



Τεχνολογικό Εκπαιδευτικό Ίδρυμα  
Κρήτης

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Μηχανικών Πληροφορικής

## Πτυχιακή εργασία

Πλατφόρμα αρχειοθέτησης εργασιών

Κωνσταντίνος Θεοφίλης (ΑΜ: 2195)

Επιβλέπων Καθηγητής: Σπύρος Παναγιωτάκης  
Ημερομηνία Παρουσίασης: 28/3/2014

## **Ευχαριστίες**

Ευχαριστώ θερμά τον Κώστα Καπετανάκη, που παρόλο τον μεγάλο φόρτο εργασία του, ανέλαβε να με βοηθήσει με την πτυχιακή εργασία.

Επίσης ευχαριστώ το φίλο μου Δημήτρη που έτρεξε για τα διαδικαστικά...

## **Abstract**

The dissertation is on the development of an integrated content management system (CMS) for the scientific output of the multimedia lab. The CMS will be comprised of specialized areas for document viewing, advanced search and result filtering. Further, it will make use of the ability to categorize documents based on keywords assigned to the documents and this categorization will be exploited to allow produce a "Tag-Cloud" view of the keywords used. There will be a dedicated "admin" application area for the insertion and content-management of the material and associated data. To test and verify the application we intend to use actual scientific content stemming from the media lab.

## Σύνοψη

Για την πτυχιακή εργασία θα δημιουργηθεί ένα ολοκληρωμένο σύστημα διαχείρισης περιεχομένου, ειδικά αναπτυγμένο για την προβολή και αναζήτηση εγγράφων σχετικών με το επιστημονικό περιεχόμενο του εργαστηρίου πολυμέσων. Θα περιλαμβάνει ειδικές περιοχές παρουσίασης των εγγράφων, σύνθετη αναζήτηση και επισκόπηση των αποτελεσμάτων βασισμένη στα στοιχεία του περιεχομένου. Επίσης θα υπάρχει η δυνατότητα κατηγοριοποίησης των εγγράφων με βάση τις λέξεις κλειδιά που αυτά περιέχουν και η εμφάνιση όλων των λέξεων σε μορφή Tag-Cloud. Η εφαρμογή θα υποστηρίζεται πλήρως από περιοχή διαχείρισης όπου θα καταχωρούνται τα έγγραφα και τα στοιχεία που τα συνοδεύουν. Για να επιβεβαιωθεί η σωστή λειτουργία της εφαρμογής, αυτή θα δοκιμαστεί με πραγματικό υλικό.

# Περιεχόμενα

Ευχαριστίες .....	ii
Abstract .....	iii
Σύνοψη.....	iv
Περιεχόμενα .....	v
Πίνακας εικόνων.....	vii
1. Εισαγωγή.....	1
1.1 Περίληψη.....	1
1.2 Κίνητρο για τη Διεξαγωγή της Εργασίας .....	1
1.3 Σκοπός και Στόχοι Εργασίας .....	1
1.4 Δομή της εργασίας.....	2
2. Μεθοδολογία ανάπτυξης ιστοσελίδων.....	3
2.1 Στατικές ιστοσελίδες .....	3
2.2 Δυναμικές ιστοσελίδες.....	3
2.2.1 Εμφάνιση της ιστοσελίδας.....	3
2.2.2 Front End Framework .....	7
2.2.3 Responsive Web Design.....	8
2.2.4 Δυναμικό κομμάτι της ιστοσελίδας .....	10
2.2.5 Php Frameworks.....	10
3. Τεχνολογίες & Γλώσσες Προγραμματισμού .....	13
3.1 Html5 & Css3.....	13
3.1.1 Html5 .....	13
3.1.2 CSS3 .....	15
3.2 PHP(PHP: Hypertext Preprocessor).....	18
3.3 SQL (Structured Query Language).....	18
3.4 JavaScript.....	19
3.5 jQuery .....	20
3.6 WebGL .....	21
4. Εργαλεία Λογισμικού .....	23
4.1 Bootstrap .....	23
4.2 Web Server Apache .....	23
4.3 WAMP.....	24
4.4 CakePHP.....	24
5. Σχέδιο Δράσης για την εκπόνηση της εργασίας.....	27

5.1 Επιλογή τρόπου σχεδίασης της εμφάνισης της ιστοσελίδας.....	27
5.2 Επιλογή τρόπου δημιουργίας δυναμικού μέρους.....	29
6. Ανάπτυξη του συστήματος.....	31
6.1 Απαιτήσεις Συστήματος .....	31
6.2 Υλοποίηση .....	31
6.2.1 Σχεδίαση .....	31
6.2.2 Δημιουργία αρχείων Html & Css.....	32
6.2.3 Εγκατάσταση του Server.....	33
6.2.4 Εγκατάσταση του Cakephp Framework .....	34
6.2.5 Υλοποίηση της βάσης δεδομένων.....	35
6.2.6 Δημιουργία σελίδων εφαρμογής.....	38
6.2.7 Δημιουργία περιοχής διαχείρισης .....	56
6.2.8 Δημιουργία τρισδιάστατης βιβλιοθήκης.....	68
7. Συμπεράσματα .....	76
7.1 Συνολική αποτίμηση .....	76
7.2 Επεκτάσεις .....	76
8. Βιβλιογραφία.....	77

## Πίνακας εικόνων

Εικόνα 1 html.....	4
Εικόνα 2 CSS .....	5
Εικόνα 3 Ιεραρχία στοιχείων στο DOM.....	6
Εικόνα 4 Δείγμα fixed layout.....	7
Εικόνα 5 Δείγμα fluid layout.....	7
Εικόνα 6 Μέγιστος αριθμός στηλών ανά συσκευή.....	8
Εικόνα 7 Εμφάνιση της ίδιας ιστοσελίδας σε διαφορετικές συσκευές.....	8
Εικόνα 8 Δείγμα Media Query για οθόνες μέχρι 480pixels .....	9
Εικόνα 9 Server side responsive design .....	9
Εικόνα 10 Στατιστικά χρήσης γλωσσών για το έτος 2012 .....	10
Εικόνα 11 Λογότυπα των πιο δημοφιλών php frameworks .....	12
Εικόνα 12 Δείκτες σύγκρισης δημοφιλίας με βάση τις δουλειές.....	12
Εικόνα 13 Λογότυπο της Html5.....	13
Εικόνα 14 Διαφορές Html5 με τις προηγούμενες εκδόσεις .....	14
Εικόνα 15 Λογότυπο CSS3 .....	16
Εικόνα 16 Λογότυπο php .....	18
Εικόνα 17 Λογότυπο Mysql .....	19
Εικόνα 18 Λογότυπο jQuery .....	20
Εικόνα 19 Λογότυπο WebGL.....	22
Εικόνα 20 Λογότυπο bootstrap.....	23
Εικόνα 21 Λογότυπο Apache server.....	23
Εικόνα 22 Μοντέλο client - server .....	24
Εικόνα 23 Λογότυπο wamp .....	24
Εικόνα 24 Λογότυπο Cakephp .....	25
Εικόνα 25 Πλεονεκτήματα CakePhp .....	25
Εικόνα 26 Λογότυπο Photoshop.....	27
Εικόνα 27 Μεταφορά psd σε html .....	28
Εικόνα 28 Το λογότυπο της εφαρμογής μας.....	31
Εικόνα 29 Η σχεδίαση της αρχικής σελίδας της εφαρμογής στο Photoshop .....	32
Εικόνα 30 Σχεδίαση σελίδας αναζήτησης σε Html και Css.....	33
Εικόνα 31 Εικόνα μετά την επιτυχή εγκατάσταση του wamp.....	33
Εικόνα 32 Αρχική σελίδα phpMyAdmin .....	34
Εικόνα 33 Αρχική εικόνα Cakephp .....	35
Εικόνα 34 Μοντέλο οντοτήτων συσχετίσεων .....	35
Εικόνα 35 Κονσόλα εντολών .....	38
Εικόνα 36 προεπιλεγμένη εμφάνιση σελίδας.....	39
Εικόνα 37 Εμφάνιση του header .....	40
Εικόνα 38 Εμφάνιση footer.....	42
Εικόνα 39 Εμφάνιση αρχικής σελίδας .....	43
Εικόνα 40 Εμφάνιση αρχικής σελίδας σε μεγάλη οθόνη .....	44
Εικόνα 41 Εμφάνιση σε συσκευή tablet.....	44
Εικόνα 42 Εμφάνιση σε συσκευή κινητού .....	44
Εικόνα 43 Εμφάνιση σελίδας about.....	45

Εικόνα 44 Σελίδα αναζήτησης.....	50
Εικόνα 45 Εμφάνιση σελίδας άρθρου .....	53
Εικόνα 46 Εμφάνιση σελίδας συντάκτη.....	54
Εικόνα 47 Εμφάνιση σελίδας κατηγορίας .....	55
Εικόνα 48 Φόρμα εισαγωγής χρήστη .....	57
Εικόνα 49 Φόρμα εισόδου διαχειριστή.....	58
Εικόνα 50 Επεξεργασία χρήστη .....	59
Εικόνα 51 Λίστα χρηστών.....	59
Εικόνα 52 Εισαγωγή συντάκτη .....	60
Εικόνα 53 Επεξεργασία συντάκτη.....	61
Εικόνα 54 Λίστα συντακτών .....	61
Εικόνα 55 Προσθήκη νέου άρθρου .....	63
Εικόνα 56 Σελίδα επεξεργασίας άρθρου. ....	67
Εικόνα 57 Σελίδα εμφάνισης λίστας άρθρων .....	68
Εικόνα 58 Τρισδιάστατη βιβλιοθήκη.....	69
Εικόνα 59 Τρισδιάστατη βιβλιοθήκη με επιλεγμένο βιβλίο.....	70
Εικόνα 60 Httptrack website copier.....	71
Εικόνα 61 Βιβλιοθήκη .....	72



# 1. Εισαγωγή

## 1.1 Περίληψη

Σε αυτή την πτυχιακή εργασία παρουσιάζεται αναλυτικά η δημιουργία και η χρήση μιας δυναμικής ιστοσελίδας που θα φιλοξενεί και θα αρχειοθετεί τα ψηφιακά αρχεία του εργαστηρίου πολυμέσων. Για να διευκολυνθεί ο χρήστης στην αναζήτηση των αρχείων, αναπτύχθηκαν δυο διαφορετικές προβολές του περιεχομένου, η μια είναι η κλασσική διαδικασία αναζήτησης σε μια βάση δεδομένων, ενώ η δεύτερη δημιουργεί μια τρισδιάστατη απεικόνιση μιας κυκλικής βιβλιοθήκης.

Για να υλοποιηθεί αυτή η διαδικτυακή εφαρμογή, χρησιμοποιήθηκαν διάφορες τεχνολογίες, γλώσσες προγραμματισμού και έτοιμα κομμάτια λογισμικού (frameworks) που βοηθούν στην γρηγορότερη υλοποίηση. Για το front-end κομμάτι της εφαρμογής χρησιμοποιήθηκε το bootstrap framework, που περιέχει κάποια έτοιμα κομμάτια HTML και CSS, ενώ για να βελτιωθεί η ευκολία της αναζήτησης χρησιμοποιήθηκε η τεχνολογία jQuery που είναι βασισμένη στην scripting γλώσσα Javascript. Για το back-end που κάνει την εφαρμογή μας δυναμική, χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP σε συνεργασία με το CakePHP framework. Για την βάση δεδομένων έγινε χρήση του MySQL και όλα αυτά στήθηκαν πάνω στον Apache Web Server που εγκαταστάθηκε με την χρήση του λογισμικού Wamp που είναι ένα ολοκληρωμένο πακέτο και περιέχει ότι χρειάζεται για την σωστή λειτουργία μιας ιστοσελίδας όπως αυτή που κατασκευάσαμε.

Τέλος για την κατασκευή της τρισδιάστατης βιβλιοθήκης χρησιμοποιήθηκε η εφαρμογή 3D bookcase που είναι βασισμένη στην τεχνολογία WebGL και μετά από επεξεργασία του κώδικά της μετατράπηκε ώστε να παρουσιάζει το δυναμικά μεταβλητό περιεχόμενο της σελίδας μας.

## 1.2 Κίνητρο για τη Διεξαγωγή της Εργασίας

Ο κύριος λόγος για την διεξαγωγή αυτής της πτυχιακής εργασίας είναι ξεκάθαρα πρακτικός και αφορά την ανάγκη του εργαστηρίου πολυμέσων να συγκεντρώνει κάπου όλα τα χρήσιμα αρχεία όπως δημοσιεύσεις και άρθρα που προκύπτουν από την λειτουργία του. Για αυτό το λόγο δημιουργήθηκε και αυτή η ιστοσελίδα που συγκεντρώνει όλο αυτό το υλικό.

## 1.3 Σκοπός και Στόχοι Εργασίας

Σκοπός της εργασίας είναι η διευκόλυνση τον τελικού χρήστη, είτε στο να βρει μια συγκεκριμένη δημοσίευση που πιθανόν ψάχνει (με την χρήση κάποιων φίλτρων ή λέξεων κλειδιά), είτε να προβάλει το σύνολο των δημοσιεύσεων, των συγγραφέων κλπ που περιέχονται στην εφαρμογή.

Επίσης αυτή η πτυχιακή εργασία αποσκοπεί στο να βελτιωθεί η εξοικείωση του γράφοντος με τις χρησιμοποιούμενες τεχνολογίες και γλώσσες προγραμματισμού που χρησιμοποιούνται, αλλά και με τη σωστή δόμηση και παρουσίαση της διαδικασίας δημιουργίας της εφαρμογής με αναλυτικά βήματα.

## 1.4 Δομή της εργασίας

Στο δεύτερο κεφάλαιο γίνεται η κατηγοριοποίηση των ιστοσελίδων σε στατικές και δυναμικές και στη συνέχεια περιγράφεται αναλυτικά η μέθοδος ανάπτυξης μιας δυναμικής ιστοσελίδας σε δυο μέρη, αυτό της εμφάνισης και το δυναμικό μέρος.

Στο τρίτο κεφάλαιο περιγράφονται αναλυτικά οι τεχνολογίες και οι γλώσσες προγραμματισμού, ενώ στο τέταρτο κεφάλαιο περιγράφονται τα λογισμικά και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο πέμπτο κεφάλαιο περιγράφεται το σχέδιο δράσης για την εκπόνηση της εργασίας και συγκρίνονται οι διαθέσιμες τεχνολογίες μέχρι να καταλήξουμε σε αυτές που θα χρησιμοποιήσουμε στην εφαρμογή.

Στο έκτο κεφάλαιο αναπτύσσεται η διαδικασία της δημιουργίας της εφαρμογής χωρισμένη σε αναλυτικά βήματα και στο επόμενο κεφάλαιο αναλύονται τα συμπεράσματα που βγήκαν από την ολοκλήρωση της πτυχιακής εργασίας.

## 2. Μεθοδολογία ανάπτυξης ιστοσελίδων

Είναι πολύ βασικό να κατανοήσουμε την διαφορά των δυναμικών ιστοσελίδων με τις στατικές πριν ξεκινήσουμε ανάπτυξη της δικής μας ιστοσελίδας, αφού θα πρέπει να καθορίσουμε σε ποια από τις δύο κατηγορίες αυτή ανήκει και να προχωρήσουμε αναλόγως στην υλοποίηση της. Στην συνέχεια θα πρέπει να αναλύσουμε διεξοδικά τον τρόπο ανάπτυξης της.

### 2.1 Στατικές ιστοσελίδες

Στατικές ιστοσελίδες ονομάζονται αυτές που αποτελούνται κυρίως από αρχεία Html και Css που μεταδίδονται αυτούσια από τον διακομιστή που τις φιλοξενεί, χωρίς να υπάρχει κάποιου είδους επεξεργασία πριν. Μια στατική ιστοσελίδα είναι πολύ εύκολο να αναγνωριστεί αφού κάθε φορά που την επισκεπτόμαστε η εμφάνιση της παραμένει ίδια και αυτή δεν αλληλεπιδρά με τον χρήστη. Πλέον δεν συναντάμε συχνά τέτοιες ιστοσελίδες σε σχέση με παλαιότερα, αφού έχουν πολύ περιορισμένες δυνατότητες και δεν ανταποκρίνονται στις ανάγκες του σύγχρονου χρήστη. Πλέον χρησιμοποιούνται μόνο σε σπάνιες περιπτώσεις που αφορούν την προβολή πολύ απλών πραγμάτων, όπως πχ για να δηλωθεί ότι μια ιστοσελίδα είναι υπό κατασκευή κλπ.

### 2.2 Δυναμικές ιστοσελίδες

Δυναμική είναι μια ιστοσελίδα που χρησιμοποιεί μία ή περισσότερες γλώσσες προγραμματισμού για να επεξεργαστεί το περιεχόμενο των αρχείων που θα μεταφέρει στον χρήστη για προβολή. Πλέον δυναμικές είναι οι περισσότερες ιστοσελίδες που υπάρχουν αυτή τη στιγμή στο διαδίκτυο, αφού μπορούν να καλύψουν ένα πολύ ευρύ φάσμα αναγκών. Σε αυτή την κατηγορία ανήκουν οι ιστοσελίδες που αλληλεπιδρούν με τον χρήστη, αυτές που έχουν την δυνατότητα να διαχειρίζονται δεδομένα και γενικότερα να τροποποιούν το αποτέλεσμα που βλέπει ο τελικός χρήστης ανάλογα με την εφαρμογή. Κάποια χαρακτηριστικά παραδείγματα τέτοιων ιστοσελίδων μπορεί να είναι ένα ενημερωτικό site, ένα ηλεκτρονικό κατάστημα μέχρι και ένα κοινωνικό δίκτυο. Οι δυνατότητες που έχουν αυτού του είδους οι ιστοσελίδες είναι πραγματικά πάρα πολλές και ποικίλουν ανάλογα με την τεχνολογία που θα επιλέξουμε να αναλάβει το δυναμικό κομμάτι του προγραμματισμού.

Με βάση τις παραπάνω κατηγοριοποιήσεις συμπεράνουμε εύκολα ότι εφαρμογή που πραγματεύεται αυτή η πτυχιακή εργασία είναι μια ξεκάθαρα δυναμική ιστοσελίδα και γι αυτό σε αυτό το κεφάλαιο θα αναλύσουμε την κατασκευή μιας τέτοιας ιστοσελίδας.

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να τη χωρίσουμε σε μικρότερα μέρη και να τα αναλύσουμε επιμέρους, εξετάζοντας τις διαθέσιμες τεχνολογίες. Όλες οι δυναμικές ιστοσελίδες χωρίζονται σε 2 βασικά μέρη, την εμφάνιση (front end ) και το δυναμικό κομμάτι (back end).

#### 2.2.1 Εμφάνιση της ιστοσελίδας

Η εμφάνιση μια ιστοσελίδας είναι η πρώτη εικόνα που αντικρίζουμε μόλις την ανοίξουμε αλλά και το πράγμα που συνήθως ξεκινάμε στην διαδικασία κατασκευής. Ουσιαστικά αποτελείται

από αρχεία δομημένα κατά τέτοιο τρόπο ώστε να μπορούν να διαβαστούν από τους browsers. Οι τεχνολογίες που χρησιμοποιούνται για αυτή τη δουλειά είναι η HTML και το CSS

### Html (Hyper Text Markup Language)



Εικόνα 1 html

Η Html είναι η γλώσσα με την οποία κατασκευάζεται το οπτικό μέρος μιας ιστοσελίδας. Οι ιστοσελίδες που επισκεπτόμαστε στο Internet δεν είναι τίποτε άλλο παρά αρχεία τα οποία περιέχουν κώδικα γραμμένο στην γλώσσα Html. Τα αρχεία αυτά έχουν επέκταση .htm ή .html. Οι browsers στην συνέχεια διαβάζουν αυτά τα αρχεία και τα εμφανίζουν το οπτικό αποτέλεσμα στην οθόνη μας.

Ο κώδικα html μιας σελίδας μπορεί να βρεθεί πολύ εύκολα κάνοντας δεξί κλικ επάνω σε οποιοδήποτε σημείο μιας σελίδας (όχι επάνω σε εικόνα ή σύνδεσμο) και στην συνέχεια στην επιλογή View page source ή View source.

Για την συγγραφή του κώδικα html χρειάζεται η χρήση ενός WYSIWYG (What You See Is What You Get) προγράμματος επεξεργασίας ιστοσελίδων. Μερικά τέτοια προγράμματα είναι το Dreamweaver της Macromedia, το CoffeeCup HTML Editor της CoffeeCup Softwares, το Effective Site Studio της Effective Studio και πολλά άλλα. Υπάρχει πληθώρα τέτοιων προγραμμάτων στο Internet είτε για δωρεάν κατέβασμα, είτε επί πληρωμή. Αρκετοί σχεδιαστές Ιστοσελίδων που χρησιμοποιούν περισσότερα από ένα πρόγραμμα αξιοποιώντας έτσι τα πλεονεκτήματα του καθενός από αυτά, αφού είναι φυσικό ορισμένα προγράμματα να υπερτερούν σε κάποιες λειτουργίες εκεί που άλλα υστερούν.

Αυτός είναι ο κώδικας για μια πολύ απλή σελίδα html στην έκδοση 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title> this is the title </title>
</head>
<body>
    <p>Page content.</p>
</body>
</html>
```

## CSS (Cascading Style Sheets)



Εικόνα 2 CSS

Το Css χρησιμοποιείται για να απλοποιήσει τον κώδικα html και να του προσθέσει παραπάνω δυνατότητες, αφού είναι αυτό που τροποποιεί την εμφάνιση των διαφόρων στοιχείων σε μια σελίδα html.

Γράφοντας τις σελίδες μας μόνο με Html κώδικα, μπορούμε να ορίσουμε το χρώμα και το μέγεθος του κειμένου αλλά και άλλων στοιχείων της σελίδας (όπως πίνακες, links, εικόνες κτλ). Για να αλλάξουμε το χρώμα κάποιου κειμένου ή το χρώμα ενός πίνακα, θα πρέπει να βρούμε το χρώμα αυτό μέσα στον κώδικα και να το αλλάξουμε. Η διαδικασία αυτή μπορεί να φαντάζει εύκολη όταν έχουμε να διαχειριστούμε μια μόνο σελίδα, αλλά ένα site αποτελείται από δεκάδες σελίδες τις οποίες χρειάζεται να διαχειριζόμαστε εύκολα και γρήγορα.

Αυτό το πρόβλημα λύνεται με τη χρήση Css σε εξωτερικά αρχεία όπου μπορούμε να ορίζουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόσουμε τα στυλ αυτά στα στοιχεία των σελίδων της ιστοσελίδας μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό. Έτσι αν έχουμε ορίσει ένα στυλ για το κύριο μενού του site, τότε θα χρειάζεται να αλλάξουμε το χρώμα του στυλ αυτού και αυτόματα θα εφαρμοστεί σε όλες τις σελίδες.

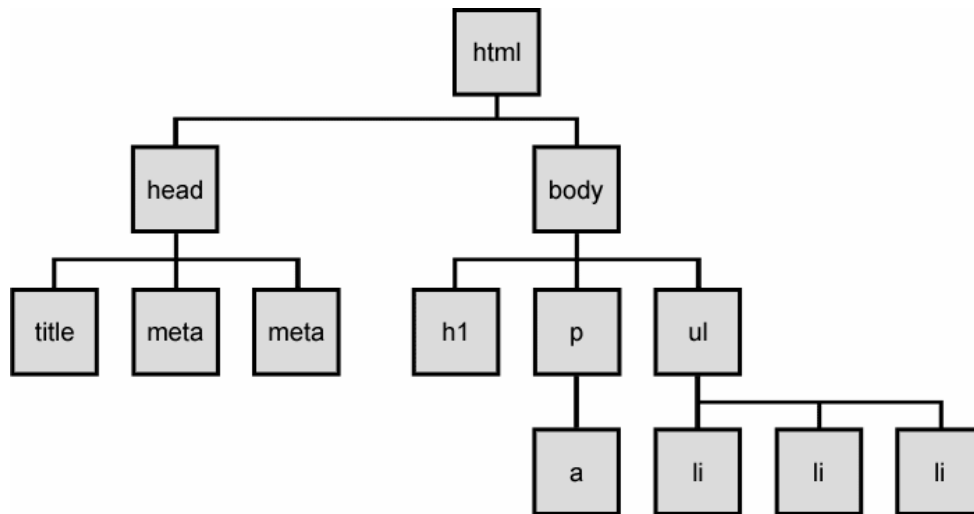
Επίσης ένα σημαντικό πλεονέκτημα της χρήσης του Css είναι ότι απλοποιείται ο κώδικας και δεν χρειάζονται πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον browser μόνο μια φορά και έπειτα αποθηκεύεται στην cache memory, μειώνοντας έτσι το μέγεθος της πληροφορίας που γίνεται download από τους browsers.

Η βασική δομή ενός αρχείου Css αποτελείται από κανόνες και δηλώσεις στυλ για κάθε στοιχείο.

Για παράδειγμα ένας απλός κανόνας που κάνει κόκκινο το φόντο μιας σελίδας είναι αυτός.

