικά της εταιρείας αυτής όπως φαίνεται στο σχήμα που ακολουθεί.



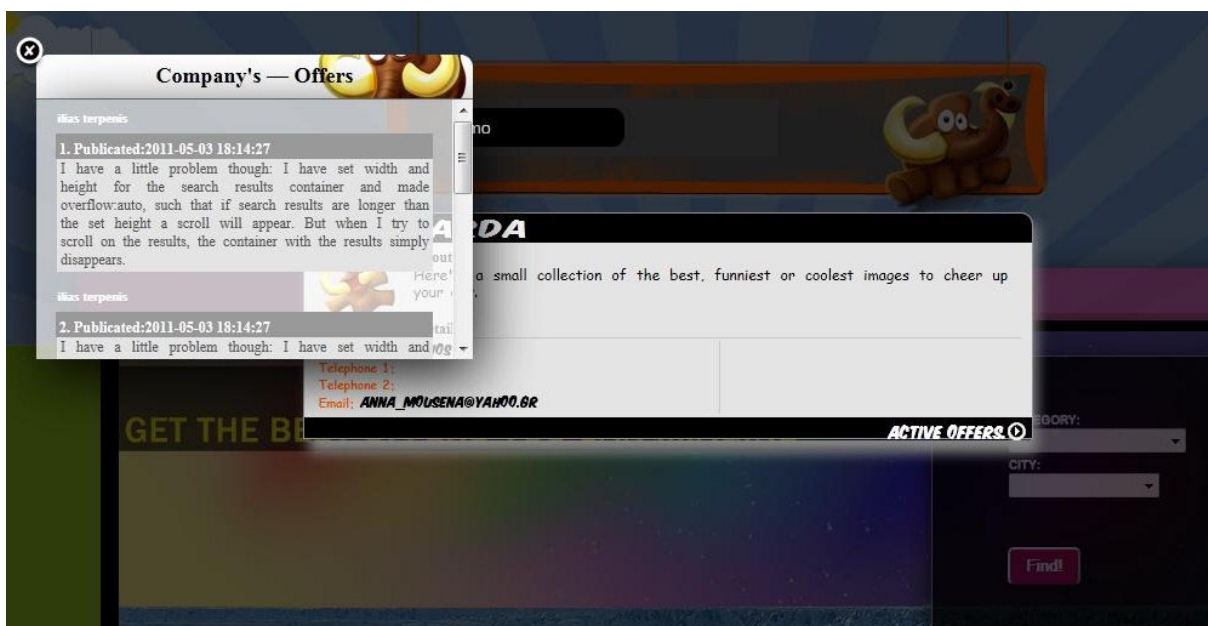
Σχ.11 Χαρακτηριστικά της εταιρείας

Σε περίπτωση που ενδιαφερόμαστε για τις προσφορές της εταιρείας αυτής, κάνουμε κλικ στην επιλογή “ACTIVE OFFERS”, και εμφανίζονται οι ενεργές προσφορές της εταιρείας. Εάν δεν υπάρχουν εμφανίζεται σχετικό μήνυμα.



