

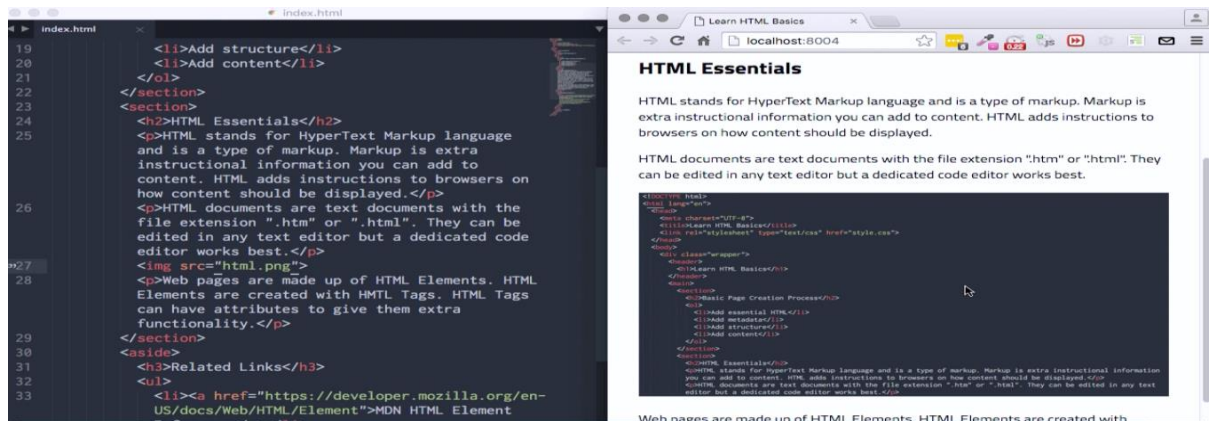
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Προώθηση επίμαχων αναρτήσεων από το FACEBOOK προς site.



Σπουδάστρια: Καλτσά Αντιγόνη (AM 5814)

Επιβλέπων καθηγητής: Στάμος Κωνσταντίνος

Πάτρα 2016

Προώθηση επίμαχων αναρτήσεων από το FACEBOOK προς site.

Περιεχόμενα

| | |
|--|----|
| Περίληψη | 1 |
| Εισαγωγή..... | 3 |
| 1. Θεωρητική ανάλυση των τεχνολογιών του προγράμματος | 6 |
| 1.1 Hypertext Transfer Protocol (HTTP) | 6 |
| 1.2 Το μοντέλο client-server | 7 |
| 1.3 Το επίπεδο client..... | 8 |
| 1.4 HyperText Markup Language (HTML) | 8 |
| 1.5 Cascading Style Sheets (CSS) | 9 |
| 1.6 Twitter Bootstrap..... | 10 |
| 1.7 Javascript..... | 10 |
| 1.8 jQuery..... | 11 |
| 1.9 Asynchronous JavaScript and XML (AJAX) | 12 |
| 1.10 Το επίπεδο server | 13 |
| 1.11 Hypertext Processor (PHP)..... | 14 |
| 1.12 Περιηγητές Ιστού (Web browsers) | 14 |
| 1.13 HTML5 | 15 |
| 1.14 JSON | 16 |
| 1.15 Facebook | 17 |
| 1.16 Facebook Apps | 17 |
| 2. Ανάλυση του πρακτικού μέρους του προγράμματος | 19 |
| 2.1 Εγκατάσταση σουίτας XAMPP: Apache server και PHP language..... | 19 |
| 2.2 Δημιουργία του Facebook App: UniversityOfPeloponnese..... | 20 |
| 2.3 Ανάλυση index.html..... | 22 |
| 2.4 Ανάλυση styles.css | 28 |
| 2.5 Ανάλυση index.js..... | 33 |
| 2.6 Ανάλυση posts.php | 51 |
| 2.7. Ανάλυση posts.html..... | 51 |
| 3. Στιγμιότυπα από τη λειτουργία χρήσης..... | 54 |
| Στιγμιότυπο 1: Πρώτη φόρτωση της σελίδας index.html | 54 |