```
body {  
    background: red;  
}
```

## DOM (Document Object Model)



Εικόνα 3 Ιεραρχία στοιχείων στο DOM

Το DOM είναι ένα μοντέλο παρουσίασης της W3C (World Wide Web Consortium), ανεξάρτητο από γλώσσες και πλατφόρμες που χρησιμοποιείται για την αναπαράσταση html και xml αρχείων. Στην ουσία είναι οι κανόνες που ορίζουν την δομή ενός αρχείου και δημιουργούν την ιεραρχία των στοιχείων που το συνθέτουν, ώστε να επιτρέψουν την πρόσβαση σε εξωτερικά scripts. Αυτή η δομή χρησιμοποιείται κυρίως από την JavaScript για να αναγνωρίσει τα στοιχεία τα οποία θα επηρεάσει δυναμικά και όχι στην εμφάνιση του html αρχείου από τον browser.

### Αναλυτικά η JavaScript μπορεί μέσω του DOM να:

- Αλλάξει όλα τα Html στοιχεία μιας σελίδας
- Αλλάξει όλες τις ιδιότητες των Html στοιχείων μιας σελίδας
- Αλλάξει όλους τους κανόνες των Html στοιχείων μιας σελίδας
- Αφαιρέσει υπάρχοντα Html στοιχεία μιας σελίδας
- Προσθέσει νέα Html στοιχεία σε μια σελίδα
- Έχει πρόσβαση σε όλα τα γεγονότα (events) που συμβαίνουν στα Html στοιχεία
- Να δημιουργήσει καινούρια γεγονότα που συμβαίνουν στα Html στοιχεία

### Υπάρχουν 3 είδη DOM

- 1) Core DOM – για όλα τα έγγραφα
- 2) XML DOM – για XML έγγραφα
- 3) HTML DOM – για Html έγγραφα

Αυτό που μας ενδιαφέρει εδώ είναι τα έγγραφα Html που περιλαμβάνει αναλυτικά:

- 1) Τα Html στοιχεία ως αντικείμενα
- 2) Τις ιδιότητες όλων των Html στοιχείων
- 3) Τις μεθόδους πρόσβασης σε όλα τα Html στοιχεία
- 4) Τα γεγονότα που συμβαίνουν στα Html στοιχεία

## 2.2.2 Front End Framework

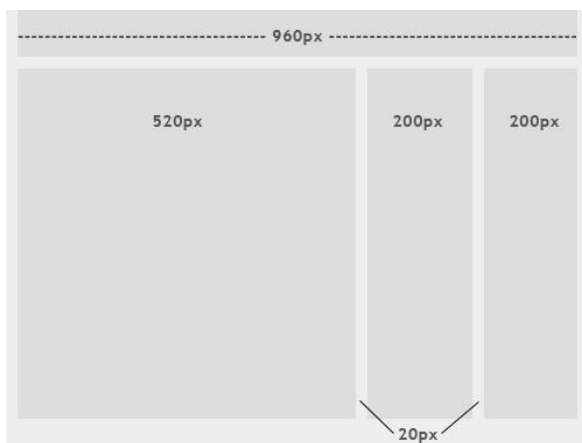
Στην καθημερινή μας ζωή πολλές φορές πρέπει να πάρουμε κάποιες αποφάσεις που αφορούν τα εργαλεία που θα χρησιμοποιήσουμε για να κάνουμε μια συγκεκριμένη δουλειά. Συνήθως επιλέγουμε αυτά που ταιριάζουν περισσότερο στις ανάγκες μας και μας γλιτώνουν πολύτιμο χρόνο. Αυτό μπορεί να γίνει με την επιλογή ενός εργαλείου που έχει φτιάξει κάποιος άλλος, καθώς δεν χρειάζεται κάθε φορά να «ανακαλύπτουμε τον τροχό».

Αν λοιπόν το θέμα μας είναι η σχεδίαση μιας ιστοσελίδας, δεν χρειάζεται να το κάνουμε από την αρχή για κάθε στοιχείο της, αλλά μπορούμε να χρησιμοποιήσουμε ένα front end framework. Δηλαδή στην ουσία, κάποια έτοιμα κομμάτια κώδικα Html, Css και JavaScript που μπορούν να χρησιμοποιηθούν από τους σχεδιαστές του front end, δηλαδή του οπτικού κομματιού μιας ιστοσελίδας. Με την επιλογή αυτή εξοικονομούμε πολύτιμο χρόνο καθώς επίσης με τη χρήση ενός framework είμαστε σίγουροι ότι η εκάστοτε ιστοσελίδα που κατασκευάζουμε θα εμφανίζεται σωστά σε όλους τους browsers όλων των συσκευών που κυκλοφορούν στην αγορά, αν φυσικά αυτές υποστηρίζονται στις προδιαγραφές του framework που επιλέξαμε.

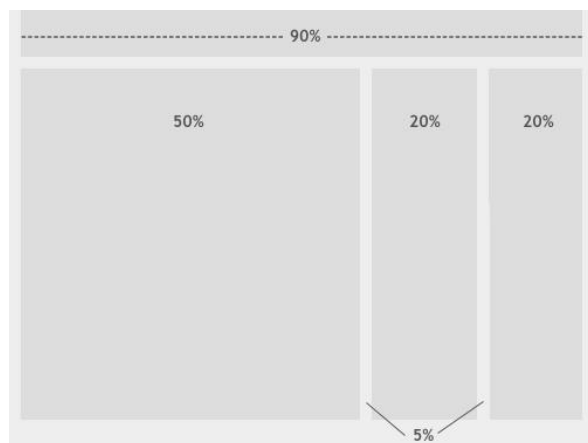
Αυτή τη στιγμή υπάρχουν πάνω από 40 διαφορετικά front end frameworks διαθέσιμα για download δωρεάν και έτοιμα για χρήση, πράγμα που καθιστά πιο δύσκολη την απόφαση ποιο από όλα πρέπει να χρησιμοποιήσουμε. Για να πάρουμε την σωστή απόφαση θα πρέπει να γνωρίζουμε τις ανάγκες που πρέπει αυτό να καλύπτει και να συγκρίνουμε τις διαφορές δυνατότητες που αυτά έχουν.

Για το λόγο αυτό επιλέξαμε τα 5 πιο δημοφιλή που κυκλοφορούν και συγκρίναμε τις δυνατότητές τους. Στον παρακάτω πίνακα παρατίθενται:

- 1) Οι δυνατότητες εμφάνισης, δηλαδή αν είναι οι στήλες που φιλοξενούν το περιεχόμενο, προσδιορίζονται σε αριθμό pixels (fixed) ή ως ένας αριθμός % του πλάτους της οθόνης προβολής.

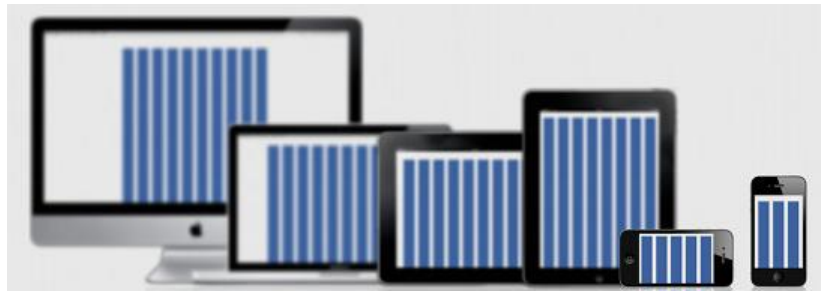


Εικόνα 4 Δείγμα fixed layout



Εικόνα 5 Δείγμα fluid layout

- 2) Μέγιστος αριθμός στηλών που μπορεί να δημιουργήσει.



Εικόνα 6 Μέγιστος αριθμός στηλών ανά συσκευή

- 3) Πρόσθετες δυνατότητες, προσθετοί κανόνες που μπορεί να φιλοξενεί, εικονίδια που περιέχονται ή γραμμές, κώδικας JavaScript κλπ.
- 4) Άδεια που μας δίνεται από τον δημιουργό για τις διάφορες χρήσεις που μπορούμε να κάνουμε.

Πίνακας 1 Σύγκριση δημοφιλών front end frameworks

Όνομα	Δυνατότητες	Στήλες	Πρόσθετα	Άδεια
Bootstrap	Fixed, fluid, Responsive	12	Icons, typography,++	Apache v2.0
Foundation	fluid	12	Navigation, Js	MIT
Skeleton	Fixed, Responsive	16	Forms, typography	MIT
Html5 Boilerplate	Fixed, Responsive	12	Icons, Js	Open Source
Less	Fixed, Responsive	10	-	MIT

## 2.2.3 Responsive Web Design



Εικόνα 7 Εμφάνιση της ίδιας ιστοσελίδας σε διαφορετικές συσκευές

Παρακολουθώντας τις εξελίξεις της τεχνολογίας μπορούμε να παρατηρήσουμε ότι παράγονται όλο και μικρότερες σε μέγεθος συσκευές με δυνατότητα σύνδεσης στο διαδίκτυο και



μάλιστα αυτές γίνονται όλο και πιο δημοφιλείς. Πλέον οι κινητές συσκευές όπως κινητά και tablets καταλαμβάνουν ένα μεγάλο μερίδιο στην αγορά των ηλεκτρονικών συσκευών πράγμα που σημαίνει ότι όλο και περισσότεροι χρήστες επιλέγουν να πλοηγηθούν στο διαδίκτυο με αυτές.

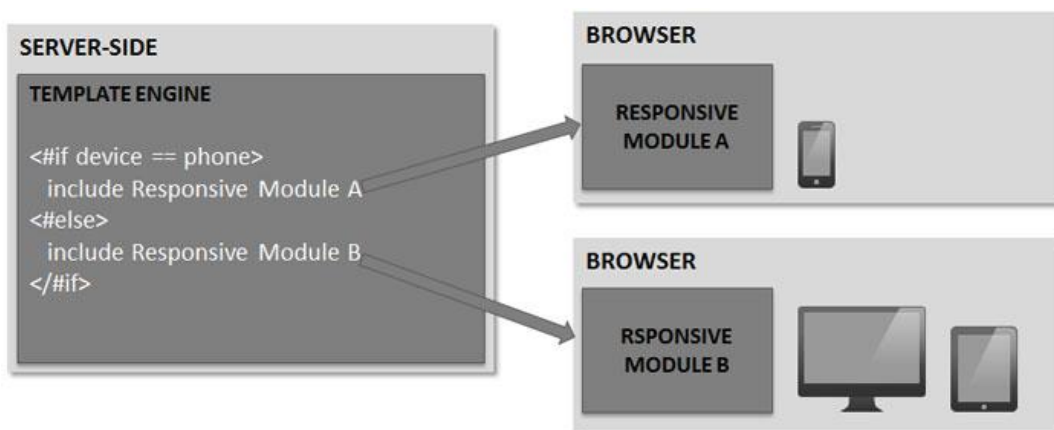
Αυτό σημαίνει για ένα σχεδιαστή ιστοσελίδων ότι πρέπει να προβλέψει ότι η δημιουργία του θα πρέπει να εμφανίζεται σωστά σε όλους τους τύπους συσκευών, ανεξάρτητα με το μέγεθος της οθόνης που αυτές διαθέτουν. Από αυτή την ανάγκη γεννήθηκε ο όρος responsive web design και περιγράφει την προσαρμογή που πρέπει να έχει μια ιστοσελίδα στην εκάστοτε οθόνη του χρήστη που την βλέπει. Αυτό πρακτικά σημαίνει ότι μια ιστοσελίδα θα έπρεπε να εμφανίζεται σωστά από τη οθόνη της μικρότερης κινητής συσκευής μέχρι μια οθόνη υψηλής ευκρίνειας (HD).

**Για να επετεύχθη αυτός ο σκοπός χρησιμοποιούνται 2 τρόποι σε συνδυασμό μεταξύ τους.**

```
/* for 480px or less */  
@media screen and (max-width: 480px) {  
  
    #header {  
        height: auto;  
    }  
    h1 {  
        font-size: 24px;  
    }  
    #sidebar {  
        display: none;  
    }  
}
```

Εικόνα 8 Δείγμα Media Query για οθόνες μέχρι 480pixels

- 1) Χρήση CSS Media Queries. Δηλαδή κώδικας CSS που εφαρμόζει ξεχωριστούς κανόνες για κάθε περίπτωση, ανάλογα με την ανάλυση ή τον τύπο της συσκευής που αφορά.
- 2) Χρήση fluid στοιχείων. Δηλαδή στοιχεία που το μέγεθός τους ορίζεται με βάση το ποσοστό % της οθόνης προβολής και όχι με συγκεκριμένο αριθμό pixels, έτσι ώστε να προσαρμόζονται στην οθόνη.



Εικόνα 9 Server side responsive design

Επίσης μια εναλλακτική προσέγγιση που υπάρχει για να λυθεί το πρόβλημα της εμφάνισης μιας ιστοσελίδας σε οποιαδήποτε οθόνη, είναι ο έλεγχος από την πλευρά του διακομιστή (server) για το τι είδους συσκευή ζητάει την εμφάνιση της ιστοσελίδας. Έτσι υπάρχει η δυνατότητα να μεταδίδει μια διαφορετική έκδοση της ίδιας ιστοσελίδας που θα ήταν πχ πιο φιλική σε μικρές οθόνες αν αυτή ζητείται από μια κινητή συσκευή.

Τα **πλεονεκτήματα** αυτής της λύσης είναι αρκετά καθώς με το να υπάρχει ειδική έκδοση ανάλογα με την συσκευή μπορεί να εκμεταλλεύεται πιο πολλές δυνατότητες που αυτή έχει, πχ το gps ή την κάμερα του κινητού. Επίσης με αυτό τον τρόπο μπορούν να φτιαχτούν πιο σύνθετες εφαρμογές ή πιο ελαφριές, αφού θα μπορούν να αφαιρεθούν οι περιττές δυνατότητες ανάλογα με την συσκευή.

Τα **μειονεκτήματα** που προκύπτουν βέβαια είναι ότι η εφαρμογή γίνεται πιο σύνθετη και πλέον θα χρειάζεται να συντηρούνται και να ανανεώνονται περισσότερες εκδόσεις από εκεί που χρειαζόταν μόνο μία.

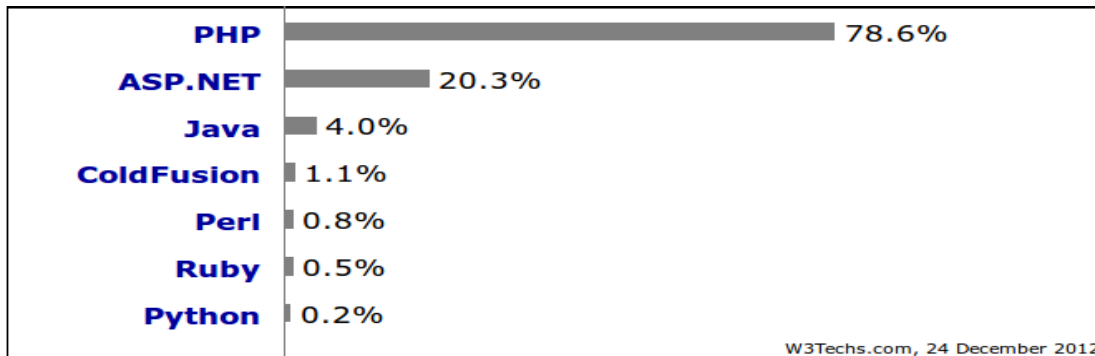
## 2.2.4 Δυναμικό κομμάτι της ιστοσελίδας

Το δυναμικό κομμάτι (back end) είναι αυτό που την κάνει να ξεχωρίζει από μια στατική ιστοσελίδα και έχει να κάνει με την γλώσσα προγραμματισμού που τρέχει στην πλευρά του server και αναλαμβάνει να διαμορφώσει τα αρχεία πριν αυτά σταλούν στον τελικό χρήστη.

Οι διαθέσιμες τεχνολογίες που υπάρχουν και είναι κατασκευασμένες γι αυτή τη δουλειά είναι πολλές και η κάθε μια προσφέρει τα δικά της χαρακτηριστικά αλλά και καταναλώνει και τους ανάλογους πόρους.

Οι πιο δημοφιλείς server-side γλώσσες προγραμματισμού:

- 1) Php
- 2) Java
- 3) Python
- 4) Ruby
- 5) Perl
- 6) Asp.net
- 7) Coldfusion



Εικόνα 10 Στατιστικά χρήσης γλωσσών για το έτος 2012

## 2.2.5 Php Frameworks

Όπως και στο front end κομμάτι μιας ιστοσελίδας, έτσι και στο back end, προσπαθούμε να εξοικονομήσουμε πολύτιμο χρόνο και να μειώσουμε την περιττή δουλειά παίρνοντας κάποια κομμάτια κώδικα έτοιμα, ενώ ταυτόχρονα προσπαθούμε να κρατήσουμε την ευελιξία της εφαρμογής μας.

**Σε αυτή την περίπτωση έχουμε δυο επιλογές:**

- 1) Τη χρήση κάποιου CMS (Content Management System) που είναι μια ολοκληρωμένη λύση που παρέχει την ευκολία της δημιουργίας μιας εφαρμογής εξολοκλήρου χωρίς να χρειαστεί να γράψουμε ούτε μια γραμμή κώδικα. Αυτή η λύση έχει το πλεονέκτημα της γρήγορης κατασκευής αφού υπάρχουν έτοιμα κομμάτια κώδικα σχεδόν για κάθε δουλειά και ενδείκνυται για αρχάριους, περιορίζει όμως την ευελιξία και τις δυνατότητες της εφαρμογής μας.
- 2) Τη χρήση κάποιου framework που αναλαμβάνει ένα μεγάλο κομμάτι διαχείρισης της δουλειάς που πρέπει να γίνει. Αυτή η λύση χρειάζεται εξοικείωση με τον προγραμματισμό καθώς αναλόγως ποιο framework θα επιλέξουμε θα χρειαστεί να γράψουμε και ανάλογο μέρος του κώδικα. Αυτή η επιλογή όμως μας δίνει πολύ περισσότερες δυνατότητες στην ευελιξία της εφαρμογής μας, αφού την κατασκευάζουμε όπως ακριβώς τη χρειαζόμαστε.

**Πλεονεκτήματα στη χρήση ενός framework:**

- 1) Πολύ γρήγορη ανάπτυξη της εφαρμογής λόγω των πολλών δυνατοτήτων που δίνονται στον developer.
- 2) Ευκολία στην συντήρηση και την επέκταση της εφαρμογής στο μέλλον και της προσθήκης καινούριων δυνατοτήτων, ή της αφαίρεσης κάποιων άλλων πιθανών παρωχημένων.
- 3) Μεγάλη ασφάλεια που συμβαδίζει με τις εξελίξεις της τεχνολογίας καθώς τα frameworks ανανεώνονται συνεχώς για να συμβαδίζουν με αυτές.
- 4) Πολλές πρόσθετες δυνατότητες και βιβλιοθήκες.
- 5) Μεγάλη κοινότητα χρηστών που μπορεί να υποστηρίξει τον developer σε οποιοδήποτε πρόβλημα προκύψει στην κατασκευή.

Στην περίπτωση που χρησιμοποιούμε την γλώσσα προγραμματισμού PHP, θα πρέπει να επιλέξουμε ένα αξιόπιστο framework, από τα πολλά που υπάρχουν και είναι φτιαγμένα για αυτή τη γλώσσα.

Για να επιλέξουμε το πιο σωστό framework που να ταιριάζει στις ανάγκες της εφαρμογής που θέλουμε να φτιάξουμε, πρέπει να δούμε τα χαρακτηριστικά του καθενός και να τα συγκρίνουμε μεταξύ τους.

**Βασικά κριτήρια για να επιλέξουμε το σωστό framework είναι:**

- 1) Το μέγεθος της εφαρμογής που θέλουμε να φτιάξουμε. Άλλα framework ενδείκνυται για μια μεγάλη εφαρμογή με σύνθετα χαρακτηριστικά και άλλα για μια πιο απλή.
- 2) Η έκδοση της PHP που υποστηρίζει, καθώς η γλώσσα PHP χρησιμοποιείται σε δυο εκδόσεις, την 4 και την 5 ενώ δεν υποστηρίζουν όλα τα frameworks και τις δύο εκδόσεις.
- 3) Η εξοικείωση μας με την γλώσσα PHP. Ανάλογα με το framework θα κληθούμε να γράψουμε μεγαλύτερο ή μικρότερο μέρος του κώδικα, αλλά και να μάθουμε τον τρόπο λειτουργίας του, που ποικίλει σε πολυπλοκότητα.
- 4) Το ποσό δημοφιλές είναι. Όσο μεγαλύτερη κοινότητα υποστήριξης έχει, τόσο γρηγορότερα λύνονται τα προβλήματα που μπορεί να προκύψουν.
- 5) Οι πρόσθετες δυνατότητες που αυτό δίνει.

**Κάποιες σημαντικές δυνατότητες είναι:**

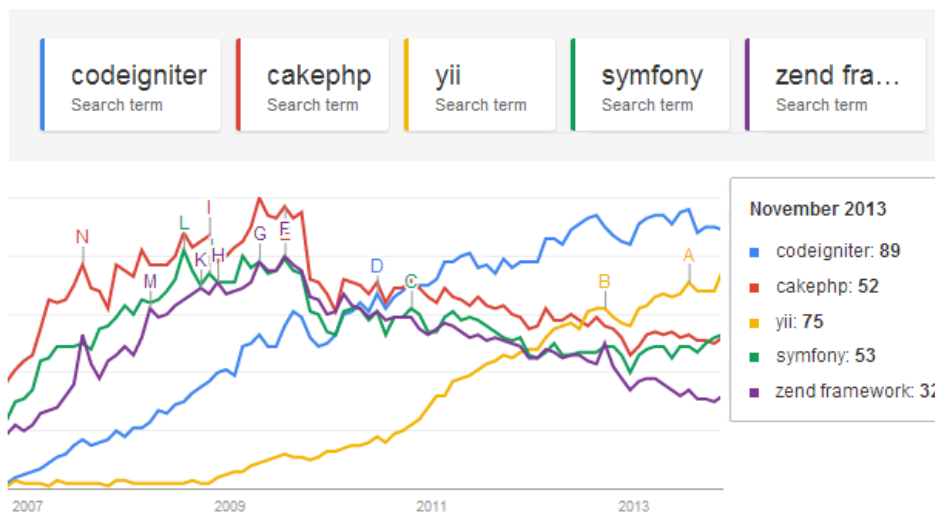
- A) Ενσωματωμένο σύστημα διαχείρισης αιτημάτων AJAX
- B) Έτοιμα templates, δηλαδή ξεχωριστές εμφανίσεις της ιστοσελίδας.
- Γ) Υποστήριξη πολλαπλών βάσεων δεδομένων.
- Δ) Validation, δηλαδή έλεγχος των εισερχομένων πληροφοριών.
- E) Authentication, δηλαδή διαδικασία εισόδου πιστοποιημένων χρηστών.



Εικόνα 11 Λογότυπα των πιο δημοφιλών php frameworks

Πίνακας 2 Σύγκριση δημοφιλών PHP frameworks

Όνομα	Έκδοση PHP	Πρόσθετα χαρακτηριστικά
CakePHP	4, 5	Ajax, Authentication, Multiple DB's
CodeIgniter	4,5	Templates , Multiple DB's,
Symfony	5	Ajax, Authentication, Multiple DB's
Yii	5	Ajax, Templates, Authentication, Multiple DB's
Send	5	Ajax, Templates, Authentication, Multiple DB's



Εικόνα 12 Δείκτες σύγκρισης δημοφιλίας με βάση τις δουλειές

## 3. Τεχνολογίες & Γλώσσες Προγραμματισμού

### 3.1 Html5 & Css3

#### 3.1.1 Html5

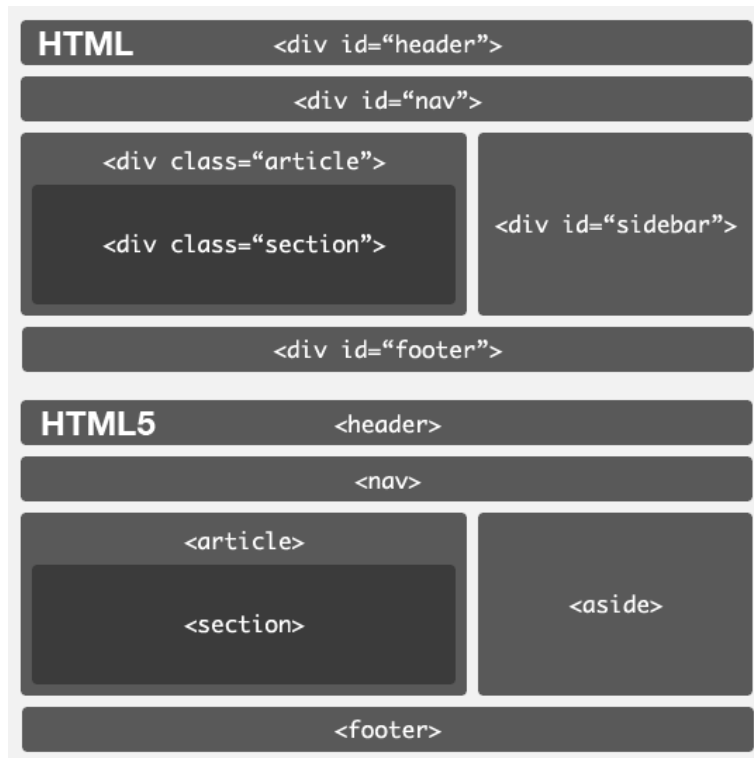


Εικόνα 13 Λογότυπο της Html5

Η Html5 είναι η τελευταία έκδοση της markup γλώσσας Html (Hyper Text Markup Language), που χρησιμοποιείται ως είσοδος για τους διάφορους browser ώστε αυτοί να συνθέσουν την εμφάνιση μιας ιστοσελίδας. Κάθε ιστοσελίδα που υπάρχει στο διαδίκτυο περιέχει σίγουρα έστω και ένα αρχείο html που ακολουθεί κάποιες βασικές αρχές που ορίζονται από τη δομή τη γλώσσας ανάλογα με την έκδοση της. Η βασική δομή της γλώσσας οργανώνεται με τα λεγόμενα tags που είναι αυτά που στη συνέχεια ο browser τα μεταφράζει στα διάφορα στοιχεία της ιστοσελίδας, όπως πχ ένα κείμενο, ένα μενού ή μια εικόνα.

Αν θέλαμε να γράψουμε ένα αρχείο Html5 που να τυπώνει απλά μια φράση αυτός είναι ο πιο σύντομος και επικυρωμένος κώδικας.