Σχ.12 προσφορές εταιρείας

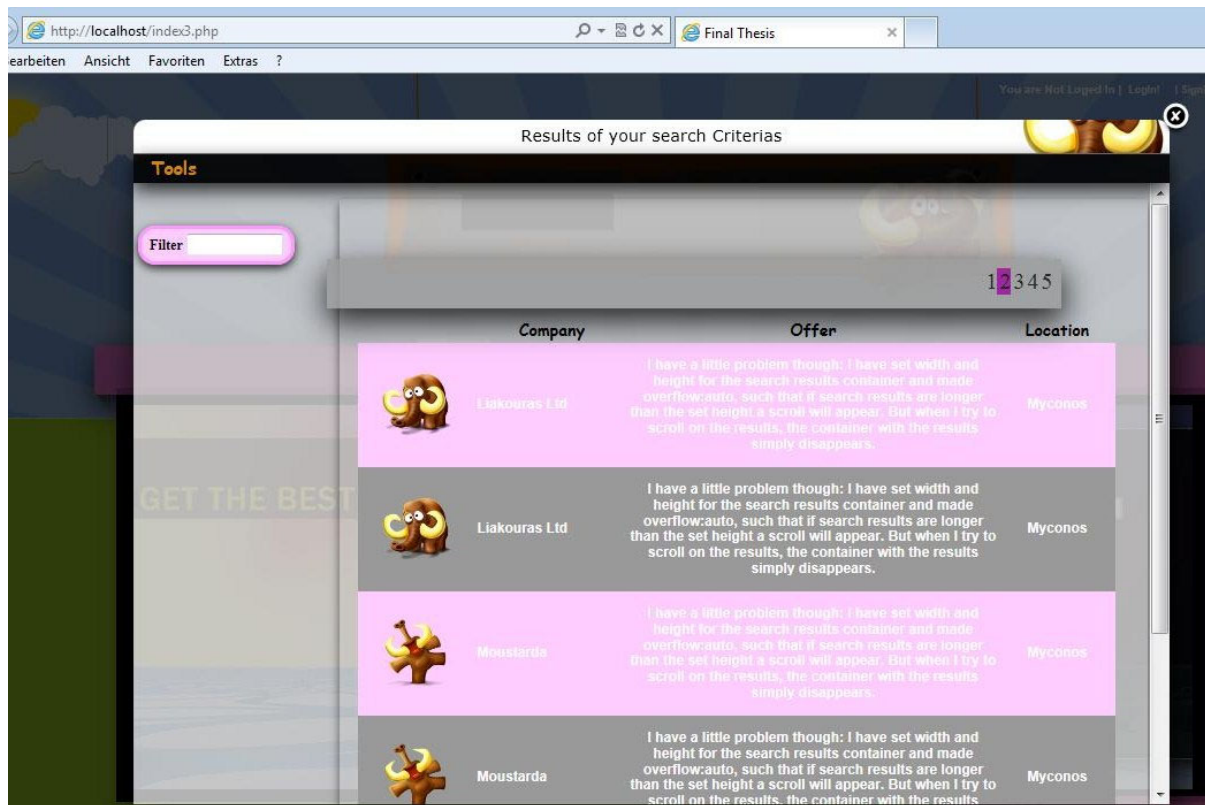
Είναι ενδιαφέρον να τονίσουμε στο σημείο αυτό, τι είναι αυτό που κάνει την διαφορά. Όπως φαίνεται στο σχήμα 12 οι προσφορές εμφανίζονται σε χωριστό παράθυρο, με αποτέλεσμα στα δεξιά να βλέπουμε τα χαρακτηριστικά της εταιρείας και στα αριστερά τις προσφορές της. Όμως αυτό που θα παρατηρούσε ένας προσεκτικός αναγνώστης είναι το γεγονός ότι το παράθυρο των προσφορών επικαλύπτεται από το παράθυρο της εταιρείας. Μήπως πρόκειται για κάποια αστοχία του κώδικα. Και απαντάμε πως όχι. Για να δώ τις προσφορές αρκεί να φέρουμε το ποντίκι πάνω από το παράθυρο των προσφορών.