```
<!DOCTYPE html>  
<html>  
  <body>  
    Hello World  
  </body>  
</html>
```



Εικόνα 14 Διαφορές Html5 με τις προηγούμενες εκδόσεις

Η Html5 φέρνει μια σειρά από καινοτομίες και απλοποιεί πολλές διεργασίες που πριν ήταν αρκετά πολύπλοκες να γίνουν. Ενσωματώνει αρκετά στοιχεία JavaScript μέσα σε ένα μόλις tag και απαλλάσσει τους developers από τη χρήση παλαιότερων τεχνολογιών που αποσύρονται. Η πιο χαρακτηριστική περίπτωση είναι η σταδιακή αντικατάσταση του Adobe flash που ήδη δεν υποστηρίζεται σε πολλές κινητές συσκευές.

Μερικές από τις πιο σημαντικές καινοτομίες που φέρνει είναι οι εξής:

**A) Νέα tags που απλοποιούν τη δομή όπως φαίνεται στην εικόνα 11.**

<header>	Εισάγει την κεφαλίδα στο έγγραφο
<hgroup>	Κατηγοριοποιεί τις κεφαλίδες
<nav>	Εισάγει το μενού περιήγησης
<section>	Εισάγει μια περιοχή
<main>	Καθορίζει την κύρια περιοχή ενός έγγραφου
<article>	Εισάγει ένα άρθρο
<aside>	Εισάγει δευτερεύον περιεχόμενο
<footer>	Εισάγει το τελείωμα ενός έγγραφου
<details>	Εισάγει λεπτομερείς που μπορούν και να κρυφτούν
<summary>	Εισάγει μια σύνοψη για τις λεπτομέρειες
<figure>	Εισάγει μια εικόνα ή ένα γράφημα
<figcaption>	Εισάγει τις πληροφορίες του παραπάνω tag
<mark>	Εισάγει επιλεγμένο κείμενο
<time>	Εισάγει ημερομηνία/ώρα
<wbr>	Εισάγει πιθανή αλλαγή γραμμής
<dialog>	Εισάγει πεδίο διαλόγου
<command>	Εισάγει μια εντολή που μπορεί να δοθεί

<meter>	Εισάγει το μέτρο σύγκρισης για ένα αριθμό
<progress>	Προβάλλει το ποσοστό ολοκλήρωσης μιας διαδικασίας

#### **B) Validation στις φόρμες ανάλογα με τον τύπο τους.**

Δυνατότητα validation για email, url, αριθμούς, τηλέφωνα, ημερομηνίες αλλά και υποχρεωτικά πεδία.

#### **Γ) Ενσωμάτωση βίντεο, ήχου και JavaScript μέσω canvas.**

Μέχρι σήμερα, για να παίξουμε ένα video σε έναν browser χρειαζόμαστε κάποιο plug-in (πρόσθετο) όπως το Flash ή το QuickTime. Η HTML5 εισάγει έναν νέο τρόπο, και αυτός είναι με την χρήση του στοιχείου video. Το μόνο που χρειάζεται να γράψουμε για να εμφανίσουμε το video μας είναι:

```
<video src="movie.ogg" controls="controls">
</video>
```

#### **Δ) Αποθήκευση στοιχείων τοπικά.**

Μέχρι τώρα η αποθήκευση δεδομένων στον υπολογιστή του επισκέπτη μιας ιστοσελίδας γινόταν με τα cookies. Τα αρχεία cookies όμως δεν είναι κατάλληλα να αποθηκεύουν μεγάλο όγκο δεδομένα, καθώς σαρώνονται από κάθε αίτηση του server κάνοντας την διαδικασία αργή και μη αποτελεσματική. Στην HTML5 τα δεδομένα δεν σαρώνονται μετά από κάθε αίτηση του server, αλλά μόνο όταν απαιτείται. Έτσι είναι εφικτό να αποθηκεύονται μεγάλοι όγκοι δεδομένων χωρίς να επηρεάζεται η απόδοση μιας ιστοσελίδας. Επίσης τα δεδομένα αποθηκεύονται σε διαφορετικές περιοχές του δίσκου για κάθε ιστοσελίδα, έτσι ώστε μια ιστοσελίδα να έχει πρόσβαση μόνο στα δικά της δεδομένα. Όπως με τα cookies, η HTML5 αποθηκεύει δεδομένα στον client χρησιμοποιώντας JavaScript

#### **E) Geolocation**

Πρόκειται για υπηρεσία εντοπισμού του γεωγραφικού στίγματος του χρήστη, με τη χρήση της JavaScript. Αυτή η λειτουργία ενεργοποιείται με τη χρήση του παρακάτω κώδικα:

```
<script
src="http://maps.google.com/maps/api/js?sensor=true"></script>
```

#### **ΣΤ) Drag & Drop σχεδόν για κάθε στοιχείο.**

Το Drag and drop είναι ένα πολύ κοινό χαρακτηριστικό και προσθέτει ευκολία στη χρήση. Στην HTML5, το drag and drop είναι μέρος του προτύπου, καθώς κάθε στοιχείο μπορεί να είναι draggable.

## **3.1.2 CSS3**

Όπως και η Html5, έτσι και το CSS3 είναι η νεότερη έκδοση της γλώσσας CSS (Cascading Style Sheets) που χρησιμοποιείται για την μορφοποίηση του κώδικα Html που δημιουργεί τα διάφορα στοιχεία.



Εικόνα 15 Λογότυπο CSS3

Το CSS3 προσφέρει κάποια καινούρια χαρακτηριστικά, τα οποία δίνουν μια σειρά από πλεονεκτήματα στον σχεδιασμό μιας ιστοσελίδας αφού έρχεται για να απλοποιήσει κάποιες διαδικασίες που παλιότερα απαιτούσαν περισσότερους πόρους και χρόνο για να φτιαχτούν. Τα κύρια οφέλη του CSS3 είναι ότι επιτρέπει την δημιουργία χαρακτηριστικών που αλλιώς θα χρειαζόντουσαν την JavaScript για να εφαρμοστούν, έτσι απλοποιεί πολύ τον κώδικα και κάνει τις σελίδες πιο ελαφριές. Ένα άλλο πλεονέκτημα στο CSS3 είναι η δυνατότητα δημιουργίας γραφικών, που μέχρι τώρα χρειάζονταν το Photoshop για να δημιουργηθούν και να εμφανιστούν μέσω εικόνων στην ιστοσελίδα.

Μερικές καινοτομίες που φέρνει το CSS3 είναι οι εξής:

### 1) Media queries

Παράδειγμα:

```
@media screen {  
  * { font-family: sans-serif }  
}
```

Δείτε περισσότερα στο κεφάλαιο 2.2.2 Responsive Web Design

### 2) Gradients

Μια καινοτομία του CSS3 καθώς στις προηγούμενες εκδόσεις του CSS δεν υπήρχε η δυνατότητα μίξης των χρωμάτων γραμμικά.

```
body{  
  background: linear-gradient(top, #000 0%, #111 50%,  
  #222 70%, #333 100%);  
}
```

### 3) Animation και Transitions

Τα Transitions είναι ένα set κανόνων CSS που μας επιτρέπουν να αλλάζουμε τιμές στις ιδιότητες ενός στοιχείου, ομαλά και σταδιακά σε συγκεκριμένο χρόνο. Με άλλα λόγια μας επιτρέπουν να αλλάζουμε τιμές στα στοιχεία με τέτοιο τρόπο ώστε να πετυχαίνουμε ένα εφέ σαν το animation.

```
body{  
  transition: width 2s;  
}
```



#### 4) Άπειρες γραμματοσειρές

Απλώς παίρνουμε το αρχείο της γραμματοσειράς που θέλουμε, το φορτώνουμε στη σελίδα και καλούμε τη γραμματοσειρά με το @font-face. Αυτό μας δίνει την δυνατότητα να δώσουμε ξεχωριστή εμφάνιση στα διάφορα κείμενα χωρίς την χρήση εικόνων.

```
@font-face{
    font-family: myfamily;
    src: url(font.woff);
}

body{
    font-family:myfamily;
}
```

#### 5) Φόντο με πολλαπλές εικόνες

Αυτό μας επιτρέπει να ορίζουμε πολλές εικόνες ως φόντο για ένα στοιχείο, τη θέση της κάθε εικόνας και την επανάληψή της.

```
body{
    background:url(img.jpg), url(img2.jpg);
    background-size:100% 100%;
    background-repeat:no-repeat;
}
```

#### 6) Rounded Corners

Στις προηγούμενες εκδόσεις του CSS, η χρήση στρογγυλεμένων γωνιών ήταν μια χρονοβόρα διαδικασία καθώς κάθε φορά ήμασταν υποχρεωμένοι να ‘κόβουμε’ τέσσερις κυκλικές εικόνες και να τις τοποθετούμε σε κάθε γωνία του div ώστε να παραχθεί το εφέ. Πλέον, αυτό το χαρακτηριστικό προστίθεται με τη χρήση μιας γραμμής CSS. Αυτό που χρειάζεται είναι να ορίσουμε την ακτίνα του κύκλου για κάθε γωνία.

```
body{
    border-radius: 10px;
}
```

#### 7) Text shadow

Ο κύριος στόχος αυτής της ιδιότητάς είναι να δώσει στους σχεδιαστές ένα εργαλείο για να δημιουργήσουν εφέ κειμένου, το οποίο θα μπορούσε να γίνει πριν μόνο με την χρήση εικόνων. Αυτό έχει το βασικό πλεονέκτημα ότι το κείμενο σε αντίθεση με πριν, παραμένει ανιχνεύσιμο από τις μηχανές αναζήτησης. Επίσης εξοικονομούνται πόροι, καθώς οι εικόνες συνήθως καταλαμβάνουν περισσότερο χώρο, από μία γραμμή κώδικα.

```
h1 {
    text-shadow: 1px 1px 1px rgba(255,255,255, 1);
}
```

## 3.2 PHP(PHP: Hypertext Preprocessor)

Η Php είναι μια scripting γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως στο διαδίκτυο, καθώς οι περισσότερες δυναμικές ιστοσελίδες είναι γραμμένες σε αυτή τη γλώσσα. Τρέχει σε ένα συμβατό διακομιστή και δημιουργεί το δυναμικό μέρος μιας ιστοσελίδας και σε συνεργασία με την Html δημιουργείται το τελικό αρχείο που θα προβληθεί από τον εκάστοτε browser που θα ζητήσει την εμφάνιση της ιστοσελίδας.



Εικόνα 16 Λογότυπο php

```
<!DOCTYPE html>
<html>
  <body>
    <?php echo "This is a Php string"; ?>
  </body>
</html>
```

«Ένα από τα πιο απλά Php scripts που μπορεί να γραφτεί.»

Πλεονεκτήματα της Php:

- 1) Ευκολία στη χρήση, Η δομή της γλώσσας είναι επίπεδες φτιαγμένη για να εξασφαλίζει την ευκολία στη χρήση και την ευκολότερη εκμάθηση από αρχάριους χρήστες.
- 2) Διαχείριση πόρων. Η php καταναλώνει λιγότερους πόρους στο σύστημα που φιλοξενείται, σε σχέση με άλλες γλώσσες που χρησιμοποιούνται για τον ίδιο λόγο. Έτσι συμφέρει περισσότερο αφού δεν χρειάζεται διακομιστή με εκτεταμένους πόρους.
- 3) Ανοιχτός Κώδικας. Αυτό πρακτικά σημαίνει ότι είναι δωρεάν και ότι υπάρχει μια κοινότητα προγραμματιστών που την υποστηρίζει.
- 4) Υποστηρίζει πολλές διαφορετικές βάσεις δεδομένων.
- 5) Υποστηρίζει πληθώρα από βιβλιοθήκες που της δίνουν πάρα πολλές δυνατότητες

## 3.3 SQL (Structured Query Language)

Η SQL είναι μια γλώσσα που χρησιμοποιείται για τη διαχείριση βάσεων δεδομένων και η δομή της οργανώνεται με ερωτήματα. Μέσω αυτών των ερωτημάτων, η SQL έχει τη δυνατότητα της ανάκτησης και ενημέρωσης των δεδομένων μιας βάσης, ενώ έχει την δυνατότητα να δημιουργεί, να

τροποποιεί και να σβήνει πίνακες που αποθηκεύονται δεδομένα και να τροποποιεί τις σχέσεις μεταξύ αυτών.

```
SELECT * from users WHERE id=1
```

«Απλό SQL ερώτημα σε μια βάση»



Εικόνα 17 Λογότυπο Mysql

## MySQL

Η MySQL είναι το πιο δημοφιλές και αξιόπιστο σύστημα διαχείρισης βάσεων δεδομένων. Συνεργάζεται άμεσα με την PHP και μέσω ερωτημάτων της γλώσσας SQL δίνει την δυνατότητα στον χρήστη να δημιουργήσει μια βάση δεδομένων και να διαχειριστεί τα δεδομένα που είναι αποθηκευμένα σε αυτήν.

### Πλεονεκτήματα:

- 1) Μεγάλη απόδοση.
- 2) Ανοιχτός κώδικας και δωρεάν χρήση.
- 3) Συμβατή με όλα τα δημοφιλή λειτουργικά συστήματα και πλατφόρμες.
- 4) Δυνατότητα απομακρυσμένης σύνδεσης για διαχείριση μέσω δικτύου.
- 5) Ταυτόχρονη χρήση από πολλαπλούς χρήστες.

## 3.4 JavaScript



Αν και τα τελευταία χρόνια η JavaScript τρέχει και σε αρκετές εφαρμογές από την πλευρά του διακομιστή, η βασική της λειτουργία για χρόνια, ήταν μια client – side γλώσσα προγραμματισμού (σε αντίθεση με την php) και εκτελούνταν αποκλειστικά στους browsers. Σκοπός την είναι να διαμορφώνει δυναμικά τα html αρχεία και να τους δίνει πρόσθετες δυνατότητες.

Το πλεονέκτημά της σε σχέση με τις γλώσσες που εκτελούνται σε ένα server είναι η δυνατότητα της ασύγχρονης φόρτωσης δεδομένων, δηλαδή η δυναμική αλλαγή ενός μέρους μιας ιστοσελίδας χωρίς να χρειαστεί η ανανέωση ολόκληρης της σελίδας. Έτσι εξοικονομούνται πόροι του συστήματος και η εξυπηρέτηση των χρηστών γίνεται με μεγαλύτερη ταχύτητα.

```
<script type="text/javascript">  
alert("This is a Javascript String");  
</script>
```

«Δείγμα κώδικα JavaScript»

## 3.5 jQuery



Εικόνα 18 Λογότυπο jQuery

Το jQuery είναι μια βιβλιοθήκη της JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client side) της HTML. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig) και σήμερα χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοσελίδων με τη μεγαλύτερη επισκεψιμότητα παγκοσμίως. Επίσης είναι σημαντικό να αναφέρουμε ότι είναι ένα λογισμικό ανοιχτού κώδικα και διανέμεται δωρεάν για κάθε χρήση.

Αυτή η βιβλιοθήκη έχει πολύ μικρό μέγεθος ώστε να μην επιβαρύνει τους browsers, και μας δίνει τη δυνατότητα κάνουμε πολλά πράγματα με λίγο κώδικα. Δηλαδή έχει τη φιλοσοφία «write less, do more». Ο σκοπός του jQuery είναι να μας διευκολύνει να προσθέσουμε κώδικα JavaScript στην ιστοσελίδα μας. Το jQuery αναλαμβάνει να κάνει πολλές από τις συνηθισμένες διαδικασίες που απαιτούν πολλές γραμμές κώδικα JavaScript. Επίσης το jQuery μας δίνει συναρτήσεις με τις οποίες μπορούμε να κάνουμε κάποιες διαδικασίες με μόνο μια γραμμή κώδικα.

Το jQuery επίσης έχει απλοποιήσει πάρα πολλές από τις περίπλοκες διαδικασίες της JavaScript, όπως είναι για παράδειγμα οι κλήσεις AJAX καθώς και ο χειρισμός του μοντέλου DOM.

Συγκεκριμένα η βιβλιοθήκη του jquery περιλαμβάνει τα ακόλουθα χαρακτηριστικά:

- 1) Χειρισμό του html και του DOM
- 2) Χειρισμό του CSS
- 3) Μεθόδους γεγονότων του HTML
- 4) Εφέ και animations
- 5) AJAX
- 6) Plug-ins

### 7) Διάφορα άλλα utilites

Για να συμπεριλάβουμε την βιβλιοθήκη στα Html αρχεία μας θα πρέπει να χρησιμοποιήσουμε τον παρακάτω κώδικα κατά προτίμηση μέσα στο head tag του αρχείου.

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

Ενώ για να δώσουμε τις διάφορες εντολές θα πρέπει να τις συμπεριλάβουμε σε ένα αρχείο JavaScript που να περιλαμβάνει τα εξής:

```
$( document ).ready(function() {  
  // εντολές jQuery  
});
```

Στην συγκεκριμένη εφαρμογή που φτιάχνουμε σε αυτήν την πτυχιακή εργασία η βιβλιοθήκη του jQuery είναι ένα αναπόσπαστο μέρος, καθώς χρησιμοποιείται σε διάφορα σημεία και βελτιώνει την εμφάνιση, την χρηστικότητα και την ταχύτητα της εφαρμογής.

Συγκεκριμένα την χρησιμοποιούμε στα εξής σημεία:

- 1) Στην εμφάνιση και την απόκρυψη των dropdown menus.
- 2) Στην προσαρμογή των μενού ανάλογα με το μέγεθος της οθόνης προβολής.
- 3) Στην συμπλήρωση της μπάρας που δείχνει την πρόοδο της φόρτωσης, στο πάνω μέρος της σελίδας.
- 4) Στην εμφάνιση των αποτελεσμάτων στη σελίδα της αναζήτησης, όταν πατήσουμε το κουμπί της αναζήτησης, είτε αλλάξουμε την επιλογή κάποιου φίλτρου πχ την κατηγορία ή την ημερομηνία του εγγράφου. Σε αυτό το σημείο γίνεται χρήση κλήσεων AJAX για να τροποποιηθεί μόνο ένα μέρος της σελίδας και να μην επαναφορτωθεί ολόκληρη η σελίδα και έτσι βελτιστοποιείται ο χρόνος εμφάνισης των αποτελεσμάτων.
- 5) Η τεχνολογία AJAX χρησιμοποιείται επίσης και στην εμφάνιση των αυτόματων αποτελεσμάτων που ταιριάζουν με την αναζήτησή μας, καθώς πληκτρολογούμε στη φόρμα της αναζήτησης. Αυτό γίνεται με συνεχή queries στη βάση δεδομένων κάθε φορά που πατάμε κάποιο πλήκτρο.
- 6) Στην εμφάνιση και την απόκρυψη των λεπτομερειών των αποτελεσμάτων αναζήτησης.
- 7) Στην σελίδα της τρισδιάστατης βιβλιοθήκης γίνεται χρήση της βιβλιοθήκης σε αρκετά σημεία, με σημαντικότερα:
  - την απόκρυψη των περιττών στοιχείων κατά τη φόρτωση, όπως ο footer και ο header όταν δεν υπάρχει ο κέρσορας πάνω στα μενού,
  - στην σύνδεση της τρισδιάστατης βιβλιοθήκης με τα περιεχόμενα της εφαρμογής. Αυτό γίνεται και πάλι με κλήσεις AJAX και απάντηση από την δυναμική πλευρά της εφαρμογής με κωδικοποιημένα αρχεία JSON που περιέχουν τις απαραίτητες πληροφορίες για προβολή.

## 3.6 WebGL



Εικόνα 19 Λογότυπο Webgl

Το WebGL είναι μια σχετικά καινούρια τεχνολογία που χρησιμοποιείται για την προβολή τρισδιάστατων γραφικών μέσα από ένα web browser. Αν και ακόμα δεν υποστηρίζεται από όλους τους browsers και τις συσκευές που κυκλοφορούν, έχει γίνει αρκετά δημοφιλές λόγω της ευκολίας στη χρήση, χωρίς την ανάγκη εγκατάστασης πρόσθετων εφαρμογών. Το WebGL σε συνεργασία με την Html5 προσφέρει μια ολοκληρωμένη λύση 3D αλλά και 2D προβολής και έρχεται να αντικαταστήσει το απαρχαιωμένο flash καθώς προσφέρει μια σειρά από πλεονεκτήματα.

**Μερικά από αυτά είναι:**

- 1) Τρέχει στην πλευρά του client και έτσι κάνει χρήση της επεξεργαστικής ισχύος της κάρτας γραφικών (GPU)
- 2) Η μίξη με άλλα html στοιχεία.
- 3) Η διαχείριση μέσω JavaScript.

```
<body onload="start()">
<canvas id="glcanvas">
  Your browser doesn't appear to support the HTML5
</canvas>
</body>
```

**“Html5 markup για να ενσωματώσουμε το WebGL.”**

## 4. Εργαλεία Λογισμικού

### 4.1 Bootstrap



Εικόνα 20 Λογότυπο bootstrap

Το Bootstrap είναι ένα front-end framework που δίνει την δυνατότητα στους σχεδιαστές, να ελαχιστοποιήσουν τον χρόνο δημιουργίας μιας ιστοσελίδας από την αρχή. Αυτό επιτυγχάνεται με τη χρήση των ενσωματωμένων στοιχείων που παρέχει και είναι απαραίτητα για τον σχεδιασμό μιας σύγχρονης ιστοσελίδας. Περιέχει αρχεία html, Css, JavaScript καθώς και εικόνες

#### Τι περιέχει το bootstrap:

- 1) Εικονίδια
- 2) Αναπτυσσόμενα μενού
- 3) Κουμπιά διαφόρων χρωμάτων και μεγεθών
- 4) Φόρμες αποστολής
- 5) Μενού περιήγησης
- 6) Σελιδοποίηση
- 7) Ετικέτες
- 8) Έτοιμη δομή σελίδων
- 9) Κεφαλίδες
- 10) Προεπισκόπηση εικόνων
- 11) Ειδοποιήσεις
- 12) Μπάρες προόδου
- 13) Λίστες
- 14) Καρτέλες
- 15) Εφέ JavaScript

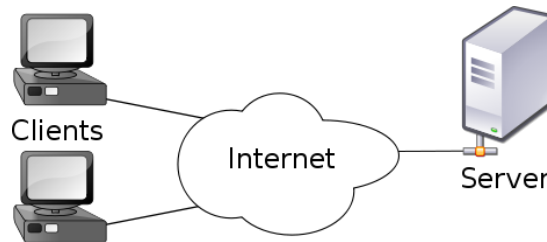
### 4.2 Web Server Apache

APACHE  
HTTP SERVER



Εικόνα 21 Λογότυπο Apache server

Κάθε φορά που ένας χρήστης του διαδικτύου επιθυμεί πχ να προβάλει μια ιστοσελίδα πληκτρολογεί την διεύθυνση της στον browser του και αυτός με τη σειρά του κάνει ένα αίτημα μέσω του πρωτοκόλλου http στον αποκρουσμένο διακομιστή στον οποίο αντιστοιχεί η διεύθυνση της ιστοσελίδας. Ένα από τα δημοφιλέστερα λογισμικά που αναλαμβάνουν να διαχειριστούν αυτά τα αιτήματα από την πλευρά του διακομιστή και να αποστείλουν πίσω τα αρχεία που θα εμφανίσουν την ιστοσελίδα, είναι ο Apache Http server. Το λογισμικό αυτό συνεργάζεται με τις εγκατεστημένες γλώσσες προγραμματισμού και του συστήματος διαχείρισης βάσεων δεδομένων ώστε να δημιουργήσει το αποτέλεσμα της δυναμικής ιστοσελίδας ως ένα html αρχείο. Στην συνέχεια το αποστέλλει πίσω στον browser του χρήστη που το αιτήθηκε.



Εικόνα 22 Μοντέλο client - server

## 4.3 WAMP



Εικόνα 23 Λογότυπο wamp

Το Wamp προέρχεται από τις λέξεις Windows, Apache, MySQL και PHP και όπως λέει και το όνομά του, είναι ένα ολοκληρωμένο πακέτο υπηρεσιών και γλωσσών προγραμματισμού έτοιμο να εγκατασταθεί στο λειτουργικό σύστημα των windows. Συνήθως χρησιμοποιείται για τη δημιουργία ενός τοπικού server που θα τρέχουμε την εφαρμογή μας κατά τη διάρκεια της ανάπτυξης της, είτε για μικρού μεγέθους projects που τρέχουν είτε τοπικά είτε μέσω δικτύου.

## 4.4 CakePHP

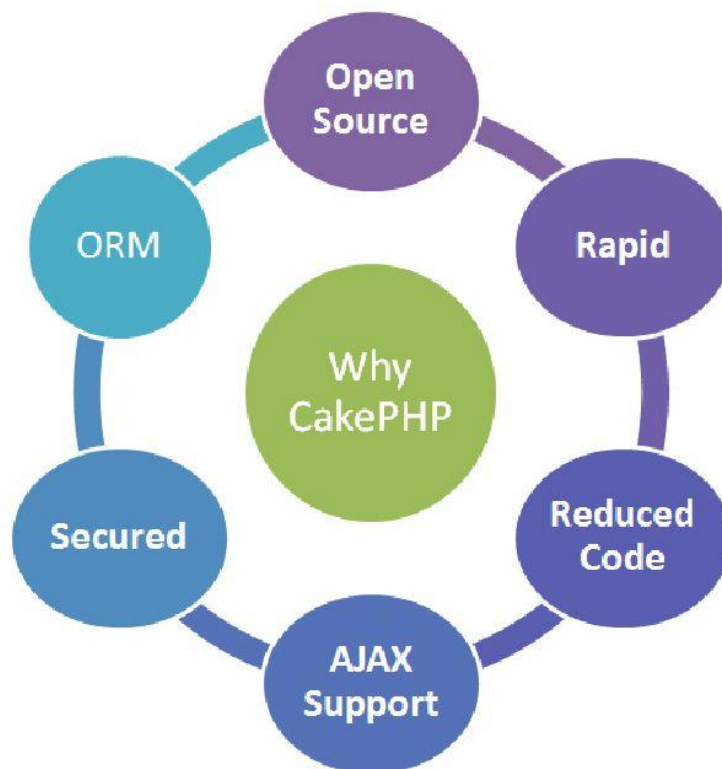
Το Cakerphp είναι ένα από τα πολλά php frameworks που υπάρχουν στο διαδίκτυο και διατίθενται για δωρεάν κατέβασμα και χρήση. Ποια είναι όμως τα χαρακτηριστικά και οι δυνατότητες του που το κάνουν να διαφέρει;





Εικόνα 24 Λογότυπο Cakephp

Το CakePHP είναι μια πλατφόρμα ανοικτού κώδικα για PHP που προορίζετε για να κάνει την δημιουργία, τη δημοσίευση και τη συντήρηση των εφαρμογών ευκολότερη. Επίσης προσφέρει πολλά χρήσιμα μοντέλα σχεδίασης, όπως είναι το MVC (Μοντέλο – Προβολή – Ελεγκτής) που συναντάμε και σε άλλες δημοφιλείς πλατφόρμες. Η πλατφόρμα CakePHP επίσης παρέχει ένα μεγάλο αριθμό πολύτιμων βιβλιοθηκών που διευκολύνουν τη διαχείριση συχνών εργασιών. Κάποιες βασικές από αυτές είναι η σύνδεση με την βάση δεδομένων και η διαχείριση των δεδομένων που περιέχονται σε αυτή, όπως και η σύνδεση των διαφόρων κομματιών, ώστε να αποφεύγεται η επανάληψη του ίδιου κώδικα.



Εικόνα 25 Πλεονεκτήματα CakePhp

#### Μερικά στοιχεία που το κάνουν να ξεχωρίζει

1. Είναι ανοικτού κώδικα με την άδεια χρήσης του MIT και επιτρέπει να χρησιμοποιείτε το CakePHP ελεύθερα.

2. Συμβατό με την PHP4 και PHP5. Η πιο παλαιά έκδοση που είναι συμβατή με το CakePHP είναι η 4.3.2
3. Υποστηρίζει πολλά συστήματα διαχείρισης βάσεων δεδομένων.
4. Υποστηρίζει το μοντέλο σχεδίασης MVC (Model View Control)
5. Εύκολη σύνδεση με τη βάση δεδομένων για δημιουργία, ανάγνωση, αναβάθμιση και διαγραφή εγγραφών.
6. Προσχέδια για να εξοικονομείται χρόνος κατά την διαδικασία ανάπτυξης.
7. Φιλικά URL για τις μηχανές αναζήτησης
8. Εργαλεία για επικύρωση και καθαρισμό των δεδομένων κατά την είσοδο για να είναι πιο ασφαλείς οι εφαρμογές.
9. Είναι άμεσα επεκτάσιμη με plug-in, δηλαδή έτοιμα κομμάτια που γράφονται από τρίτους και δίνουν πρόσθετες δυνατότητες στην εφαρμογή μας.
10. Είναι πολύ δημοφιλές, που σημαίνει ότι έχει μεγάλη κοινότητα υποστήριξης αλλά και ότι αναβαθμίζεται πολύ συχνά για να καλύπτει όλες τις τελευταίες προδιαγραφές της τεχνολογίας και της ασφάλειας.
11. Είναι σχεδιασμένο με βάση την απλότητα στη χρήση και ενδείκνυται ακόμα και για αρχάριους με την php χρήστες, αφού βοηθάει τους προγραμματιστές να γράψουν λιγότερο κώδικα.
12. Διαθέτει αυτόματη λειτουργία δημιουργίας του κώδικα των μοντέλων παίρνοντας ως είσοδο τη δομή των πινάκων που βρίσκονται στην βάση δεδομένων, που είναι ένα από τα πιο χρήσιμα χαρακτηριστικά του, αφού έτσι εξοικονομούμε πολύ χρόνο.

## 5. Σχέδιο Δράσης για την εκπόνηση της εργασίας

Σε αυτό το κεφάλαιο εξετάσουμε όλους τους δυνατούς τρόπους υλοποίησης μιας δυναμικής εφαρμογής όπως αυτή που θέλουμε να φτιάξουμε, και θα αξιολογήσουμε όλα τα διαθέσιμα εργαλεία μέχρι να αποφασίσουμε ποια είναι πιο αποτελεσματικά και ποια τελικά ταιριάζουν καλύτερα στις ανάγκες μας.

Πίνακας 3 Συσχέτιση στόχων, μεθοδολογίας και βιβλιογραφίας

Στόχος	Μεθοδολογία	Βιβλιογραφία
Εμφάνιση Ιστοσελίδας (front-end)	Photoshop, Html & Css, Bosttrap framework	1, 2, 5, 12
Δυναμικό κομμάτι (back-end)	PHP μέσω CakePHP framework	6, 11
Βάση δεδομένων	MySQL	7
Διακομιστής	Apache Server μέσω Wamp Server	10
Βελτίωση απόδοσης	Javascript & Jquery	8

Μετά από την προσεκτική ανάλυση όλων των δυνατών επιλογών προκύπτει ο παραπάνω πίνακας που παρουσιάζει τη συσχέτιση των επιμέρους στόχων που θέλουμε να πετύχουμε με τις βέλτιστες λύσεις. Στα επόμενα υποκεφάλαια θα δούμε αναλυτικά γιατί επιλέξαμε αυτές τις λύσεις, χωρισμένα σε δυο κομμάτια το στατικό και το δυναμικό μέρος.

### 5.1 Επιλογή τρόπου σχεδίασης της εμφάνισης της ιστοσελίδας



Εικόνα 26 Λογότυπο Photoshop

Για να ξεκινήσουμε να σχεδιάζουμε την εμφάνιση και τα χαρακτηριστικά μιας ιστοσελίδας από την αρχή, ο πιο σωστός τρόπος είναι η χρήση κάποιου προγράμματος σχεδίασης. Στην αγορά υπάρχει μια πληθώρα προγραμμάτων που είτε διακινούνται ελεύθερα, είτε με κάποιο αντίτιμο. Για την σχεδίαση της εμφάνισης μιας ιστοσελίδας το πιο δημοφιλές και πρακτικό λογισμικό είναι το

Adobe Photoshop που δεν είναι τυχαία η πρώτη επιλογή των επαγγελματιών που ασχολούνται με το κλάδο των γραφικής σχεδίασης.

Εκεί σχεδιάζουμε τις διάφορες σελίδες, το logo, τα μενού, τις φόρμες και γενικότερα την εικόνα της ιστοσελίδας που θα φτιάξουμε. Αυτό βοηθάει ώστε να φτιάξουμε μια πρωτότυπη εμφάνιση αλλά και να δούμε πως ταιριάζουν μεταξύ τους τα διάφορα στοιχεία που θέλουμε να προσθέσουμε στην εφαρμογή.

Στην ουσία δημιουργούμε μια πρώτη εικόνα για την ιστοσελίδα που θα φτιάξουμε, γνωρίζοντας ότι κατά την ολοκλήρωσή της ίσως αυτή να διαφέρει, καθώς οι ανάγκες που προκύπτουν κατά τη διάρκεια της υλοποίησης, μπορούν να αλλάξουν και αντίστοιχα να αναπροσαρμοστεί και η εικόνα της εφαρμογής. Η δημιουργία της αρχικής εικόνας είναι επίσης ιδιαίτερα χρήσιμη, ειδικά όταν συνεργάζονται παραπάνω από ένα άτομα για μια εφαρμογή, είτε ένας σχεδιαστής φτιάχνει μια εφαρμογή για κάποιο πελάτη και έτσι θα μπορεί να του δείξει πως θα μοιάζει αυτή όταν τελειώσει και να συμφωνηθούν οι αλλαγές.



Εικόνα 27 Μεταφορά psd σε html

Αφού καταλήξουμε στην αρχική εικόνα, θα πρέπει αυτή να μεταφραστεί σε ένα αρχείο που θα μπορεί να διαβαστεί από τους διάφορους browsers. Δηλαδή θα πρέπει να μετατρένουμε τα κομμάτια της εικόνας σε στοιχεία Html και να τα μορφοποιήσουμε με την χρήση του Css. Ή όπως λέγεται αυτό, να «κόψουμε» το αρχείο psd (επέκταση αρχείων Photoshop) και να φτιάξουμε την σελίδα μας. Για να το κάνουμε αυτό υπάρχουν δύο τρόποι που διαφέρουν ως προς την ευχρηστία, τον χρόνο δημιουργίας και την αποτελεσματικότητά τους.

Ο πρώτος τρόπος είναι να αρχίζουμε να γράφουμε από την αρχή τον κώδικα Html και Css, ώσπου η ιστοσελίδα μας να αρχίσει να μοιάζει με την εικόνα που σχεδιάσαμε. Αυτός ο τρόπος είναι ο πιο ακριβής, αφού τροποποιούμε το κάθε στοιχείο σύμφωνα με τους κανόνες που έχουμε βάλει, έχει όμως ένα πολύ βασικό μειονέκτημα. Είναι τρομερά χρονοβόρος καθώς απαιτεί να γράψουμε κανόνες για το κάθε στοιχείο από την αρχή, καθώς και να τσεκάρουμε αναλυτικά την εμφάνιση σε όλους τους browser και τις συσκευές που θα εμφανίζεται η ιστοσελίδα μας.

Ένας δεύτερος τρόπος που λύνει τα μειονεκτήματα του παραπάνω είναι η χρήση ενός front-end framework που μας γλιτώνει αρκετό χρόνο από τη δημιουργία, καθώς παρέχει πολλά έτοιμα κομμάτια και κάποιες λύσεις για εύκολη επαναχρησιμοποίηση κομματιών κώδικα Html και Css. Γνωρίζοντας ότι αυτά τα frameworks είναι σχεδιασμένα για να ικανοποιούν αυτές ακριβώς τις ανάγκες, εξοικονομούμε πολύτιμο χρόνο και από τις δοκιμές σε διαφορετικούς browsers και συσκευές. Επίσης ένα πολύ σημαντικό πλεονέκτημα που έχουμε χρησιμοποιώντας ένα καλό framework, είναι η προς τα πίσω συμβατότητα με παλιότερους browsers αλλά και η αναβάθμιση για να ταιριάζει στις μελλοντικές ανάγκες που αλλάζουν συνεχώς. Τα frameworks έχουν όμως και κάποια μειονεκτήματα ανάλογα με τη χρήση τους. Πχ η υπερβολική χρήση κάποιων στοιχείων που παρέχονται έτοιμα, μπορεί να αποκλίνει πολύ από την αρχική σχεδίαση που είχαμε κάνει, αλλά και να εκμηδενίσει την πρωτοτυπία στην εμφάνιση της ιστοσελίδας μας.

Για την υλοποίηση της εφαρμογής που πραγματεύεται αυτή η εργασία, επιλέξαμε το **Bootstrap Framework** που είναι ένα από τα πιο δημοφιλή frameworks αυτή τη στιγμή και παρέχει ολοκληρωμένες λύσεις όπως περιγράφεται στο κεφάλαιο 2.4. Η επιλογή έγινε με βάση την ευχρηστία

και την ευελιξία του συγκεκριμένου εργαλείου αλλά και την πληθώρα των επιλογών που δίνει στην υλοποίηση.

## 5.2 Επιλογή τρόπου δημιουργίας δυναμικού μέρους

Αφού σχεδιάσουμε την εμφάνιση της εφαρμογής και φτιάξουμε τα αρχεία σε μορφή Html και Css, έχουμε στην ουσία μια στατική ιστοσελίδα. Αυτή θα πρέπει να την μετατρέψουμε σε δυναμική, να της προσθέσουμε δηλαδή τα χαρακτηριστικά που θα της δώσουν λειτουργικότητα μέχρι να εξυπηρετεί το σκοπό για τον οποίο φτιάχτηκε. Αυτό θα γίνει με την ενσωμάτωση δυναμικού κώδικα της γλώσσας προγραμματισμού που θα επιλέξουμε, μέσα στα στατικά αρχεία. Ατός ο κώδικας πλέον θα πρέπει να τρέχει στο περιβάλλον ενός διακομιστή που θα εκτελεί τις εντολές και θα απαντάει στα διάφορα αιτήματα.

Συγκρίνοντας τα διάφορες επιλογές που υπάρχουν και καταγράφοντας τις ανάγκες που πρέπει να εξυπηρετεί η γλώσσα που θα επιλέξουμε, καταλήξαμε στην επιλογή της **Php** για μία σειρά από λόγους:

- 1) Είναι γλώσσα ανοιχτού κώδικα και διαθέσιμη δωρεάν σε αντίθεση με την Asp.net.
- 2) Χρειάζεται λιγότερους πόρους και συνεργάζεται με περισσότερους διακομιστές από ότι οι περισσότερες κορυφαίες γλώσσες, όπως η Java.
- 3) Είναι εύκολη στη εκμάθηση και τη χρήση.
- 4) Είναι η πιο οικονομική και τρέχει πιο γρήγορα εφόσον καταναλώνει λιγότερους πόρους.
- 5) Είναι η πιο δημοφιλής και διαδεδομένη από όλες τις υπόλοιπες και διαθέτει πολύ μεγάλη κοινότητα υποστήριξης.
- 6) Διαθέτει μια σειρά από frameworks που απλοποιούν την χρήση της και μειώνουν τον κώδικα που πρέπει να γραφτεί.

Όπως αναφέρουμε και στα πλεονεκτήματα της, η Php διαθέτει πολλά frameworks που μπορούμε να χρησιμοποιήσουμε για να επιταχύνουμε τη δημιουργία της εφαρμογής μας. Οπότε καλούμαστε να αποφασίσουμε αν θα χρησιμοποιήσουμε κάποιο από αυτά, ή αν θα γράψουμε τον κώδικα της Php από την αρχή. Με βάση τα πλεονεκτήματα που περιγράφουμε στα κεφάλαιο 2.2.5 και 4.3 επιλέξαμε τη χρήση του framework Cakerhp για την υλοποίηση του δυναμικού μέρους της εφαρμογής μας.

Το επόμενο στάδιο είναι να επιλέξουμε ένα συμβατό σύστημα διαχείρισης βάσεων δεδομένων από τις διαθέσιμες επιλογές που συνεργάζονται με το Cakerhp.

**Οι συμβατές λύσεις είναι οι παρακάτω:**

- MySQL (4 ή νεότερο)
- PostgreSQL
- Microsoft SQL Server
- Oracle
- SQLite

Αυτό που θα επιλέξουμε τελικά είναι η λύση του MySQL με βάση τα πλεονεκτήματα που περιγράφονται αναλυτικά στο κεφάλαιο 3.3, αλλά και η πρόθεση μας να χρησιμοποιήσουμε τον Wamp Server που είναι ένα λογισμικό που εγκαθιστά στον υπολογιστή μας τοπικά, ένα πακέτο που περιέχει όλες τις τεχνολογίες που θα μας χρειαστούν και μέσα σε αυτές και το MySQL.

Τέλος καλούμαστε να επιλέξουμε μια scripting γλώσσα που θα εκτελείται στο browser και θα επιταχύνει την διαδικασία της αναζήτησης μέσω Ajax requests, αλλά και θα διαμορφώνει έτσι τη

Πτυχιακή εργασία  
Κωνσταντίνος Θεοφίλης

ΤΕΙ ΚΡΗΤΗΣ Τμήμα μηχανικών πληροφορικής

σελίδας μας ώστε να είναι πιο εύχρηστη. Έτσι οδηγούμαστε στον μονόδρομο της χρήσης της JavaScript που κάνει ακριβώς αυτό που θέλουμε με γρήγορο και εύκολο τρόπο.

## 6. Ανάπτυξη του συστήματος

### 6.1 Απαιτήσεις Συστήματος

Οι απαιτήσεις του συστήματος που θα φιλοξενεί την εφαρμογή μας καθορίζονται από τα εργαλεία που χρησιμοποιούμε και τις γλώσσες προγραμματισμού που επιλέξαμε να γράψουμε την εφαρμογή. Δεδομένου ότι χρησιμοποιούμε το Cakephp framework, οι απαιτήσεις του συστήματος διαμορφώνονται ως εξής:

- Χρήση ενός HTTP διακομιστή. Για παράδειγμα ο Apache είναι ο ιδανικός αφού υποστηρίζει και εντολές mod\_rewrite για να παρουσιάζονται οι urls πιο σωστά.
- PHP 4.3.2 ή νεότερη. Το Cakephp τρέχει στις εκδόσεις 4 και 5 της Php
- Συστήματα διαχείρισης βάσης δεδομένων MySQL (4 ή νεότερο)

### 6.2 Υλοποίηση

#### 6.2.1 Σχεδίαση

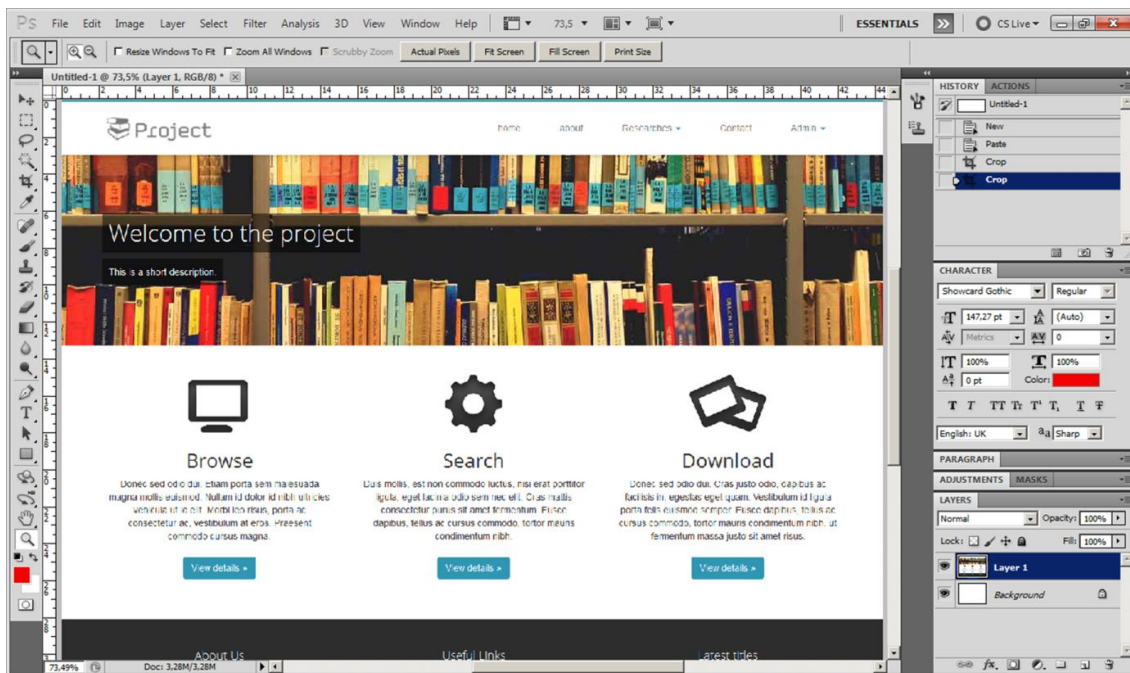
Όπως περιγράψαμε και στο προηγούμενο κεφάλαιο ξεκινάμε με την σχεδίαση στο Photoshop, της εικόνας που θέλουμε να έχει η ιστοσελίδα μας.

Ξεκινάμε με την δημιουργία του logo:



Εικόνα 28 Το λογότυπο της εφαρμογής μας

Συνεχίζουμε με την σχεδίαση της αρχικής σελίδας.

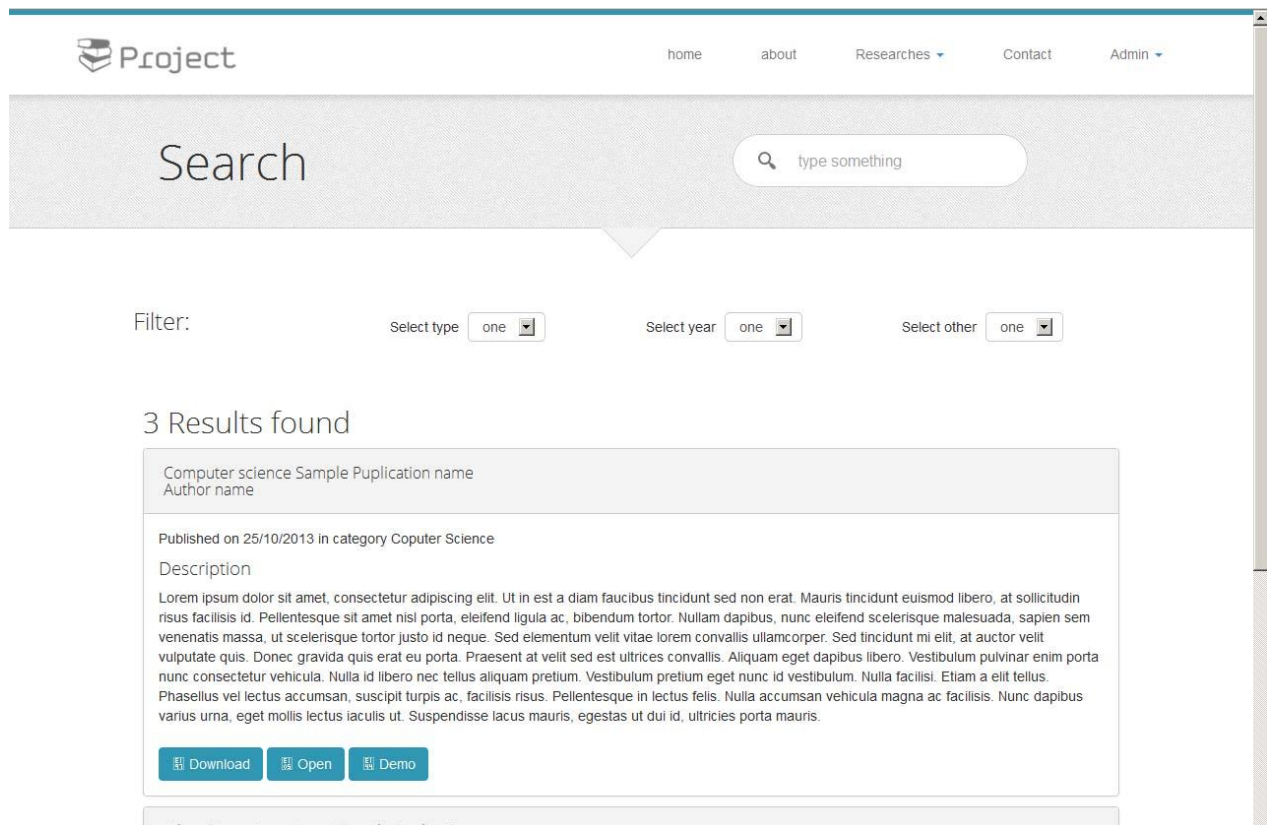


Εικόνα 29 Η σχεδίαση της αρχικής σελίδας της εφαρμογής στο Photoshop

## 6.2.2 Δημιουργία αρχείων Html & Css.

Με τη χρήση του Bootstrap framework φτιάξαμε τα διάφορα στοιχεία της σελίδας. Αυτά που παραμένουν σταθερά σε όλες τις σελίδες, δηλαδή ο header που είναι το πάνω κομμάτι, ο footer που είναι το κάτω και το φόντο. Αλλά και τα κομμάτια από τις βασικότερες σελίδες που αλλάζουν από σελίδα σε σελίδα και στη συνέχεια τα ενώσαμε σε ένα, ώστε να βγει η τελική μορφή μιας στατικής σελίδας μας.

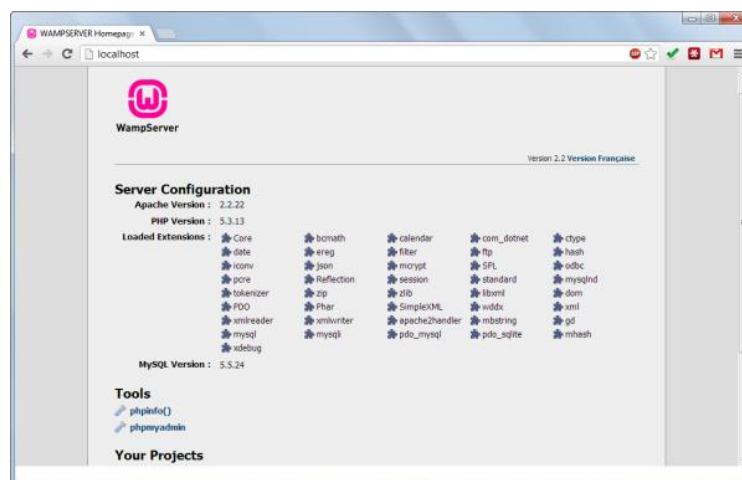




Εικόνα 30 Σχεδίαση σελίδας αναζήτησης σε Html και Css

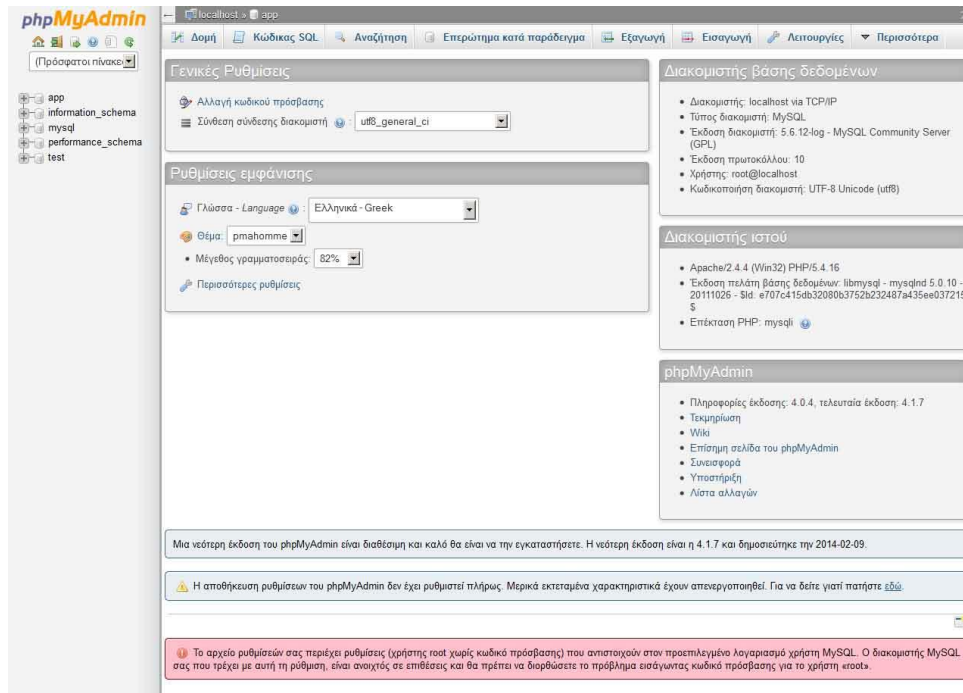
## 6.2.3 Εγκατάσταση του Server

Στη συνέχεια προχωράμε στο δυναμικό κομμάτι της εφαρμογής και ξεκινάμε εγκαθιστώντας τον Wamp Server τοπικά στον υπολογιστή που θα χρησιμοποιήσουμε για την κατασκευή.



Εικόνα 31 Εικόνα μετά την επιτυχή εγκατάσταση του wamp

Ακολουθούμε τις αναλυτικές οδηγίες για την εγκατάσταση του λογισμικού Wamp μέχρι να δούμε την παραπάνω εικόνα όταν πληκτρολογήσουμε στον browser την διεύθυνση που εγκαταστήσαμε τον web server. Επίσης αν εγκατασταθεί σωστά το σύστημα διαχείρισης βάσεων δεδομένων phpMyAdmin, βλέπουμε την παρακάτω εικόνα.

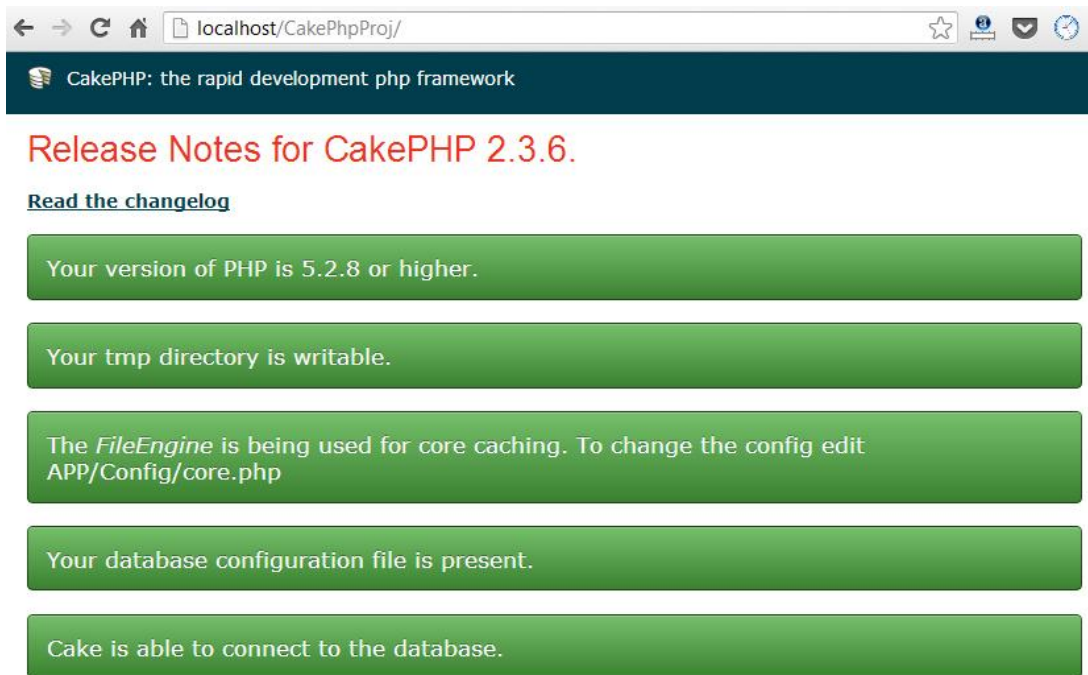


Εικόνα 32 Αρχική σελίδα phpMyAdmin

## 6.2.4 Εγκατάσταση του Cakephp Framework

Ακολουθούμε τις οδηγίες για εγκαταστήσουμε το CakePhp στον web server που μόλις εγκαταστήσαμε:

- 1) Κατεβάζουμε την τελευταία έκδοση από την επίσημη ιστοσελίδα **cakephp.org**
- 2) Αποσυμπιέζουμε τα αρχεία στον επιθυμητό φάκελο εγκατάστασης.
- 3) Δημιουργούμε μια καινούρια βάση δεδομένων στο phpMyAdmin και επεξεργαζόμαστε το αρχείο **database.php** ώστε να συνδέσουμε την βάση δεδομένων με το framework.
- 4) Ρυθμίζουμε τα **mod\_rewrites** του server, μέχρι να εμφανίζονται σωστά όλα τα στοιχεία της ιστοσελίδας και να βλέπουμε την παρακάτω εικόνα:

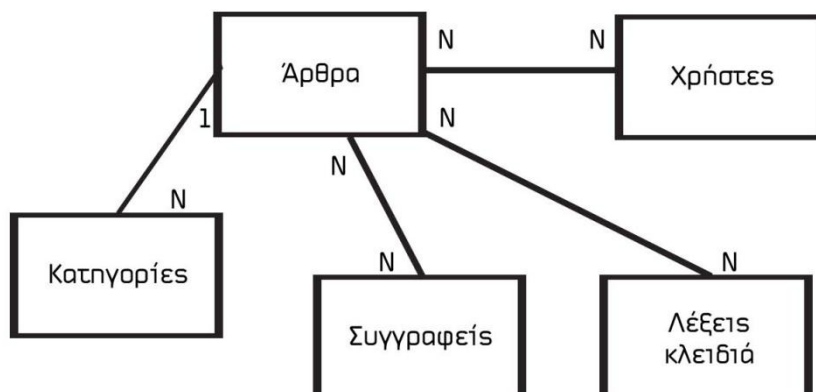


Εικόνα 33 Αρχική εικόνα Cakerhp

## 6.2.5 Υλοποίηση της βάσης δεδομένων

Αφού εγκαταστήσουμε την τελευταία έκδοση της Cakerhp χωρίς σφάλματα, είμαστε έτοιμοι να αρχίσουμε την υλοποίηση της βάση δεδομένων με τους πίνακες και τις συσχετίσεις των διαφόρων πεδίων βασισμένα στις οντότητες που έχουμε και στους κανόνες για το χτίσιμο της εφαρμογής με την Cakerhp.

Οι προϋποθέσεις που πρέπει να ακολουθούν οι οντότητές μας είναι οι εξής:



Εικόνα 34 Μοντέλο οντοτήτων συσχετίσεων

- Τα αρχεία του εργαστηρίου (άρθρα) πρέπει να ανήκουν σε μια μόνο κατηγορία.
- Ένα άρθρο πρέπει να έχει έναν ή περισσότερους συγγραφείς.
- Ένα άρθρο μπορεί να έχει πολλές λέξεις κλειδιά (tags), το οποίο μπορεί να αντιστοιχεί σε πολλά άρθρα.

Οι κανόνες που πρέπει να ακολουθεί η δομή και τα ονόματα των πινάκων ώστε να διαχειρίζονται σωστά από την cakerhp είναι οι εξής:

- Η αντιστοίχιση N:N πρέπει να γίνεται σε ένα ξεχωριστό πίνακα που να έχει τα ονόματα των πινάκων που συνδέει χωρισμένα με το σύμβολο \_ και δυο πεδία με τα μοναδικά κλειδιά των πινάκων.
- Τα πεδία created και modified συμπληρώνονται αυτόματα όταν δημιουργείται ή τροποποιείται μια εγγραφή.

Έτσι προκύπτουν οι παρακάτω πίνακες με τον αντίστοιχο κώδικα sql:

#### Authors

id	name	Title	Contact	email	linkedin	created	modified
<pre>CREATE TABLE `authors` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `name` varchar(255) CHARACTER SET utf8 NOT NULL,   `title` varchar(255) COLLATE utf8_unicode_ci NOT NULL,   `contact` varchar(500) COLLATE utf8_unicode_ci NOT NULL,   `email` varchar(100) COLLATE utf8_unicode_ci NOT NULL,   `linkedin` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,   PRIMARY KEY (`id`));</pre>							

#### Authors\_posts

id	Post_id	Author_id	created	modified
<pre>CREATE TABLE `authors_posts` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `post_id` int(11) NOT NULL,   `author_id` int(11) NOT NULL,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,   PRIMARY KEY (`id`),   KEY `post_id` (`post_id`),   KEY `author_id` (`author_id`));</pre>				

Post\_id: εξωτερικό κλειδί πίνακα Posts

Author\_id: εξωτερικό κλειδί πίνακα authors

#### Categories

id	name	description	created	Modified
<pre>CREATE TABLE `categories` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `name` varchar(255) NOT NULL,   `description` text,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,   PRIMARY KEY (`id`));</pre>				

### Posts

id	Category_id	title	location	body	link
Pdf_file_name	Date_published	created	modified	published	
<pre>CREATE TABLE `posts` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `category_id` int(11) NOT NULL DEFAULT '0',   `title` varchar(255) NOT NULL,   `location` varchar(255) NOT NULL,   `body` text,   `link` text,   `pdf_file_name` varchar(255) DEFAULT NULL,   `date_published` year(4) DEFAULT NULL,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,   `published` tinyint(1) NOT NULL DEFAULT '0',   PRIMARY KEY (`id`),   KEY `category_id` (`category_id`));</pre>					

Category\_id: εξωτερικό κλειδί πίνακα Categories

### Posts\_tags

id	Post_id	Tag_id	created	Modified
<pre>CREATE TABLE `posts_tags` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `post_id` int(11) NOT NULL,   `tag_id` int(11) NOT NULL,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,   `times_tagged` int(11) NOT NULL DEFAULT '1',   PRIMARY KEY (`id`),   KEY `post_id` (`post_id`),   KEY `tag_id` (`tag_id`));</pre>				

Post\_id: εξωτερικό κλειδί πίνακα Posts

Tag\_id: εξωτερικό κλειδί πίνακα tags

### Tags

id	name	created	Modified
<pre>CREATE TABLE `tags` (   `id` int(11) NOT NULL AUTO_INCREMENT,   `name` varchar(255) CHARACTER SET utf8 NOT NULL,   `created` datetime DEFAULT NULL,   `modified` datetime DEFAULT NULL,</pre>			