Σχ.13 προσφορές

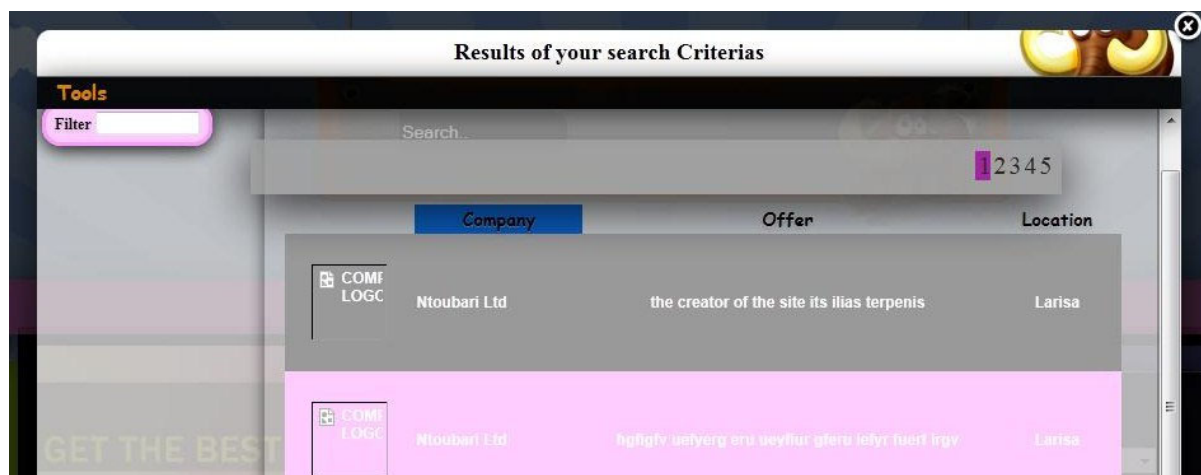
Δεν πρόκειται για κάτι μαγικό απλά είναι μία από τις δυνατότητες τις οποίες μας δίνει το CSS3.

Ας υποθέσουμε ότι θέλουμε να αναζητήσουμε προσφορές με βάση ορισμένα κριτήρια, για παράδειγμα συγκεκριμένη κατηγορία και συγκεκριμένη πόλη. Εμείς για τις ανάγκες του παραδείγματος δεν θα επιλέξουμε ούτε πόλη, ούτε και κατηγορία απλά θα πιάσουμε το πλήκτρο “Find”.



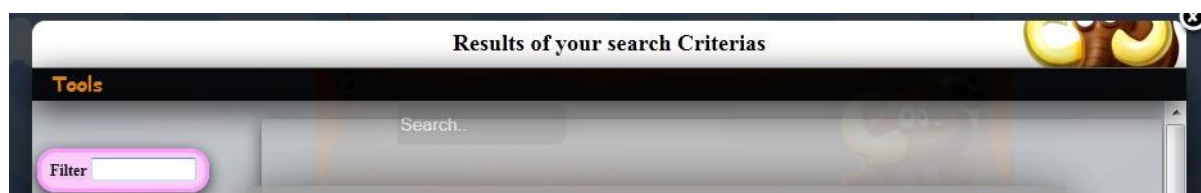
Σχ.14 Αποτελέσματα αναζήτησης χωρίς κριτήρια

Στο σχήμα 14 φαίνεται το αποτέλεσμα της ενέργειας αυτής. Παρατηρούμε ότι έχουμε 5 σελίδες αποτελεσμάτων, στις οποίες μπορούμε να μεταβαίνουμε κάνοντας κλικ στον αριθμό της κάθε μίας. Σε περίπτωση που θέλουμε να τοποθετήσουμε σε αλφαβητική σειρά τα αποτελέσματα ως προς το όνομα της εταιρείας, αρκεί να κάνουμε κλικ πάνω στον τίτλο της στήλης με το όνομα “Company”.