```
PRIMARY KEY (`id`));
```

## Users

id	username	password	role	created	Modified
CREATE TABLE `users` ( `id` int(10) unsigned NOT NULL AUTO_INCREMENT, `username` varchar(50) DEFAULT NULL, `password` varchar(50) DEFAULT NULL, `role` varchar(20) DEFAULT NULL, `created` datetime DEFAULT NULL, `modified` datetime DEFAULT NULL, PRIMARY KEY (`id`));					

## 6.2.6 Δημιουργία σελίδων εφαρμογής

Η Cakephp έχει ένα πολύ χρήσιμο εργαλείο που λέγεται Bake και παράγει τον αρχικό κώδικα της εφαρμογής παίρνοντας ως είσοδο την βάση δεδομένων, και από τους πίνακες και τις συσχετίσεις που δημιουργήσαμε πριν, παράγει τα μοντέλα, τους ελεγκτές και την εμφάνιση του καθενός με βάση κάποια πολύ απλά πρότυπα.

```
Bake All
-----
Use Database Config: (default/test)
[default] >
Possible Models based on your current database:
1. Category
2. Job
3. User
Enter a number from the list above,
type in the name of another model, or 'q' to exit
[q] > 1

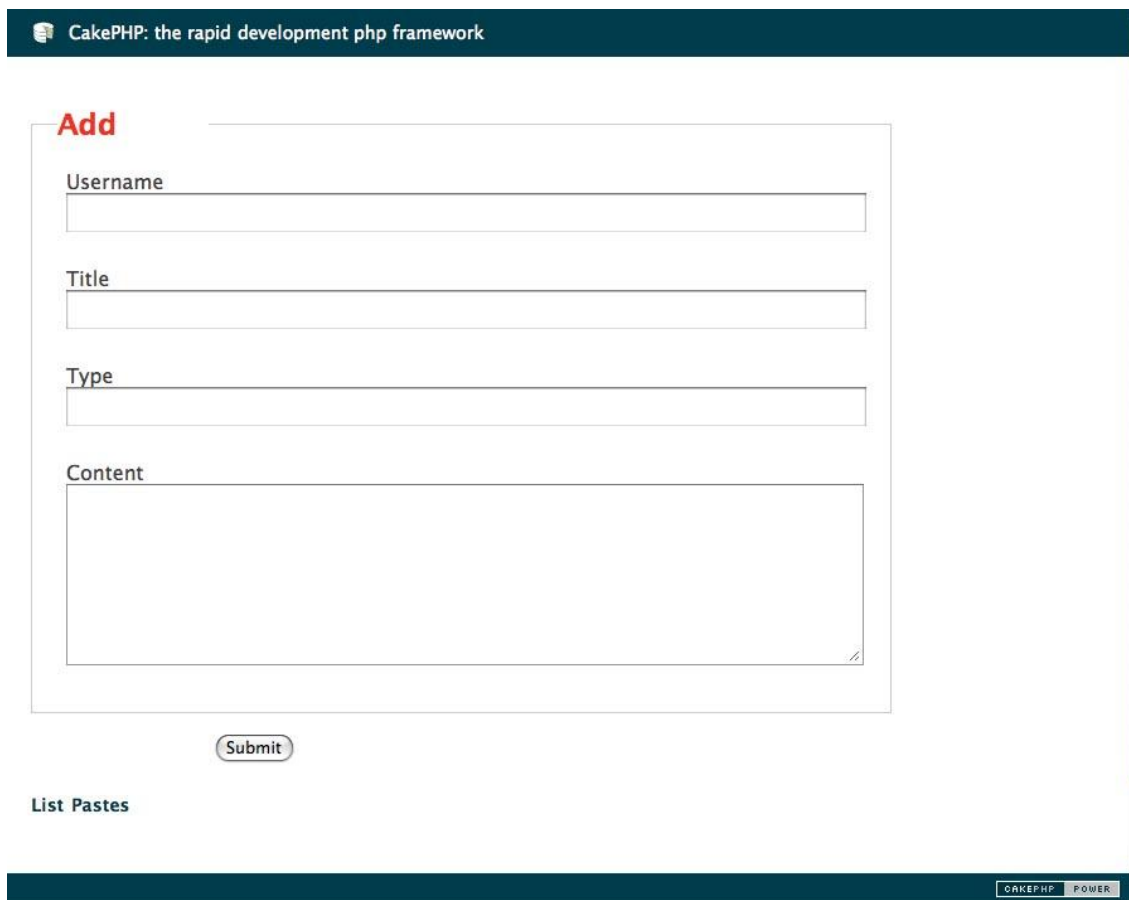
Baking model class for Category...

Creating file C:\wamp\www\privat\codeforest\cake_new\app\models\category.php
Wrote 'C:\wamp\www\privat\codeforest\cake_new\app\models\category.php'

Creating file C:\wamp\www\privat\codeforest\cake_new\app\controllers\categories_
controller.php
Wrote 'C:\wamp\www\privat\codeforest\cake_new\app\controllers\categories_control
ler.php'
Category Controller was baked.
SimpleTest is not installed. Do you want to bake unit test files anyway? (y/n)
[y] >
```

Εικόνα 35 Κονσόλα εντολών

Σε περιβάλλον κονσόλας εντολών των windows, πληκτρολογούμε την εντολή **cake bake all** στον καθορισμένο φάκελο της εφαρμογής και έτσι παράγεται ο κώδικας που στην συνέχεια πρέπει να τροποποιήσουμε ώστε να καλύπτει τις ανάγκες μας.



Εικόνα 36 προεπιλεγμένη εμφάνιση σελίδας

Στη συνέχεια προχωράμε στην τροποποίηση της προεπιλεγμένης εμφάνισης της εφαρμογής και της αντικατάστασης της με την εμφάνιση που δημιουργήσαμε στο κεφάλαιο 6.2.1. Αυτό γίνεται με την επεξεργασία του κώδικα του αρχείου **default.ctp** που είναι υπεύθυνο για τον header και τον footer όλων των σελίδων, επίσης όμως αλλάζουμε και το περιεχόμενο κάθε σελίδας μέχρι να γίνει όπως ακριβώς το σχεδιάσαμε.

## Header

Ο header είναι το κομμάτι του κώδικα που εμφανίζεται στο πάνω μέρος όλων των σελίδων της εφαρμογής και περιέχει το λογότυπο που είναι σύνδεσμος για την αρχική σελίδα καθώς και τα menu που επιτρέπουν στον χρήστη να πλοηγηθεί στην ιστοσελίδα.

```
<div id="top-bar"></div>
<header role="navigation" class="navbar navbar-fixed-top section-
row" id="navbar-main">
  <div class="container">
    <div class="navbar-header">
```

```
<?php echo $this->Html->link( $this->Html->image('logo.png',
array('alt' => __d('cake', 'McLab Research Index')), '/',
array("class" => "navbar-brand", 'escape' => false) );?>
</div>
<div class="navbar-collapse collapse" style="height: 1px;">
  <ul class="nav navbar-nav navbar-right">
    <li> <?php echo $this->Html->link( __d('cake', 'About'),
'/pages/about/', array() );?></li>
    <li class="dropdown"><a data-toggle="dropdown"
class="dropdown-toggle" href="#"><?php echo __d('cake',
'Researches')?><b class="caret"></b></a>
<ul class="dropdown-menu">
  <li><?php echo $this->Html->link( __d('cake', 'Browse'),
'/posts/browse', array() );?></li>
  <li><?php echo $this->Html->link( __d('cake', 'Tag cloud'),
'/tags/browse', array() );?> </li>
  <li><?php echo $this->Html->link( __d('cake', '3d Bookcase'),
'/posts/bookcase/', array() );?> </li>
</ul></li>
    <?php if ($this->Session->read('Auth.User')){?>
      <li class="dropdown"><a data-toggle="dropdown"
class="dropdown-toggle" href="#"><?php echo __d('cake', 'Admin');?>
<b class="caret"></b></a>
      <ul class="dropdown-menu">
        <li><?php echo $this->Html->link( __d('cake',
'Posts'), '/posts/', array());?></li>
        <li> <?php echo $this->Html->link( __d('cake',
'Users'), '/users/', array());?>li>
        <li><?php echo $this->Html->link( __d('cake',
'Authors'), '/authors/', array());?></li>
        <li><?php echo $this->Html->link( __d('cake',
'Categories'), '/categories/', array());?></li>
        <li><?php echo $this->Html->link( __d('cake', 'Tags'),
'/tags/', array());?>li>
        <li><?php echo $this->Html->link("Log out",
'/users/logout/', array());?>li>
      </ul>
    </li> <?php } ?>
  </ul>
</div>
<!--/.nav-collapse -->
</div>
<div id="toggle-button">
</div>
</header>
```



Εικόνα 37 Εμφάνιση του header



## Footer

Είναι το κομμάτι που εμφανίζεται στο κάτω μέρος της κάθε σελίδας της εφαρμογής και περιέχει χρήσιμες πληροφορίες όπως στοιχεία επικοινωνίας με τους διαχειριστές κάποια χρήσιμα link, καθώς και τα τελευταία άρθρα που προστέθηκαν στην εφαρμογή με χρονολογική σειρά.

```
< footer class = "section-row-footer" > < div class = "footer-1" > <
div class = "container" > < div class = "row" >
  < div class = "col-lg-4" > < h2 > <? php echo __d('cake',
'Contact Us') ?> < /h2>
    <div class="clearfix pl">
      <p class="text-info"><i><?php echo __d('cake', 'Lorem ipsum
dolor sit amet, consectetur adipiscing, sed diam nonummy nibh
eui.')?> </i > < /p>
    </div >
      < ul id = "social" class = "tooltip-demo" >
        < li > < a rel = "tooltip" title = "twitter"href
= "#" > twitter < /a></li > < li > < a rel = "tooltip" title =
"facebook"
href = "#" > facebook < /a></li >
      < /ul>
    </div >
  < div class = "col-lg-4" >
    < h2 > <? php echo __d('cake', 'Useful Links') ?> < /h2>
    <ul class="list">
      <li><a href="#">random link</a > < /li>
      <li><a href="#">random link</a > < /li>
      <li><a href="#">random link</a > < /li>
      <li><a href="#">random link</a > < /li>
    </ul >
    < /div>
    <div class="col-lg-4">
      <h2><?php echo __d('cake', 'Latest titles')?></h2 >
      <? php echo $this - > element('latestposts',
array('cache' = > '1 day')); ?>
    < /div>
  </div >
< /div>
</div >
< div class = "footer-2" >
  < div class = "container" >
    < div class = "row" >
      < div class = "span12" > Project copyrights etc <? php
echo $this - > Html - > link("Admin Area", '/users/login/',
array()); ?> < /div>
    </div >
  < /div>
</div >
< /footer>
<?phpif (class_exists('JsHelper') && method_exists($this->Js,
'writeBuffer')) echo $this->Js->writeBuffer(); ?>
```



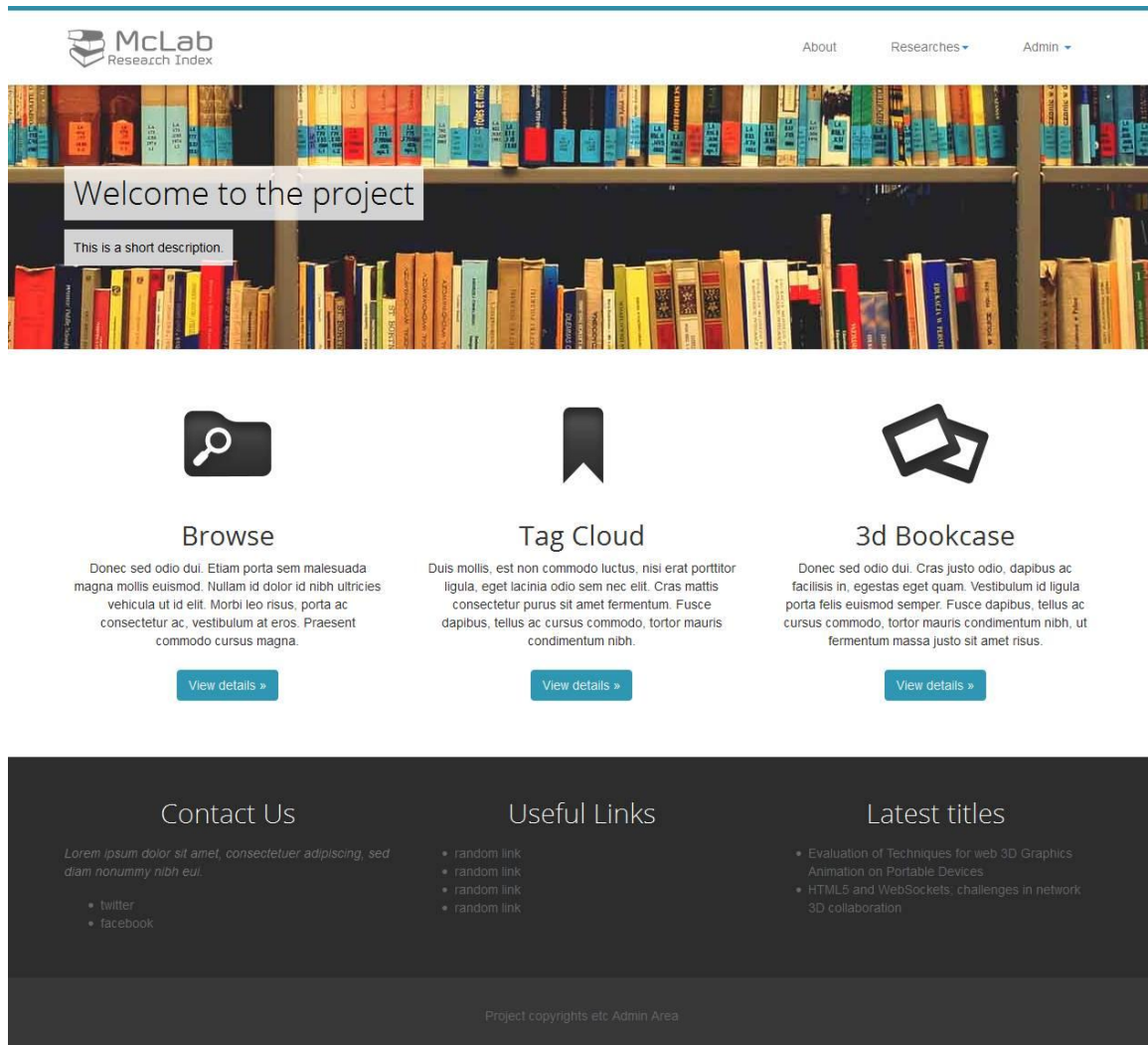
Εικόνα 38 Εμφάνιση footer

## Αρχική σελίδα

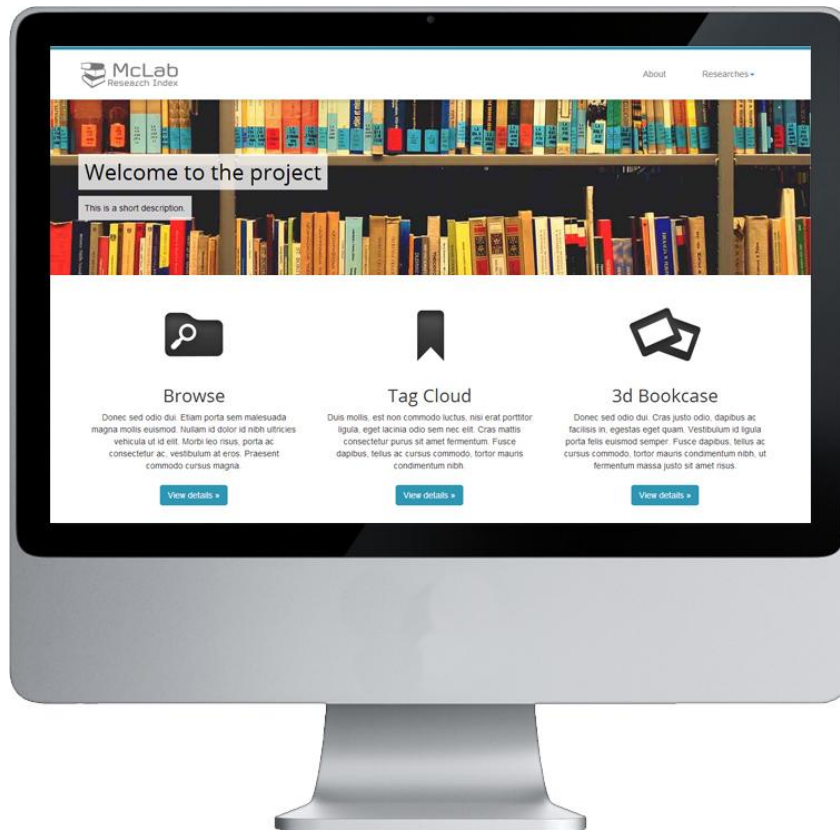
Αυτή η σελίδα είναι η πρώτη εικόνα που αντικρίζει ο επισκέπτης μόλις επισκεφθεί την εφαρμογή μας. Περιέχει μήνυμα καλωσορίσματος και συνδέσμους προς τις βασικές σελίδες της εφαρμογής.

```
<div class="intro">
  <div class="container">
    <h1><?php echo __d('cake', 'Welcome to the project')?></h1>
    <p><?php echo __d('cake', 'This is a short
description.')?></p>
  </div>
</div>
<div class="container features">
  <div class="row">
    <div class="col-lg-4 feature">
      
      <h2>Browse</h2>
      <p>...</p>
      <p><a class="btn btn-green" href="posts/browse"
role="button">View details &raquo;</a></p>
    </div><!-- /.col-lg-4 -->
    <div class="col-lg-4 feature">
      
      <h2>Tag Cloud</h2>
      <p>...</p>
      <p> <a class="btn btn-green" href="tags/browse/"
role="button">View details &raquo;</a></p>
    </div><!-- /.col-lg-4 -->
    <div class="col-lg-4 feature">
      
      <h2>3d Bookcase</h2>
      <p>...</p>
      <p><a class="btn btn-green" href="posts/bookcase/"
```

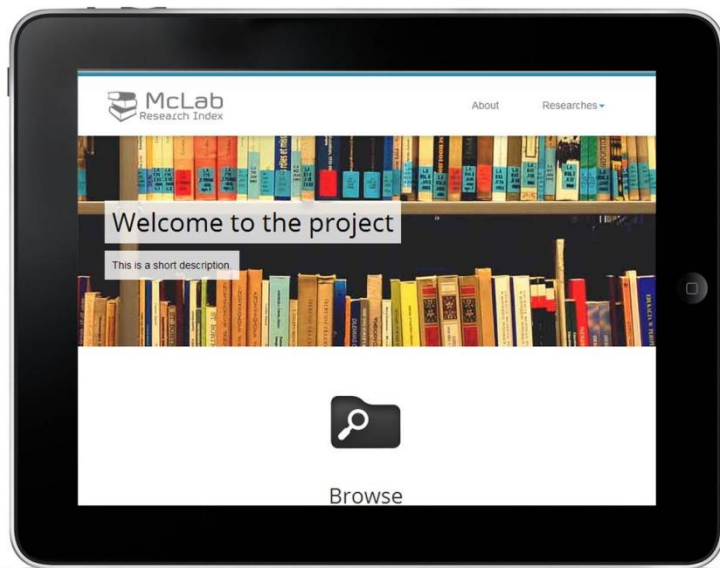
```
role="button">View details &raquo;</a></p>
</div><!-- /.col-lg-4 -->
</div><!-- /.row -->
</div><!-- /.container -->
```



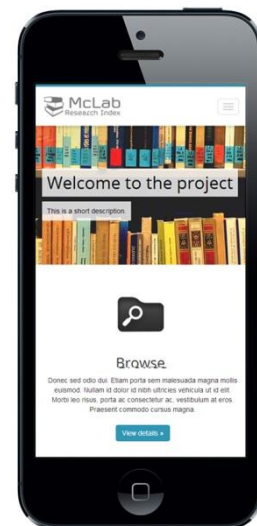
Εικόνα 39 Εμφάνιση αρχικής σελίδας



Εικόνα 40 Εμφάνιση αρχικής σελίδας σε μεγάλη οθόνη



Εικόνα 41 Εμφάνιση σε συσκευή tablet



Εικόνα 42 Εμφάνιση σε συσκευή κινητού

## About

Σε αυτή τη σελίδα ο επισκέπτης μπορεί να ενημερωθεί σχετικά με τις πληροφορίες που μπορεί να πάρει από την ιστοσελίδα, ενώ υπάρχει δυνατότητα να εμπλουτιστεί και με άλλο σχετικό περιεχόμενο που αφορά την εφαρμογή.

```
<div class="pageheader">
  <div class="row">
    <div class="col-sm-12">
      <h1 class="title"><?php echo __d('cake', 'About
Us')?></h1>
    </div>
  </div>
</div>
<div class="pagecontents">
  <div class="row">
    <div class="col-sm-12">
      <p><?php echo __d('cake', '.....')?></p>
    </div>
  </div>
</div>
```



Εικόνα 43 Εμφάνιση σελίδας about

## Αναζήτηση άρθρων

Αυτή είναι ίσως η πιο βασική σελίδα της εφαρμογής καθώς εδώ ο χρήστης μπορεί να δει όλα τα άρθρα που είναι αναρτημένα στην ιστοσελίδα αλλά και να φιλτράρει την εμφάνισή τους με βάση τα κριτήρια που θα επιλέξει:

- 1) Με βάση το όνομα του άρθρου που μπορεί να το εισάγει στο πεδίο της αναζήτησης. Σε αυτό το σημείο η εφαρμογή κάνει χρήση ενός προσθέτου που είναι γραμμένο σε JavaScript με τη χρήση του framework jQuery, λέγεται autocomplete και χρησιμοποιείται για την αυτόματη συμπλήρωση

του ονόματος ενός άρθρου που υπάρχει στην βάση δεδομένων χωρίς να χρειαστεί να ολοκληρωθεί η αναζήτηση, ούτε να ανανεωθεί η σελίδα.

- 2) Με βάση την κατηγορία, τον Συντάκτη και την χρονολογία του άρθρου, χρησιμοποιώντας τα αντίστοιχα φίλτρα. Εδώ χρησιμοποιούμε επίσης την τεχνολογία του jQuery για να κάνουμε αναζήτηση στην βάση δεδομένων και μέσω της τεχνολογίας Ajax να εμφανίζουμε τα αποτελέσματα της αναζήτησης στην σελίδα.

Ο κώδικας της συγκεκριμένης σελίδας αποτελείται από τρία βασικά μέρη, με βάση την δομή model-view-controller που εφαρμόζεται στο cakephp :

- 1) Το μοντέλο που αφορά τους κανόνες για τα άρθρα που υπάρχουν.
- 2) Η εμφάνιση που αφορά πως θα εμφανίζονται τα αποτελέσματα στην σελίδα της αναζήτησης. Ο κώδικας εδώ εμφανίζεται συγκεκριμένα σε δύο αρχεία που συνδέονται μεταξύ τους με τη χρήση του παρακάτω κώδικα JavaScript και έτσι το αρχείο **results.ctp** εμφανίζεται μέσα στο **browse.ctp**

```
<script>
jQuery(document).ready(function() {
    function submit_filter (e){
        // Show while loading
        NProgress.start();

        // Post the form using the form's action
        and data
        $.post($(this).attr('action'),
        $(this).serialize())
        // called when post has finished
        .done(function(data) {
            // inject returned html into page
            $('#results').html(data);
            NProgress.done();
        })
        // called on failure
        .fail(function(data) {
            // Flash message already on page
            if ($('#flashMessage').length) {
                // Fade out flash message

                $('#flashMessage').fadeOut('fast', function(){
                    // Remove from page
                    $(this).remove();
                    // Inject another flash
                    message
                    $('#h2').before('<div
                    class="ajax-message" id="flashMessage">An Ajax error
                    occured, please try again or refresh</div>');
                });
            } else {
                // Inject flash message
                $('#h2').before('<div
                class="ajax-message" id="flashMessage">An Ajax error
                occured, please try again or refresh</div>');
            }
        });
    }
});
```

```
        }
        $(".ajax-
message").click(function() {
        $(".ajax-
message").hide();
        });
        return false;
    };
    $('body').on('submit', '#FilterBrowseForm',
submit_filter );
    $("#FilterBrowseForm").submit();
    $("select").change(function() {
        $("#FilterBrowseForm").submit();
        $(".reset").show();
    });
    $("#title").autocomplete({
        source: "titles_dropdown",
        minLength: 2,
        delay: 2,
    });

    // var currSrc = $("nav a").attr("href");
    // $("nav a").attr("href", currSrc + "1" );
    });
</script>
```

3) Ο ελεγκτής που είναι κώδικας που κάνει όλη τη διεργασία της αναζήτησης

Ο κώδικας χωρίζεται σε τρεις συναρτήσεις:

Την συνάρτηση titles\_dropdown() που είναι υπεύθυνη για την αυτόματη συμπλήρωση των τίτλων στο πεδίο της αναζήτησης.

```
public function titles_dropdown() {
    $this->Post->recursive = -1;
    if ($this->request->is('ajax')) {
        $this->autoRender = false;
        $results = $this->Post->find('all',
array('fields' => array('Post.title'), 'conditions' =>
array('Post.title LIKE ' => '%'. $this->request->query['term']
. '%'),
        'group' => array('Post.title'),
    ));
        $titles = Set::extract('../Post/title',
$results);
        echo json_encode($titles);
    }
}
```

Την συνάρτηση browse() που είναι υπεύθυνη για την εμφάνιση των φίλτρων και την διαδικασία της αναζήτησης.

```
public function browse() {
```

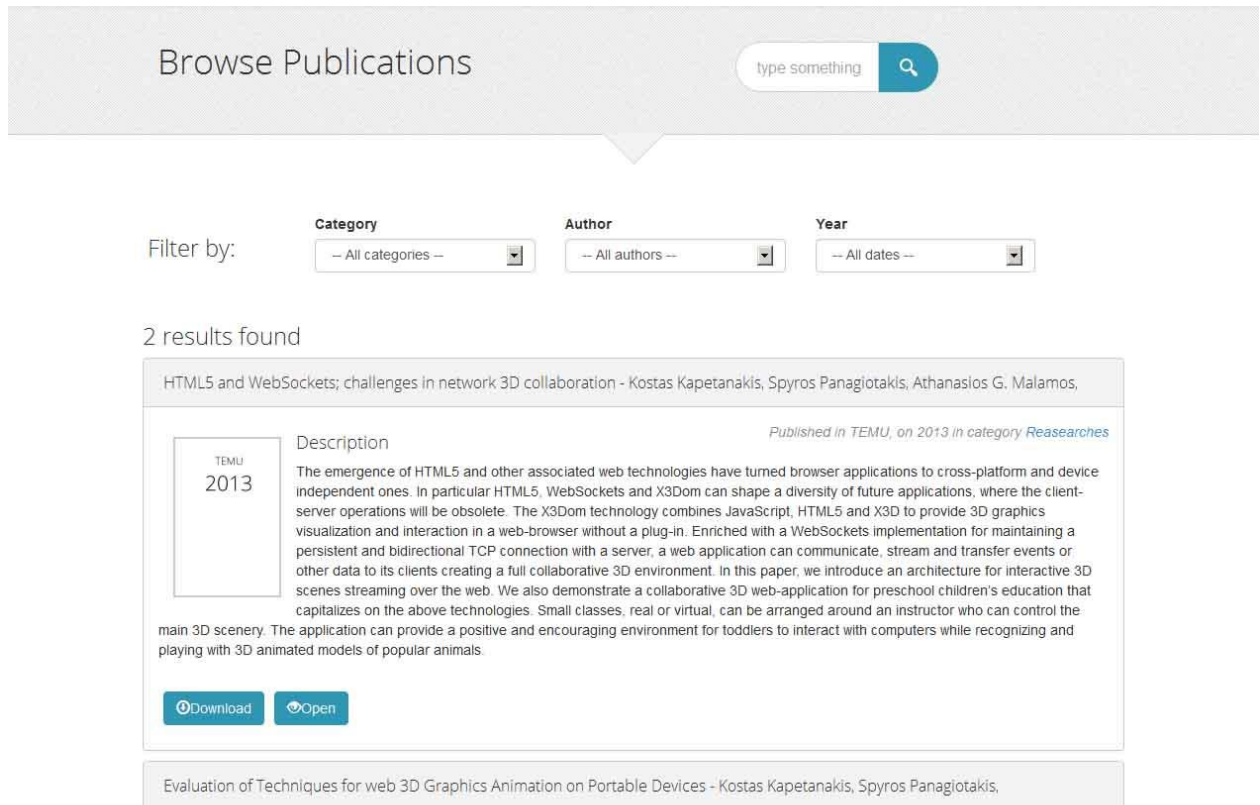
```
// get the possible values for the filters and
// pass them to the view
$categories = $this->Post->Category->find('list');
$authors = $this->Post->Author->find('list',
array('fields' => array('Author.id', 'Author.Name') ));
$dates = $this->Post->find('list', array('fields'
=> array('date_published', 'date_published'), 'group' =>
array('date_published'), 'conditions' =>
array('Post.published' => '1') ) );
$this->set(compact('categories', 'authors',
'dates'));
$this->set('search', isset($this-
>params['named']['search']) ? $this->params['named']['search']
: "");
$titles = '';
$this->set('titles', $titles);
}
```

Την συνάρτηση results() που είναι υπεύθυνη για όλα τα υπόλοιπα και αφορά την εμφάνιση των αποτελεσμάτων και την σελιδοποίησή τους.

```
public function results() {
    $conditions = array();
    //Transform POST into GET
    if(($this->request->is('post') || $this->request-
>is('put')) && isset($this->data['Filter'])){
        $filter_url['controller'] = $this->request-
>params['controller'];
        $filter_url['action'] = $this->request-
>params['action'];
        $filter_url['page'] = 1;
the generated url
        foreach($this->data['Filter'] as $name =>
$value){
            if($value ){
                $filter_url[$name] = $value;
            }
        }
        return $this->redirect($filter_url);
    }
    else {
        // Inspect all the named parameters to apply
the filters
        foreach($this->params['named'] as $name =>
$value){
            if(!in_array($name,
array('page', 'sort', 'direction', 'limit'))){
                if($name == "Author.id"){
                    $conditions['OR'] = array(
                        array('Author_id' => $value )
                    );
                }
                elseif($name == "search"){
                    $conditions['OR'] = array(
```



```
array('Post.title LIKE'
=> '%' . $value . '%'),
array('Post.body LIKE' => '%'
. $value . '%')
);
}
else {
    $conditions[$name] = $value;
}
$this->request-
>data['Filter'][$name] = $value;
}
}
}
$conditions ['AND'] = array('Post.published' =>
'1');
$this->Post->recursive = 1;
$this->paginate = array(
    'limit' => 15,
    'conditions' => $conditions,
    'joins' => array(
        array(
            'alias' =>
'Author',
            'table' =>
'authors_posts',
            'type' =>
'left',
            'conditions'
=> 'Post.id = Author.post_id'
        ),
        array(
            'group' => 'Post.id'
        )
    );
$this->set('posts', $this->paginate());
}
```



Εικόνα 44 Σελίδα αναζήτησης

### Σελίδα προβολής άρθρου

Η σελίδα αυτή χρησιμοποιείται για να προβάλλει ένα συγκεκριμένο άρθρο που επέλεξε ο χρήστης να προβάλλει από τα αποτελέσματα της αναζήτησης. Εδώ ο χρήστης μπορεί να δει τον τίτλο, τα στοιχεία του άρθρου καθώς και το εξώφυλλο του, που παράγεται αυτόματα με την χρήση της php. Επίσης από αυτή τη σελίδα υπάρχει η δυνατότητα ο χρήστης να κατεβάσει στον υπολογιστή το αρχείο pdf του άρθρου ή να το κοινοποιήσει στα social media αν το επιθυμεί.

Ο κώδικας της σελίδας χωρίζεται και πάλι σε τρία μέρη και τα πιο σημαντικά από αυτά είναι:

1) Η εμφάνιση:

```
<div class="pagecontents">
  <div class="row">
    <div class="col-sm-12 post-view single">
      " alt="thumb" />
      <?php if (!empty($post['Author'])): ?>
      <div class="authors">
        <h4>Authors:</h4>
        <?php foreach ($post['Author'] as $author): ?>
        <span class="author">
```

```
                <h2><?php echo $this->Html->link(
$author['name'], array('controller' => 'authors', 'action' =>
'display', $author['id'])); ?>, </h2>
                </span>
                <?php endforeach; ?>
        </div>
        <?php endif; ?>
        <div class="pubcite">
                Published in <?php echo
$post["Post"]["location"];?>, on <?php echo
$post["Post"]["date_published"];
                ?>, in category <?php echo
$this->Html->link($post['Category']['name'], array('controller'
=> 'categories', 'action' => 'display',
$post['Category']['id'])); ?>
        </div>
        <h4>Description:</h4>
        <p>
                <?php echo h($post["Post"]["body"]); ?>
        </p>
        <?php if (!empty($post['Tag'])): ?>
        <div class="tags">
                <h3>Tags:</h3>
                <?php foreach ($post['Tag'] as $tag): ?>
                        <span class="tag">
                                <?php echo $this->Html->link(
$tag['name'], array('controller' => 'tags', 'action' =>
'display', $tag['id'])); ?>
                        </span>
                <?php endforeach; ?>
        </div>
        <?php endif; ?>
        <div class="pubassets pull-left">
                <?php if ($post["Post"]["pdf_file_name"]){
                echo $this->Html->link('<span class="glyphicon
glyphicon-download"></span>'.__d('cake', 'Download'),
'../../'.$this->Upload->uploadUrl($post, 'Post.pdf',
array('urlize' => true )),
                array( "class" => "btn btn-green", 'target'=>
'_blank', 'escape' => false));
                }
                if ($post["Post"]["link"]){
                echo $this->Html->link('<span class="glyphicon glyphicon-
link"></span>'.__d('cake', 'Demo'),
                $post["Post"]["link"],
                array( "class" => "btn btn-green", 'target'=>
'_blank', 'escape' => false));
                }
                if ($this->Session->read('Auth.User')){
                echo $this->Html->link('<span class="glyphicon
glyphicon-pencil"></span>'.__d('cake', 'Edit'),
array("controller" => "posts", "action" => "edit",
$post["Post"]["id"],
array("class" => "btn btn-warning", 'escape' => false)); } ?>
        </div>
        <div id="single share" class=" pull-right">
```

```
<span><?php echo __d('cake', 'Share on');?></span>
<?php $link =
"http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
echo $this->Html->link(__d('cake', 'Facebook'),
'http://www.facebook.com/sharer.php?u='.$link,
array( "class" => "on-facebook", "target" => "_blank", 'escape'
=> false));
echo $this->Html->link(__d('cake', 'Twitter'),
'http://twitter.com/share?text='.urlencode($post["Post"]["title
"]).'&url='.urlencode($link),
array( "class" => "on-twitter", "target" => "_blank"));
echo $this->Html->link(__d('cake', 'Google+'),
'https://plus.google.com/share?url='.urlencode($link),
array( "target" => "_blank", "class" => "on-google"));
echo $this->Html->link(__d('cake', 'Digg'),
'http://digg.com/submit?url='.urlencode($link).'&title='.ur
lencode($post["Post"]["title"]),
array( "target" => "_blank", "class" => "on-digg"));?>
</div>
</div>
</div>
```

## 2) Ο ελεγκτής:

```
public function view($id = null) {
    if (!$this->Post->exists($id)) {
        throw new NotFoundException(__('Invalid post'));
    }
    $options = array('conditions' => array('Post.' . $this->Post-
>primaryKey => $id));
    $this->set('post', $this->Post->find('first', $options));
}
```

## HTML5 and WebSockets; challenges in network 3D collaboration

Authors:

Kostas Kapetanakis, Spyros Panagiotakis,  
Athanasios G. Malamos,

Published in TEMU, on 2013, in category *Researches*

Description:

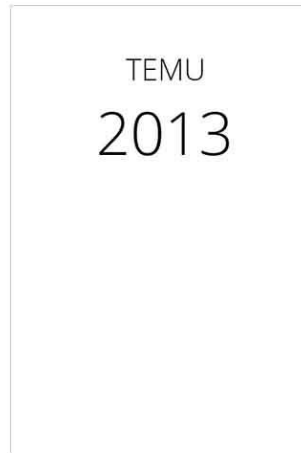
The emergence of HTML5 and other associated web technologies have turned browser applications to cross-platform and device independent ones. In particular HTML5, WebSockets and X3DOM can shape a diversity of future applications, where the client-server operations will be obsolete. The X3DOM technology combines JavaScript, HTML5 and X3D to provide 3D graphics visualization and interaction in a web-browser without a plug-in. Enriched with a WebSockets implementation for maintaining a persistent and bidirectional TCP connection with a server, a web application can communicate, stream and transfer events or other data to its clients creating a full collaborative 3D environment. In this paper, we introduce an architecture for interactive 3D scenes streaming over the web. We also demonstrate a collaborative 3D web-application for preschool children's education that capitalizes on the above technologies. Small classes, real or virtual, can be arranged around an instructor who can control the main 3D scenery. The application can provide a positive and encouraging environment for toddlers to interact with computers while recognizing and playing with 3D animated models of popular animals.

Tags:

Virtual Worlds 3D websockets HTML5 streaming interactive collaborative X3D  
X3DOM

Download Edit

Share on    



Εικόνα 45 Εμφάνιση σελίδας άρθρου

### Σελίδα Συντάκτη

Σε αυτή τη σελίδα παρουσιάζεται το όνομα του συντάκτη, τα στοιχεία επικοινωνίας, καθώς και όλα τα άρθρα τα οποία είναι συνδεδεμένα με αυτόν. Ο κώδικας που αξίζει να παρουσιαστεί για αυτή την σελίδα είναι μόνο το σημείο που αυτή διαφέρει, αφού στα υπόλοιπα σημεία γίνεται επαναχρησιμοποίηση του κώδικα που γράψαμε για την εμφάνιση των αποτελεσμάτων.

```
<div class="author-profile pageheader">
  <div class="headercontent">
    <div class="section-container">
      <div class="row">
        <div class="col-sm-6">
          <h1 class="title">
            <?php echo h($author['Author']['name']); ?>
          </h1>
          <h4>
            <?php echo h($author['Author']['title']); ?>
          </h4>
        </div>
        <div class="col-sm-6">
          <div class="info">
            <h4>Contact info: </h4>
            <p>
              <?php echo h($author['Author']['contact']); ?>
            </p>
            <?php if( $author['Author']['linkedin'] ){ ?>
            <h4>
              find on LinkedIn
            </h4>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
        </h4>
<script src="//platform.linkedin.com/in.js"
type="text/javascript"></script>
<script type="IN/MemberProfile" data-id="<?php echo
h($author['Author']['linkedin']); ?>" data-format="hover"></script>
        <h4> || </h4>
        <?php }if( $author['Author']['email'] ){ ?>
        <h4>
        <a href="mailto:<?php echo
h($author['Author']['email']); ?>">Send email</a>
        </h4>
        <?php }?>
        </div>
    </div>
</div>
</div>
```

<b>Kostas Kapetanakis</b> R & D at Dept. of Applied Informatics and Multimedia, TEI of Crete	Contact info: Department of Applied Informatics and Multimedia Technological Educational Institute of Crete Iraklio, Crete, Greece, GR 71004 <a href="#">find on LinkedIn</a>    <a href="#">Send email</a>
---	---

HTML5 and WebSockets; challenges in network 3D collaboration

TEMu  
2013

Description

Published on 2013 in category *Researches*

The emergence of HTML5 and other associated web technologies have turned browser applications to cross-platform and device independent ones. In particular HTML5, WebSockets and X3Dom can shape a diversity of future applications, where the client-server operations will be obsolete. The X3Dom technology combines JavaScript, HTML5 and X3D to provide 3D graphics visualization and interaction in a web-browser without a plug-in. Enriched with a WebSockets implementation for maintaining a persistent and bidirectional TCP connection with a server, a web application can communicate, stream and transfer events or other data to its clients creating a full collaborative 3D environment. In this paper, we introduce an architecture for interactive 3D scenes streaming over the web. We also demonstrate a collaborative 3D web-application for preschool children's education that capitalizes on the above technologies. Small classes, real or virtual, can be arranged around an instructor who can control the main 3D scenery. The application can provide a positive and encouraging environment for toddlers to interact with computers while recognizing and playing with 3D animated models of popular animals.

[Download](#) [Open](#)

Evaluation of Techniques for web 3D Graphics Animation on Portable Devices

Εικόνα 46 Εμφάνιση σελίδας συντάκτη

### Σελίδα κατηγορίας / λέξης κλειδί

Παρομοίως με την σελίδα συντάκτη, έτσι και στις σελίδες κατηγορίας και λέξης κλειδιού, εμφανίζονται τα άρθρα που τους αντιστοιχούν και γίνεται επαναχρησιμοποίηση του ίδιου κώδικα για την προβολή αποτελεσμάτων.

## Posts in category: Reasearches

HTML5 and WebSockets; challenges in network 3D collaboration - Kostas Kapetanakis, Spyros Panagiotakis, Athanasios G. Malamos, Published on 2013

TEMU  
2013

**Description**

The emergence of HTML5 and other associated web technologies have turned browser applications to cross-platform and device independent ones. In particular HTML5, WebSockets and X3Dom can shape a diversity of future applications, where the client-server operations will be obsolete. The X3Dom technology combines JavaScript, HTML5 and X3D to provide 3D graphics visualization and interaction in a web-browser without a plug-in. Enriched with a WebSockets implementation for maintaining a persistent and bidirectional TCP connection with a server, a web application can communicate, stream and transfer events or other data to its clients creating a full collaborative 3D environment. In this paper, we introduce an architecture for interactive 3D scenes streaming over the web. We also demonstrate a collaborative 3D web-application for preschool children's education that capitalizes on the above technologies. Small classes, real or virtual, can be arranged around an instructor who can control the main 3D scenery. The application can provide a positive and encouraging environment for toddlers to interact with computers while recognizing and playing with 3D animated models of popular animals.

[Download](#) [Open](#)

Evaluation of Techniques for web 3D Graphics Animation on Portable Devices - Kostas Kapetanakis, Spyros Panagiotakis,

Εικόνα 47 Εμφάνιση σελίδας κατηγορίας

### Σελίδα προβολής όλων των λέξεων κλειδιών (tag cloud).

Σε αυτή τη σελίδα προβάλλονται όλες οι λέξεις κλειδιά που έχουν αντιστοιχιστεί σε όλα τα άρθρα που φιλοξενεί η εφαρμογή. Έτσι ο χρήστης μπορεί να επιλέξει μια λέξη και να δει τα άρθρα που σχετίζονται με αυτήν.

Εδώ ο κώδικας είναι πάλι σε δυο μέρη:

#### 1) Αυτό της εμφάνισης:

```
<div class="pagecontents">
  <div class="row">
    <div class="col-md-12">
      <ul class="tag-cloud" >
        <?php foreach ($tags as $tag): ?>
          <li class="tag">
            <?php echo $this->Html->link(
              h($tag['Tag']['name']), array('action' => 'display',
              $tag['Tag']['id'])); ?>
          </li>
        <?php endforeach; ?>
      </ul>
    </div>
  </div>
</div>
```

#### 2) Και αυτό του ελεγκτή:

```
public function browse() {
```

```
$this->Tag->recursive = 0;  
$this->paginate = array( 'limit' =>150);  
$this->set('tags', $this->Paginator->paginate());  
}
```

## 6.2.7 Δημιουργία περιοχής διαχείρισης

Αφού έχουμε έτοιμη την περιοχή που θα έχουν πρόσβαση όλοι οι επισκέπτες της ιστοσελίδας, πρέπει να ορίσουμε και μια περιοχή διαχείρισης στην οποία θα έχουν πρόσβαση μόνο εξουσιοδοτημένοι χρήστες που θα έχουν την δυνατότητα να τροποποιούν το εμφανιζόμενο περιεχόμενο. Πιο συγκεκριμένα ο διαχειριστής θα μπορεί να δημιουργήσει με ευκολία καινούρια άρθρα και να τροποποιήσει τα ήδη υπάρχοντα. Επίσης θα διαχειρίζεται τους συντάκτες, τις κατηγορίες και τις λέξεις κλειδιά που σχετίζονται με τα άρθρα, όπως επίσης και τους χρήστες που θα έχουν πρόσβαση στην περιοχή διαχείρισης.

Για να δημιουργήσουμε μια περιοχή που θα έχει περιορισμένη πρόσβαση χρησιμοποιούμε την ενσωματωμένη λειτουργία πιστοποίησης χρηστών που υπάρχει στο Cakephp και προσθέτουμε τον εξής κώδικα στον ελεγκτή της οντότητας των χρηστών.