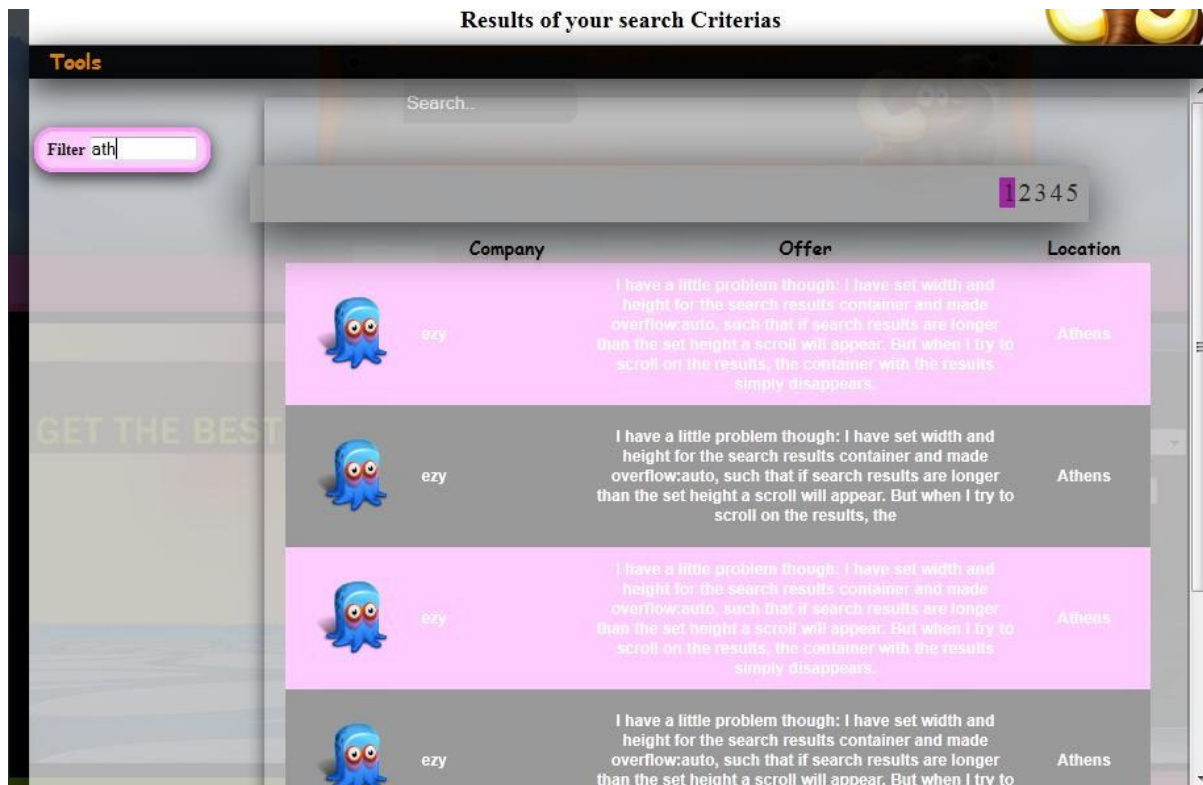
Σχ.15 Αύξουσα και φθίνουσα τοποθέτηση

Τι συμβαίνει σε περίπτωση που τα αποτελέσματα είναι χιλιάδες ή ακόμα δεκάδες χιλιάδες; Στην περίπτωση αυτή είναι αδύνατο να δεις τις προσφορές χωριστά με την σειρά, μέχρι να βρεις αυτή που σου ταιριάζει. Για τον λόγο αυτό δημιουργήσαμε ένα εργαλείο το οποίο βρίσκεται στα αριστερά της σελίδας με τα αποτελέσματα και ονομάζεται “filter”.



Σχ.16 φίλτρο αποτελεσμάτων

Καλά μαντέψατε λοιπόν ότι δεν πρόκειται για κάποιο φίλτρο ενός επίδοξου μάγου. Αποτελεί απόδειξη των δυνατοτήτων που μας δίνει το Ajax, στο να επεξεργαζόμαστε τα δεδομένα στην πλευρά του χρήστη. Έστω για παράδειγμα ότι θέλουμε προσφορές μόνο στην Αθήνα. Πληκτρολογώντας κάθε γράμμα χωριστά το εργαλείο αυτό μας φέρνει στην πρώτη σελίδα όλες τις προσφορές οι οποίες κάπου στο τίτλο της εταιρείας, η στην περιγραφή της προσφοράς ή και της πόλης έχουν τα συγκεκριμένα γράμματα. Δεν διαγράφονται οι προσφορές οι οποίες δεν πληρούν τα κριτήρια αναζήτησης, απλά τοποθετούνται σε πιο πίσω σελίδες. Αν πληκτρολογήσουμε την λέξη “ath” θα πάρουμε το αποτέλεσμα που φαίνεται στο πιο κάτω σχήμα.