```
public function beforeFilter() {  
    parent::beforeFilter();  
    $this->Auth->deny();  
}  
  
public function login() {  
    if ($this->request->is('post')) {  
        if ($this->Auth->login()) {  
            return $this->redirect($this->Auth->redirect());  
        }  
        $this->Session->setFlash(__('Invalid username or  
password, try again'));  
    }  
}  
  
public function logout() {  
    return $this->redirect($this->Auth->logout());  
}
```

Το επόμενο βήμα που πρέπει να γίνει είναι να φτιαχτεί μια καταχώρηση στην βάση δεδομένων ώστε να μπορεί η εφαρμογή μας να συγκρίνει τα στοιχεία ( username και password) με τα στοιχεία που θα δίνει ο εκάστοτε διαχειριστής ώστε να του δίνεται πρόσβαση.

Έτσι τροποποιούμε τον κώδικα που παράχθηκε αυτόματα από το Cakephp για την οντότητα του χρήστη και συγκεκριμένα τη φόρμα για την εισαγωγή του:

Ελεγκτής:

```
public function add() {  
    if ($this->request->is('post')) {  
        $this->User->create();  
        if ($this->User->save($this->request->data)) {  
            $this->Session->setFlash(__('The user has been  

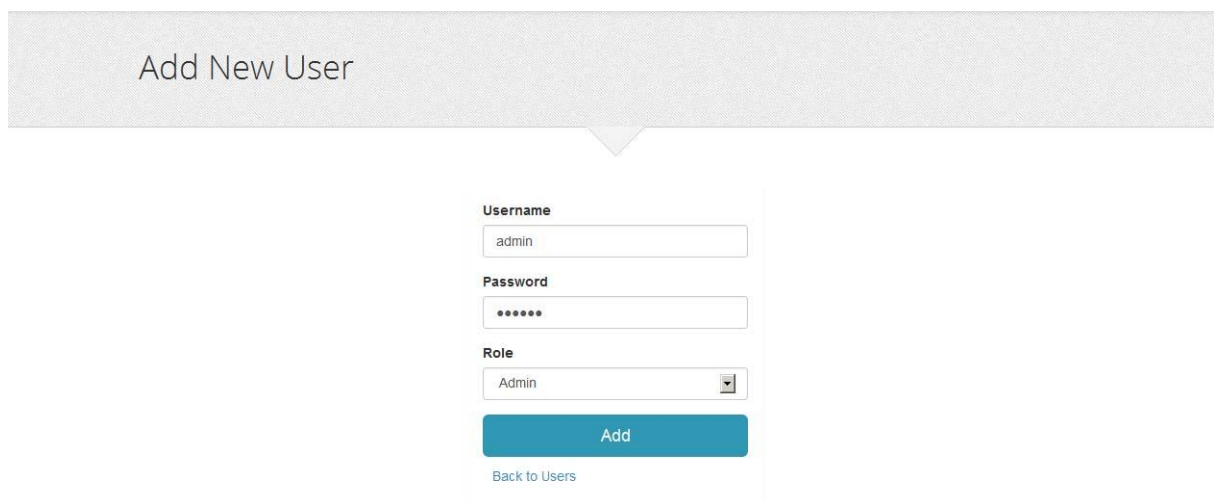
```



```
saved'));  
        return $this->redirect(array('action' => 'index'));  
    }  
    $this->Session->setFlash(__('The user could not be  
saved. Please, try again.));  
    }  
}
```

Εμφάνιση:

```
<?php echo $this->Form->create('User'); ?>  
<fieldset>  
<?php echo $this->Form->input('id'); ?>  
    <div class="form-group">  
        <?php echo $this->Form->input('username' , array('class'  
=> 'form-control', 'placeholder' => 'username') ); ?>  
    </div>  
    <div class="form-group">  
        <?php echo $this->Form->input('password' , array('class'  
=> 'form-control', 'placeholder' => 'username') ); ?>  
    </div>  
    <div class="form-group">  
        <?php echo $this->Form->input('role', array( 'class' =>  
'form-control', 'options' => array('admin' => 'Admin', 'author' =>  
'Author')) );?>  
    </div>  
</fieldset>  
<?php echo $this->Form->submit('Add',  
array('class' => 'btn btn-lg btn-green btn-block'));  
echo $this->Form->end();?>
```



Add New User

Username  
admin

Password  
\*\*\*\*\*

Role  
Admin

Add

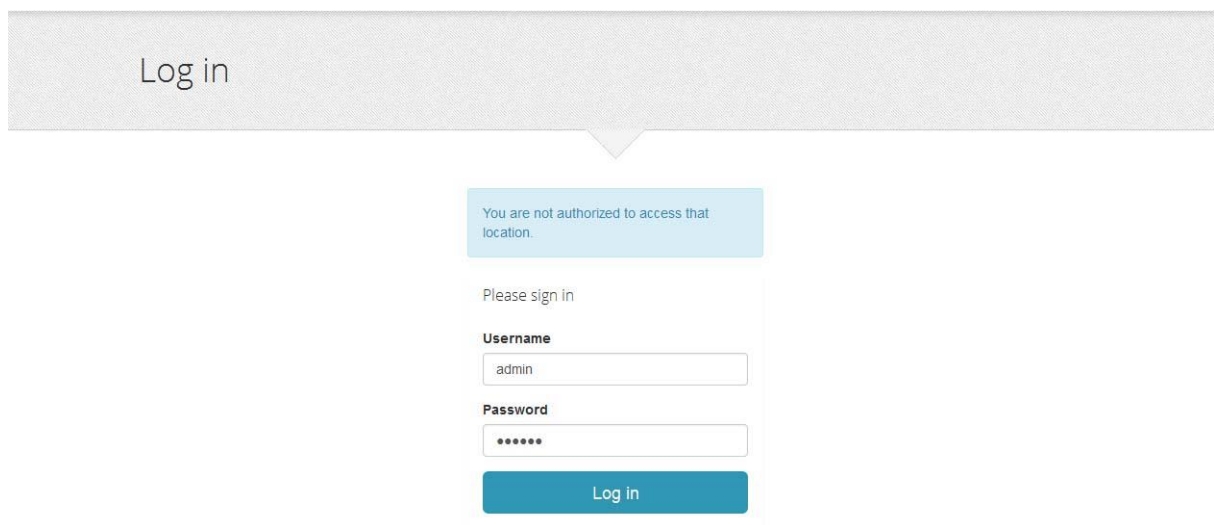
[Back to Users](#)

Εικόνα 48 Φόρμα εισαγωγής χρήστη

Στη συνέχεια δημιουργούμε μια φόρμα login για να μπορεί να εισέλθει ο διαχειριστής.

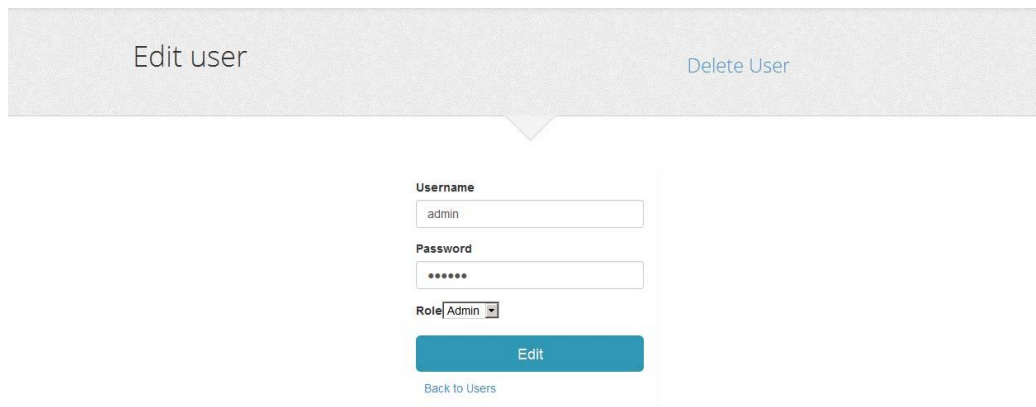
```
<?php echo $this->Form->create('User'); ?>
```

```
<fieldset>
  <div class="form-group">
    <?php echo $this->Form->input('username',
array('class' => 'form-control', 'placeholder' => 'username') ); ?>
  </div>
  <div class="form-group">
    <?php echo $this->Form->input('password',
array('class' => 'form-control', 'placeholder' => 'password')); ?>
  </div>
</fieldset>
<?php
echo $this->Form->submit('Log in',
array('class' => 'btn btn-lg btn-green btn-block'));
echo $this->Form->end();
?>
```



Εικόνα 49 Φόρμα εισόδου διαχειριστή

Εφόσον δημιουργήσουμε τον πρώτο χρήστη και τη φόρμα εισόδου μπορούμε πλέον να κάνουμε login με τα στοιχεία του και να αρχίζουμε να περιορίζουμε την πρόσβαση σε συγκεκριμένες περιοχές. Αυτό είναι πολύ βασικό αφού δεν θέλουμε πχ ο οπουδήποτε να μπορεί να δημιουργήσει ένα άρθρο. Στη συνέχεια τροποποιούμε τον κώδικα για τις σελίδες της επεξεργασίας και προβολής των υπαρχόντων χρηστών.



Username  
admin

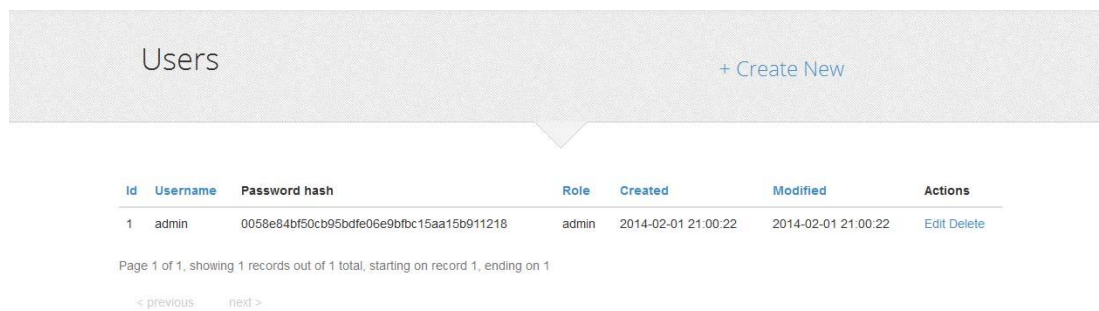
Password  
\*\*\*\*\*

Role Admin

Edit

Back to Users

Εικόνα 50 Επεξεργασία χρήστη



Id	Username	Password hash	Role	Created	Modified	Actions
1	admin	0058e84bf50cb95bdfc06e9bfbcb15aa15b911218	admin	2014-02-01 21:00:22	2014-02-01 21:00:22	Edit Delete

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

< previous next >

Εικόνα 51 Λίστα χρηστών

Στην συνέχεια τροποποιούμε τα αρχεία που έχουν να κάνουν με την προσθήκη και την επεξεργασία των κατηγοριών, των συντακτών, των λέξεων κλειδιών και της αντιστοίχισής τους με τα άρθρα.

Το αποτέλεσμα που προκύπτει για τους συντάκτες είναι αυτό και παρομοίως ισχύει και για τις κατηγορίες και τις λέξεις κλειδιά, όπου γίνεται η επαναχρησιμοποίηση του ίδιου κώδικα με μικρές αλλαγές:

## Add New Author

**Name**

**Title**

**Contact**

**Email**

**Linkedin**

**Post**

[Back to Authors](#)

Εικόνα 52 Εισαγωγή συντάκτη

Edit author Delete Author

**Name**

**Title**

**Contact**

**Email**

**Linkedin**

**Post**

[Back to Authors](#) Edit

Εικόνα 53 Επεξεργασία συντάκτη

Authors + Create New

Id	Name	Created	Modified	Actions
8	Kostas Kapetanakis	2014-01-27 16:23:37	2014-02-01 17:01:08	Edit Delete
16	Spyros Panagiotakis	2014-02-01 20:38:06	2014-02-01 20:44:52	Edit Delete
17	Athanasios G. Malamos	2014-02-01 20:48:07	2014-02-01 20:50:09	Edit Delete

Page 1 of 1, showing 3 records out of 3 total, starting on record 1, ending on 3

< previous    next >

Εικόνα 54 Λίστα συντακτών

Αφού τελειοποιήσουμε τις οντότητες που σχετίζονται με τα άρθρα θα προχωρήσουμε στην δημιουργία, την επεξεργασία και τη διαγραφή των ίδιων των άρθρων. Για να διευκολύνουμε και να επιταχύνουμε την διαδικασία της εισαγωγής των άρθρων θα πρέπει να προσθέσουμε μια σειρά από απλοποιήσεις που θα την αυτοματοποιούν. Μια βασική λειτουργία που προστέθηκε γι αυτό το λόγο είναι η αυτόματη επιλογή και εισαγωγή συντακτών με βάση τα meta data του pdf αρχείου.

Ξεκινάμε με την προσθήκη ενός άρθρου και τροποποιούμε κατά πολύ τον κώδικα που παράχθηκε αυτόματα, αφού στην παρούσα κατάσταση αυτό που χρειαζόμαστε είναι το ανέβασμα του pdf αρχείου, η αποθήκευση του στη βάση δεδομένων, η εξαγωγή των metadata και η αποθήκευσή τους στα στοιχεία του άρθρου και τέλος η ανακατεύθυνση στην σελίδα επεξεργασίας των στοιχείων ώστε να συμπληρωθούν και τα υπόλοιπα στοιχεία.

Ο κώδικας της εμφάνισης:

```
<div class="col-md-6 col-md-offset-3">
  <div class="post-form">
    <h2> Select a Pdf file </h2>
    <?php echo $this->Form->create('Post', array('type'=>'file'));
    echo $this->Form->hidden('id');?>
    <?php echo $this->Form->input('pdf' , array('class' =>
'file_upload', 'type' => 'file', 'id' => 'fileselect', 'label'=>'')
); ?>
    <?php echo $this->Form->submit('Add',
    array('class' => 'btn btn-lg btn-green pull-right'));
    echo $this->Form->end();?>
    <div id="messages">
      <p>
      </p>
    </div>
  </div>
</div>
```

Ο ελεγκτής:

```
public function add() {
    function pdfparser($document) {
        include APPLIBS.'pdfparser/vendor/autoload.php';
        $parser = new \Smalot\PdfParser\Parser();
        $pdf = $parser->parseFile($document);
        $details = $pdf->getDetails();
        if($details['Title'])
            $title = $details['Title'];
        else $title = "";
        if($details['Author'])
            $authors=explode(',', $details['Author']);
        else $authors = "";
        return array('title' => $title, 'authors' => $authors);}
    if ($this->request->is('post')) {
        $this->Post->create();
        if($filename = $this->request->data['Post']['pdf']['tmp_name']){
            $pdf_data = pdfparser($filename);
            if($pdf_data['title']){
                $this->Post->set('title', $pdf_data['title'] );}
            if($pdf_data['authors']){
                $author_names = $this->Post->Author->find('list',
array('fields' => array('Author.name')));
                $i=0;
                foreach ($pdf_data['authors'] as $new_author_name){
                    $new_author_name = preg_replace('!\s+', ' ',
                    $new_author_name);
```

```
$new_author_name = trim($new_author_name);
    $found=0;
    foreach ($author_names as $author_name){
        if( $author_name== $new_author_name){
            $author_id = $this->Post->Author-
>find('first', array('fields' => array('Author.id'), 'conditions'
=> array('Author.name' => $author_name )));
            $author_ids[$i] = $author_id['Author']['id'];
            $i++;
            $found=1;
        }
        if(!$found){
            $this->Post->Author->create();
            $this->Post->Author->set('name', $new_author_name );
            $this->Post->Author->save($this->request->data);
            $author_ids[$i]= $this->Post->Author->id;
            $i++;}}
        $i=0;
        foreach ($author_ids as $author_id){
            $this->request->data['Author']['Author'][$i] =
$author_id;
            $i++;
        }
    }
    if ($this->Post->save($this->request->data)) {
        $id=$this->Post->id;
        $this->Session->setFlash(__('Edit the information'));
        return $this->redirect(array('action' => 'edit', $this-
>Post->id));
    } else {
        $this->Session->setFlash(__('The post could not be saved.
Please, try again.'));
    }
}
```

Add New Post

Select a Pdf file



Εικόνα 55 Προσθήκη νέου άρθρου

Αφού προσθέσουμε το αρχείο του άρθρου με επιτυχία και εξαχθούν οι πληροφορίες που αυτό περιέχει, ανακατευθυνόμαστε στην σελίδα επεξεργασίας. Σε αυτή τη σελίδα έχουμε πρόσβαση και σε οποιαδήποτε άλλη στιγμή αν πατήσουμε την επιλογή edit σε ένα άρθρο στο περιβάλλον διαχείρισης. Εδώ αποθηκεύονται οι πρόσθετες πληροφορίες που παρέχουμε για το άρθρο και πριν την αποθήκευση δημιουργείται με την χρήση της php η εικόνα του εξωφύλλου και η μικρογραφία του, με βάση το που και πότε δημοσιεύτηκε το κάθε άρθρο.

Η εμφάνιση:

```
< fieldset >
  < div class = "form-group" >
    <? php echo $this - > Form - > input('category_id',
array('class' = > 'form-control')); ?>
  < /div>
  <div class="form-group">
    <?php echo $this->Form->input('title' , array('class' => 'form-
control', 'placeholder' => 'title') ); ?>
  </div >
  < div class = "form-group" >
    <?php echo $this - > Form - > input('location', array('class' =
> 'form-control', 'label' = > 'Published in', 'placeholder' = >
'Conference or jurnal')); ?>
  < /div>
  <div class="form-group">
    <?php echo $this->Form->input('body' , array('class' => 'form-
control', 'placeholder' => 'description') ); ?>
  </div >
  < div class = "form-group" >
    <? php echo $this - > Form - > input('link', array('class' = >
'form-control', 'rows' = > '2', 'placeholder' = > 'link')); ?>
  < /div>
  <div class="form-group">
    <?php echo $this->Form->input('pdf' , array('class' =>
'file_upload', 'type' => 'file' ) ); ?>
  </div >
  < div class = "form-group date-published" >
    <? php echo $this - > Form - > input('date_published',
array('class' = > 'form-control', 'placeholder' = > 'year', 'rows' =
> '1', 'label' = > 'Year published')); ?>
  < /div>
  <div class="form-group">
    <?php echo $this->Form->input('published' ); ?>
  </div >
  < div class = "form-group" >
    <? php echo $this - > Form - > input('Author', array('class'
= > 'form-control', )); ?>
  < /div>
  <div class="form-group">
    <?php echo $this->Form->input('Tag' , array('class' => 'form-
control') ); ?>
  </div >
< /fieldset>
```



Ο ελεγκτής:

```
public function edit($id = null) {
    if (!$this->Post->exists($id)) {
        throw new NotFoundException(__('Invalid post'));
    }
    if ($this->request->is(array('post', 'put'))) {
        if ($this->Post->save($this->request->data)) {
            $this->Session->setFlash(__('post saved.'));
            return $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The post could not be
saved. Please, try again.'));
        }
    } else {
        $options = array('conditions' => array('Post.' . $this->
Post->primaryKey => $id));
        $this->request->data = $this->Post->find('first',
$options);
    }
    $categories = $this->Post->Category->find('list');
    $authors = $this->Post->Author->find('list', array('fields' =>
array('Author.id', 'Author.Name' ) ));
    $tags = $this->Post->Tag->find('list');
    $this->set(compact('categories', 'authors', 'tags'));
}
```

Το μοντέλο:

```
public function afterSave($created, $options = array()){
    //create cover
    $id = $this->data['Post']['id'];
    $title = $this->data['Post']['location'];
    $year = $this->data['Post']['date_published'];
    $this->designcover($id, $title, $year);
    return true;
}

public function beforeDelete($cascade = true) {
    $id = $this->id;
    $file = new File( WWW_ROOT
.'/img/upload/images/' . $id . '_default.png' );
    $file2 = new File( WWW_ROOT
.'/img/upload/images/' . $id . '_thumb.png' );
    $file->delete();
    $file2->delete();
}

public function designcover( $id, $title, $year ){
    global $nextY;
    $nextY = 100;
    $font_file = WWW_ROOT .'/img/upload/cover/OpenSans-
Light_0.ttf';
    $image_file = WWW_ROOT .'/img/upload/cover/image sample.png';
}
```

```
$image = imagecreatefrompng($image_file);
$color = ImageColorAllocate($image,0,0,0) ;
$image= $this->imageToText($image,30 /*font*/ , 100 /*y*/, 0
/*min x*/, 190 /*max x*/, $title , $color , $font_file );
$nextY =$nextY +40;
$image= $this->imageToText($image,60 , $nextY, 0, 150, $year,
$color, $font_file );
imagepng($image, WWW_ROOT
.'/img/upload/images/' . $id . '_default.png');
$this->resize(WWW_ROOT
.'/img/upload/images/' . $id . '_default.png', WWW_ROOT
.'/img/upload/images/' . $id . '_thumb.png' );
}
function resize($originalimg, $targetimg){
list($width, $height) = getimagesize($originalimg);
$new_width = 105;
$new_height = 158;
$image_p = imagecreatetruecolor($new_width, $new_height);
$image = imagecreatefrompng($originalimg);
imagecopyresampled($image_p, $image, 0, 0, 0, 0, $new_width,
$new_height, $width, $height);
imagepng ( $image_p, $targetimg );
}
public function imageToText($img,$font,$nextY,$x1,$x2,$text,$color,
$font_file ){
global $nextY;
$fontwidth = imagefontwidth($font);
$fullwidth = strlen($text) * $fontwidth;
$maxwidth = $x2-$x1;
$targetwidth = $fullwidth-(4*$fontwidth);
if($fullwidth > $maxwidth) {
for($i = 0; $i < strlen($text) AND ((strlen($text)-($i-
4))*$fontwidth) > $targetwidth ;$i++) { }
}
$spacing=2;
if(strlen($text) > 15 ){
$new_text = wordwrap($text, 15, '\n');
$lines=explode('\n',$new_text);
for($i=0; $i< count($lines); $i++){
$newY=$nextY+($i * $font * $spacing);
imaggotfttext($img, $font, 0 , $x2 -
(strlen($lines[$i]) * $fontwidth), $newY, $color, $font_file,
$lines[$i]);}
$nextY=$nextY+($i * $font * $spacing);
}
else{
imaggotfttext ( $img , $font , 0, $x2 - (strlen($text) *
$fontwidth), $nextY, $color , $font_file , $text );
$nextY=$nextY+($font * $spacing);}
return $img;
exit;
}
```

Edit Post Delete Post

---

**Category**  
Researches

**Title**  
HTML5 and WebSockets; challenges in network 3D collaboration

**Published in**  
TEMU

**Body**  
The emergence of HTML5 and other associated web technologies have turned browser applications to cross-platform and device independent ones. In particular HTML5, WebSockets and X3Dom can shape a diversity of future applications, where the client-server operations will be obsolete. The X3Dom technology combines JavaScript, HTML5 and X3D to provide 3D graphics visualization and interaction in a web-browser without a plug-in. Enriched with a

**Link**  
link

**Pdf**  
Αναζήτηση... Δεν επιλέχθηκε αρχείο.

**Year published**  
2013

Published

**Author**  
Kostas Kapetanakis  
Spyros Panagiotakis  
Athanasios G. Malamos

**Tag**  
3D  
websockets  
HTML5  
ajax  
wp

[Back to Posts](#) Edit

Εικόνα 56 Σελίδα επεξεργασίας άρθρου.

Τέλος δημιουργούμε την σελίδα που εμφανίζει σε μια λίστα όλα τα άρθρα με δυνατότητα να αναδιάταξης με βάση σχεδόν όλα τα indexable πεδία. Ο κώδικας βασίζεται στην επαναχρησιμοποίηση των προηγούμενων. Επίσης σε κάθε άρθρο ο διαχειριστής έχει την δυνατότητα να επεξεργαστεί ή να διαγράψει ένα άρθρο.

Posts		+ Create New				
<b>Id</b>	<b>Title</b>	<b>Date Published</b>	<b>Created</b>	<b>Modified</b>	<b>Published</b>	<b>Actions</b>
2	<a href="#">HTML5 and WebSockets; challenges in network 3D collaboration</a>	2013	2014-02-01 20:48:08	2014-02-01 20:52:19	1	Edit Delete
3	<a href="#">Evaluation of Techniques for web 3D Graphics Animation on Portable Devices</a>	2012	2014-02-07 14:48:36	2014-02-07 15:11:34	1	Edit Delete

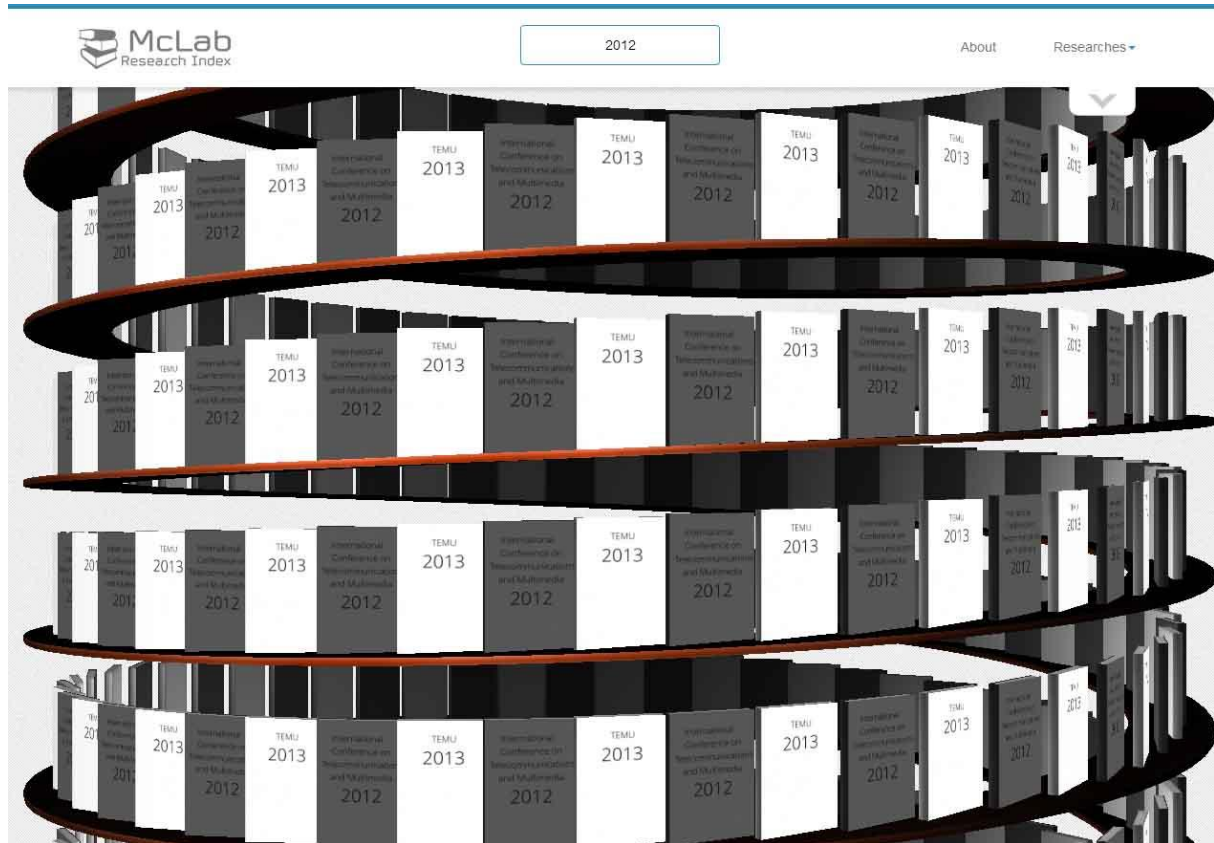
Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

< previous    next >

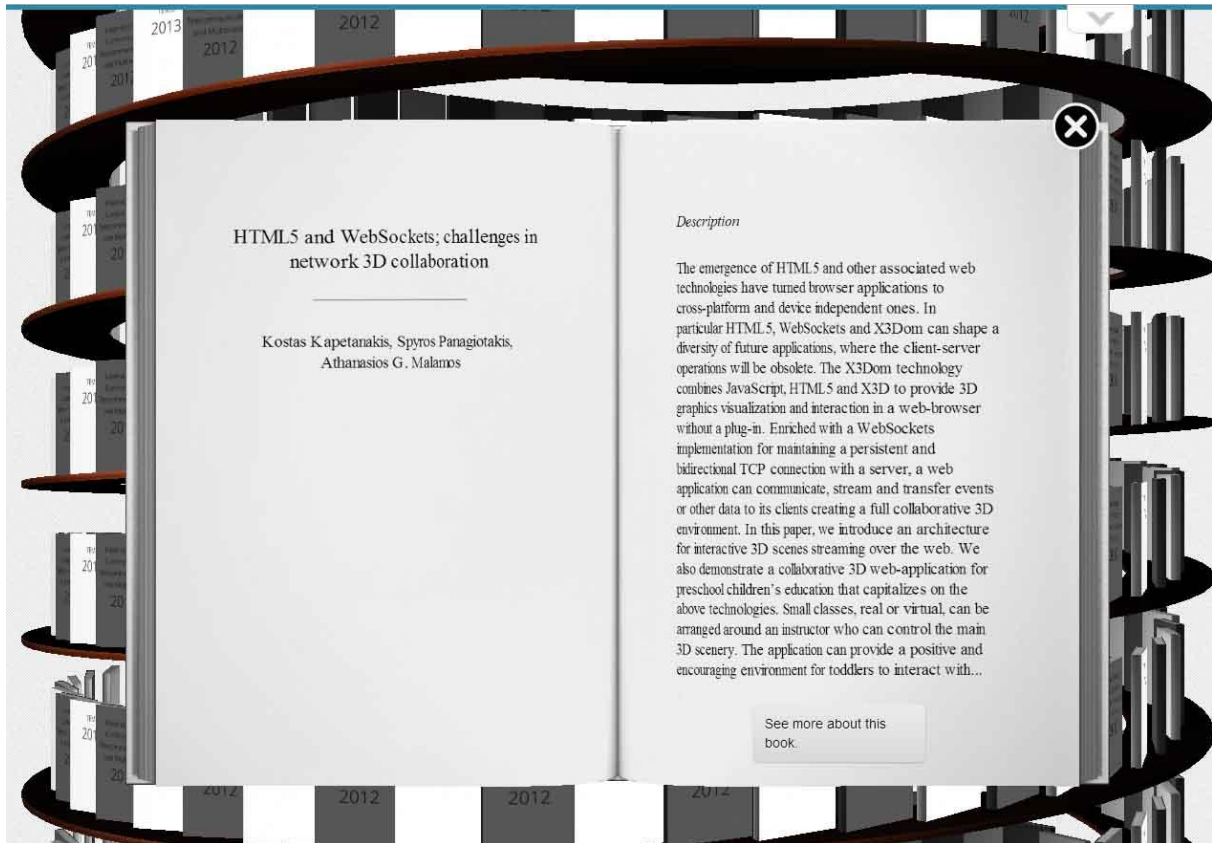
Εικόνα 57 Σελίδα εμφάνισης λίστας άρθρων

## 6.2.8 Δημιουργία τρισδιάστατης βιβλιοθήκης

Αυτή είναι η πιο εντυπωσιακή σελίδα της εφαρμογής καθώς σε αυτήν παρουσιάζονται όλα τα άρθρα σε μια τρισδιάστατη βιβλιοθήκη με την χρήση των τεχνολογιών JavaScript και WebGL. Εδώ ο χρήστης έχει την δυνατότητα να γυρίσει την βιβλιοθήκη με την χρήση του κέρσορα, σαν αυτή να ήταν μια πραγματική κυκλική βιβλιοθήκη και να επιλέξει ένα από τα βιβλία που αυτή περιέχει. Επιλέγοντας ένα βιβλίο βλέπει το εξώφυλλό του να μεγαλώνει με τη χρήση animation και πατώντας πάλι πάνω σε αυτό μπορεί να δει παραπάνω στοιχεία καθώς και ένα σύνδεσμο στην σελίδα προβολής του συγκεκριμένου άρθρου.



Εικόνα 58 Τρισδιάστατη βιβλιοθήκη



Εικόνα 59 Τρισδιάστατη βιβλιοθήκη με επιλεγμένο βιβλίο

Όλη η υλοποίηση της βιβλιοθήκης είναι βασισμένη σε ένα έτοιμο λογισμικό που έχει το όνομα WebGL Bookcase και βρίσκεται εδώ <http://www.chromeexperiments.com/detail/webgl-bookcase/>. Η εφαρμογή αυτή είναι φτιαγμένη για να παρουσιάζει τα βιβλία που φιλοξενούνται στο ηλεκτρονικό κατάστημα της Google κάνοντας χρήση του σχετικού API που έχει διαμορφωθεί για την εξαγωγή των στοιχείων των φιλοξενούμενων βιβλίων.

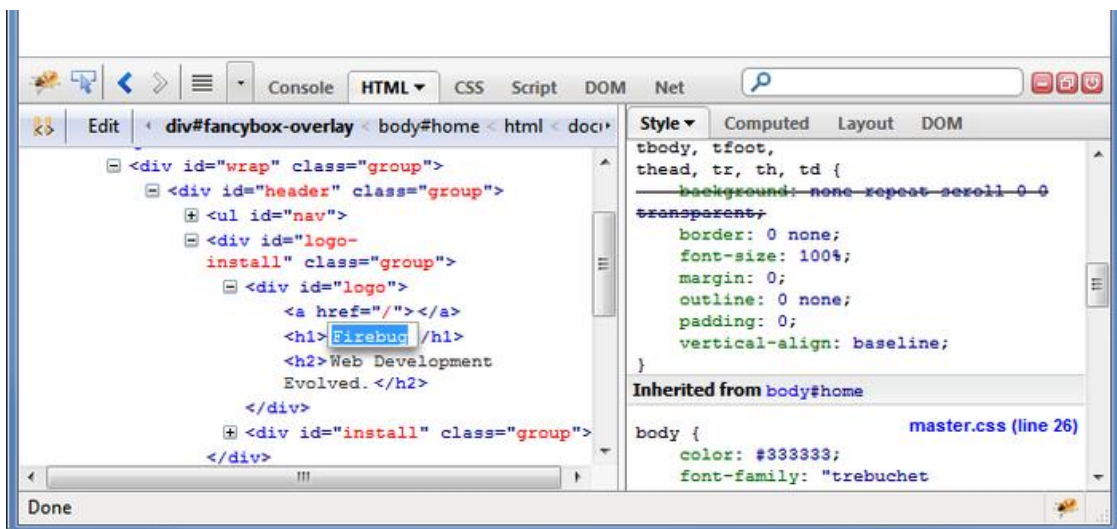
Η διαδικασία τροποποίησης του λογισμικού αυτού μέχρι να έρθει στα μέτρα της εφαρμογής μας έχει ως εξής:

- 1) Χρήση κάποιου εργαλείου για offline προβολή μιας ιστοσελίδας ώστε να αποθηκεύσουμε τοπικά στον υπολογιστή μας όλα τα αρχεία της εφαρμογής. Για αυτή τη δουλειά επιλέξαμε το HTTrack Website Copier που μας δίνει την δυνατότητα να δεχτεί ένα link και να αντιγράψει το περιεχόμενο και τις συνδεδεμένες σελίδες του στον φάκελο που θα επιλέξουμε κρατώντας τη δομή που είχε πριν.



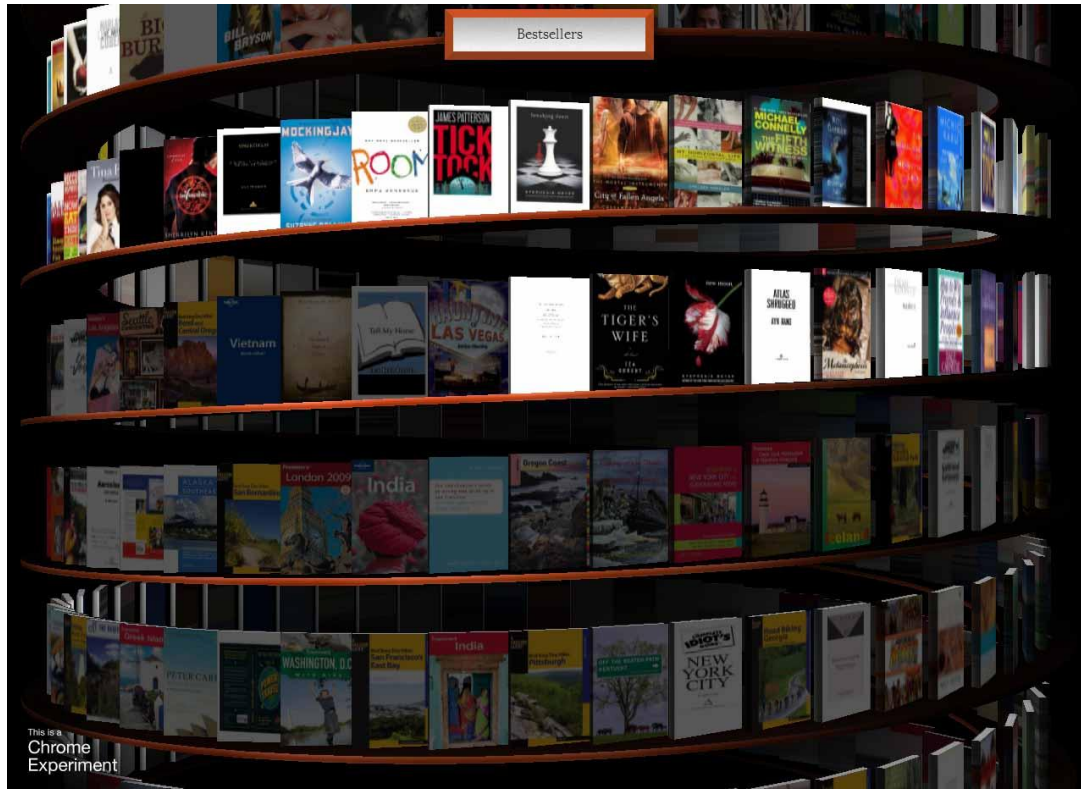
Εικόνα 60 Httrack website copier

- 2) Αφού ολοκληρωθεί το κατέβασμα της εφαρμογής εξετάζουμε προσεκτικά την δομή του κώδικα και την ιεραρχία φόρτωσης των αρχείων καθώς και την χρήση εξωτερικών βιβλιοθηκών που έχουν κατασκευάσει άλλοι (third party software). Με αυτό τον τρόπο «συναρμολογούμε» πάλι την εφαρμογή και με τη βοήθεια του firebug διορθώνουμε όλα τα πιθανά λάθη που προκύπτουν από την μεταφορά των αρχείων και διορθώνουμε τις διαδρομές που έχουν αλλάξει.





- 3) Σε αυτό το σημείο έχουμε μια πλήρως λειτουργική εφαρμογή που τρέχει τοπικά στον υπολογιστή μας. Αυτό που μένει είναι να επεξεργαστούμε τα αρχεία της ώστε να παρουσιάζει τα βιβλία που έχουμε στην δική μας εφαρμογή με τις αντίστοιχες κατηγορίες που θα παράγονται δυναμικά.



Εικόνα 61 Βιβλιοθήκη

- 4) Μεταφέρουμε όλα τα χρήσιμα αρχεία της βιβλιοθήκης στον αντίστοιχο φάκελο όπου φιλοξενούνται τα αρχεία JavaScript και τα αρχεία εικόνων στο Cakerhp και διορθώνουμε τις διαδρομές.
- 5) Βρίσκουμε τα σημεία της εφαρμογής όπου γίνονται τα Ajax requests στο JSON API του Google books και τα αντικαθιστούμε με τα δικά μας δυναμικά αρχεία αφού διαμορφώσουμε την δομή τους σύμφωνα με τα αντίστοιχα που δίνονται ως απάντηση από τα υπάρχοντα JSON αρχεία.

Έτσι δημιουργούμε τέσσερα δυναμικά αρχεία που παράγουν αρχεία json.

Το πρώτο παράγει τις κατηγορίες που στην περίπτωση μας είναι οι χρονολογίες που έχουν δημοσιευμένα άρθρα.

Ελεγκτής:

```
public function dates() {
    $dates = $this->Post->find('list', array('fields' =>
    array('date_published', 'date_published'), 'order' =>
    array('date_published'), 'conditions' =>
    array('Post.published' => '1') ) );
```



```
$this->set(compact('dates'));
$this->Post->recursive = 0;
$this->paginate = array(
    'order' => array(
        'Post.id' => 'asc'
    ),
);
$this->set('posts', $this->Paginator->paginate());
}
```

Εμφάνιση:

```
[<?php
$value='';
foreach ($dates as $date):
    $sum=0;
    foreach ($posts as $thepost):
        if($thepost['Post']['published'] == 1 AND
        $thepost['Post']['date_published']==$date)
            $sum++;
    endforeach;
    if( $sum > 0 ){
        $value .= '{"count": '.$sum.', "name": "'.$date .'}', ' ';
    }
endforeach;
$value = substr_replace( $value, "", -2);//remove last coma
echo $value;
?> ]
```

Το δεύτερο εμφανίζει τον αριθμό των δημοσιευμένων άρθρων

Ελεγκτής:

```
public function count() {
    $sum = $this->Post->find('count', array(
        'conditions' => array('Post.published' =>1 )
    ));
    $this->set(compact('sum'));
}
```

Εμφάνιση:

```
{"count": <?php echo $sum; ?>}
```

Το τρίτο εμφανίζει μια λίστα με τα δημοσιευμένα άρθρα και τα αντίστοιχα id τους συμπληρώνουν την βιβλιοθήκη.

Ελεγκτής:

```
public function thelist(){
```

```
$this->paginate = array(
    'order' => array(
        'Post.category_id' => 'asc'
    ),
);
$this->set('posts', $this->Paginator->paginate());
}
```

Εμφάνιση:

```
[<?php
$value='';
foreach ($posts as $post):
    if($post['Post']['published'] == 1 )
        $value .= '"'. $post['Post']['id'] .'", ' ;
endforeach;
$value = substr_replace( $value, "", -2); //remove last coma
echo $value;
?>]
```

Το τρίτο εμφανίζει τα στοιχεία του κάθε βιβλίου ανάλογα το id που θα του δοθεί ως είσοδο.

Ελεγκτής:

```
public function single($id = null){
    if ($this->Post->exists($id)) {
        if($id == null ) $id=1;
        $this->set(compact('callback'));
        $options = array('conditions' => array('Post.'
        .$this->Post->primaryKey => $id));
        $this->set('post', $this->Post->find('first',
        $options));
    }
}
```

Εμφάνιση:

```
localJsonpCallback({
    "id": "<?php echo $post['Post']['id']; ?>",
    "volumeInfo": {
        "title": "<?php echo $post['Post']['title']; ?>",
        "subtitle": "",
        "authors": [<?php
        $value='';
        foreach ($post['Author'] as $author):
            $value .= '"'. $author['name'] .'", ' ;
        endforeach;
        $value = substr_replace( $value, "", -2); //remove last coma
        echo $value; ?>],
        "description": "<?php echo $post['Post']['body']; ?>",
        "infoLink": "<?php echo '../view/'. $post['Post']['id']; ?>"
    }
});
```

- 6) Πλέον έχουμε αλλάξει τα βιβλία που προβάλλονται και το μόνο που μένει, είναι να αλλάξουμε και τα εξώφυλλά τους. Εδώ θα γίνει μια τροποποίηση στον ήδη υπάρχοντα κώδικα ώστε να εμφανίζονται τα εξώφυλλα που παράγονται δυναμικά από την εφαρμογή μας και όχι τα ομαδοποιημένα εξώφυλλα που έπαιρνε ως είσοδο πριν.

```
spriteSheets[spriteSheetIndex] = THREE.ImageUtils.loadTexture(
    '../..../img/upload/images/' + ids[spriteSheetIndex] +
    '_thumb.png',
    THREE.UVMapping,
    function(img) {
        shouldRender = true;
        if (renderQueued !== null) {
            clearTimeout(renderQueued);
        }
        renderQueued = setTimeout(function() {
            shouldRender = false;
            renderQueued = null;
        }, 1000);

        withRange(min, max, function(bookInfo, bookIndex) {

            bookInfo.status.thumbnail_received = true;

            applicationRequestQueue.push(bookIndex);

        });
    });
```

## 7. Συμπεράσματα

### 7.1 Συνολική αποτίμηση

Τα συμπεράσματα που εξάγονται από την παρούσα πτυχιακή εργασία είναι αρκετά και κατηγοριοποιούνται από το είδος της χρησιμότητας και το όφελος που έφερε στον καθένα. Ο σκοπός αυτής της εργασίας είναι η δημιουργία μιας πλήρως λειτουργικής εφαρμογής που θα μπορεί να φιλοξενεί τα άρθρα του εργαστηρίου πολυμέσων άρα από αυτή την άποψη καλύφθηκε αυτή η ανάγκη και πλέον θα γίνεται με πιο γρήγορο και αυτοματοποιημένο τρόπο. Επίσης με την δημιουργία της τρισδιάστατης βιβλιοθήκης η πλοήγηση στα αρχεία είναι πιο εύκολη και τα αποτελέσματα πιο εμφανίσιμα.

Σε προσωπικό επίπεδο το μεγαλύτερο όφελος από την εκπόνηση αυτής της εργασίας είναι οι εμπειρίες που πήρα και έχουν να κάνουν με την εργασία πάνω στο cakephp framework για πρώτη φορά και η χρήση της τεχνολογίας WebGL που δεν είχα χρησιμοποιήσει ποτέ στο παρελθόν. Επίσης η διαδικασία της σχεδίασης του βέλτιστου μοντέλου για την δημιουργία της εφαρμογής και της σύγκρισης όλων των πιθανών λύσεων βελτίωσε κατά πολύ τις γνώσεις μου σε θέματα κατασκευής δυναμικών εφαρμογών.

### 7.2 Επεκτάσεις

Στο μέλλον αυτή η εφαρμογή θα μπορούσε να επεκταθεί με διάφορους τρόπους αν υπάρξει αυτή η ανάγκη αφού η χρήση του Cakephp είναι σχεδιασμένη γι αυτό. Μια χρήσιμη προσθήκη θα ήταν η διαβάθμιση των δικαιωμάτων των διαφόρων χρηστών ώστε να προστεθούν χρήστες με λιγότερα δικαιώματα από ότι ο διαχειριστής και να μπορούν να επεξεργάζονται μόνο τα άρθρα που οι ίδιοι δημιούργησαν.

## 8. Βιβλιογραφία

1. A Collection of best front End frameworks for Faster & Easier Web Development <http://usablica.github.io/front-end-frameworks/compare.html> (όπως την επισκεφτήκαμε στις 12/12/2013)
2. Responsive web design [http://en.wikipedia.org/wiki/Responsive\\_web\\_design](http://en.wikipedia.org/wiki/Responsive_web_design) (όπως την επισκεφτήκαμε στις 12/12/2013)
3. Responsive web design <http://alistapart.com/article/responsive-web-design/> (όπως την επισκεφτήκαμε στις 15/12/2013)
4. PHP Frameworks <http://www.phpframeworks.com/> (όπως την επισκεφτήκαμε στις 15/12/2013)
5. Html5 <http://en.wikipedia.org/wiki/HTML5> 15/12/2013 (όπως την επισκεφτήκαμε στις 20/12/2013)
6. 12 Awesome CSS3 Features That You Can Finally Start Using <http://tutorialzine.com/2013/10/12-awesome-css3-features-you-can-finally-use/> (όπως την επισκεφτήκαμε στις 2/1/2014)
7. PHP: Hypertext Preprocessor <http://php.net/> (όπως την επισκεφτήκαμε στις 2/1/2014)
8. <http://www.mysql.com/> (όπως την επισκεφτήκαμε στις 2/1/2014)
9. <http://el.wikipedia.org/wiki/JavaScript> (όπως την επισκεφτήκαμε στις 2/1/2014)
10. Learn WebGL by example <http://learnwebgl.blogspot.gr/> (όπως την επισκεφτήκαμε στις 2/1/2014)
11. Wamp <http://noob.gr/web-server-st-pc-sas-me-to-wamp/> (όπως την επισκεφτήκαμε στις 10/2/2014)
12. CakePhp <http://www.web-resources.eu/archives/cakephp-basics> (όπως την επισκεφτήκαμε στις 10/2/2014)
13. Adobe Photoshop [http://en.wikipedia.org/wiki/Adobe\\_Photoshop](http://en.wikipedia.org/wiki/Adobe_Photoshop) (όπως την επισκεφτήκαμε στις 10/2/2014)
14. HTML5 New Elements [http://www.w3schools.com/html/html5\\_new\\_elements.asp](http://www.w3schools.com/html/html5_new_elements.asp) (όπως την επισκεφτήκαμε στις 10/3/2014)
15. Document Object Model (DOM) <http://www.w3.org/DOM/> (όπως την επισκεφτήκαμε στις 10/3/2014)
16. JavaScript HTML DOM [http://www.w3schools.com/js/js\\_htmlDOM.asp](http://www.w3schools.com/js/js_htmlDOM.asp) (όπως την επισκεφτήκαμε στις 10/3/2014)