Σχ.17 φίλτρο αποτελεσμάτων για “ath”

Συμπέρασμα

Στο παράρτημα αυτό προσπαθήσαμε να σας περιγράψουμε την λειτουργικότητα της εφαρμογής που συνοδεύει την διπλωματική εργασία. Τονίσαμε σημεία της λειτουργικότητας τα οποία είναι συνδυασμός των τεχνικών που περιγράφηκαν στα προηγούμενα κεφάλαια. Σκοπός μας δεν είναι να δημιουργήσουμε εντυπώσεις σε σχέση με αυτά που φτιάξαμε, αλλά να δείξουμε ότι με σωστή χρήση των τεχνικών που υπάρχουν στις μέρες μας και με δημιουργική φαντασία, μπορούμε να πετύχουμε εφαρμογές με δυνατότητες εντυπωσιακές.

Παράρτημα Β

Αποφασίσαμε να συμπεριλάβουμε το παράρτημα αυτό για να σας διευκολύνουμε κατά την περιήγηση σας στα αρχεία του κώδικα, τα οποία αποτελούν μέρος της διπλωματικής εργασίας.

1. `index3.php`, αποτελεί το κύριο αρχείο της εφαρμογής το οποίο καθορίζεται και το Layout της.
2. `index3.css`, το βασικό αρχείο CSS της εφαρμογής με το οποίο δίνονται τα διάφορα χαρακτηριστικά, ως προς την εμφάνιση, των περισσότερων αντικειμένων της εφαρμογής.
3. `slide.css`, δευτερεύον αρχείο CSS με το οποίο διαμορφώνονται τα χαρακτηριστικά, ως προς την εμφάνιση, του κεντρικού slider.
4. `iliasterpready.js`, αποτελεί αρχείο JavaScript στο οποίο είναι γραμμένο ένα μεγάλο κομμάτι του JQuery κώδικα.
5. `ilisidenav.js`, κώδικας JQuery σχετικός με το αριστερό μενού πλοήγησης.
6. `scriptilias.js`, κώδικας JQuery σχετικός με Ajax επικοινωνία.
7. `box.js`, κώδικας JQuery με τον οποίον καθορίζουμε την συμπεριφορά των πλαισίων στο κάτω μέρος της σελίδας μας, στα οποία όταν τοποθετούμε το ποντίκι εμφανίζεται ένα πλήκτρο.
8. `foundres.js`, αρχείο JQuery.
9. `do-compinf.php`, αρχείο PHP με το οποίο παίρνουμε πληροφορίες για μία εταιρεία εάν υπάρχει στην βάση δεδομένων.
10. `do-contact.php`, αρχείο PHP που χρησιμεύει για την επικοινωνία του χρήστη.
11. `do-login1.php`, αρχείο PHP που επιτρέπει στον χρήστη να κάνει εγγραφή.
12. `do-signin.php`, αρχείο PHP που επιτρέπει σε νέους χρήστες να εγγραφούν στις υπηρεσίες της εφαρμογής.
13. `do-thecatoffer.php`, αρχείο PHP που εμφανίζει τις προσφορές.
14. `do-thedetails.php`, αρχείο PHP με το οποίο παίρνουμε πληροφορίες για μία εταιρεία από την βάση δεδομένων.
15. `do-thenewoffers.php`, αρχείο PHP με το οποίο ο χρήστης τοποθετεί μία νέα προσφορά.
16. `do-theoffers.php`, αρχείο PHP με το οποίο εμφανίζονται οι προσφορές μιας εταιρείας.
17. `do-theprof.php`, αρχείο PHP με το οποίο εμφανίζεται το προφίλ μιας εταιρείας έτοιμο προς επεξεργασία.
18. `do-thesearch.php`, αρχείο PHP με το οποίο εμφανίζουμε τα αποτελέσματα κατά το κεντρικό search με βάση τα λήμματα.

19. `logout.php`, αρχείο PHP με το οποίο κάνουμε αποσύνδεση.
20. `offerpub.php`, αναζήτηση εταιρείας.
21. `check1.php`, ελέγχει τον χρήστη και κάνει τα στοιχεία του session.
22. `save_edit.php`, αρχείο PHP για τον έλεγχο του ονόματος χρήστη.