| | |
|---|----|
| Στιγμιότυπο 2: Ο χρήστης μόλις πάτησε το κουμπί Σάρωση | 55 |
| Στιγμιότυπο 3: Ο χρήστης πάτησε σε ένα σύνδεσμο των comments | 56 |
| Στιγμιότυπο 4: Ο χρήστης πάτησε σε ένα σύνδεσμο των likes | 57 |
| Στιγμιότυπο 5: Ο χρήστης έχει εισάγει 2 φίλτρας ανάρτησης και πάτησε το κουμπί Πρώθηση | 58 |
| Στιγμιότυπο 6: Ο χρήστης έχει εισάγει 3 φίλτρα | 59 |
| Στιγμιότυπο 7: Η σελίδα posts.html | 60 |
| Επίλογος | 61 |
| Βιβλιογραφία | 62 |
| 1. Ελληνική Βιβλιογραφία | 62 |
| 2. Ξενόγλωσση Βιβλιογραφία | 62 |
| 3. Ιστοσελίδες | 62 |

Περίληψη

Στα πλαίσια της τρέχουσας πτυχιακής εργασίας, με θέμα την προώθηση επίμαχων αναρτήσεων από ένα Facebook Group της επιλογής του φοιτητή προς το site του τμήματος της σχολής, απαιτείται η ανάπτυξη προγράμματος/λογισμικού που θα πραγματοποιεί αυτή την λειτουργία. Το Facebook Group που επιλέχθηκε είναι ειδησεογραφικό, το enikos.gr (<https://www.facebook.com/enikosgr/>).

Επίμαχες αναρτήσεις θεωρούνται αυτές που επιλέγονται από το χρήστη ως επίμαχες με βάση συγκεκριμένα κριτήρια ή φίλτρα, όπως θα δούμε παρακάτω, που ορίζει ο ίδιος σε κάθε «σάρωση» των αναρτήσεων. Η προβολή των προωθημένων αναρτήσεων γίνεται σε διαφορετική σελίδα.

Το όφελος που προκύπτει από τη δημιουργία αυτής της εφαρμογής είναι η αυτοματοποίηση που επιτρέπει την ανακάλυψη επίμαχων αναρτήσεων από τον υπολογιστή, σύμφωνα με τα κριτήρια του χρήστη, και την αποθήκευσή τους σε αρχείο. Ο χρήστης κερδίζει σε χρόνο που άλλοτε θα ξόδευε για την ανάλυση κάθε ανάρτησης μία προς μία.

Abstract

At the basis of the current graduate assignment, which is themed as the forwarding of contest posts by a Facebook Group as selected by the student, it is requested the development of software that will handle this operation. The Facebook Group that was selected is a news blog, called enikos.gr. Contest posts are the ones are selected by the user as such based on specific criteria or filters, which will visit in the next sections.

These criteria are set by the user by each posts scan. The forwarded posts are displayed on another page. The advantage that arise by this application development is automation that allows the contest posts' discovery via a computer and then are stored in a file. The user is

benefited in terms of saving time that they would spend of analyzing each post one by one otherwise.

Εισαγωγή

Πόσα likes, shares ή σχόλια γίνονται; Από ποιόν, γιατί και πότε γίνονται όλα αυτά; Ποιες αναρτήσεις είναι οι πιο δημοφιλείς; Μέχρι που μπορεί να φτάσει κάποιος για να κερδίσει ένα like; Όποιος έχει έστω και την παραμικρή επαφή με πλατφόρμες κοινωνικής δικτύωσης γνωρίζει πόσο σημαντικά είναι αυτά τα ερωτήματα.

Πως μπορούμε να αξιοποιήσουμε την κίνηση στο προφίλ μας ώστε να έχουμε τη μέγιστο όφελος ή τη μεγαλύτερη δυνατή αναγνωρισιμότητα και αποδοχή; Σήμερα θεωρείται αδιανόητο να μην έχουν ιδιώτες και επιχειρήσεις το δικό τους προφίλ στο Facebook και στις άλλες πλατφόρμες. Όσοι δεν ανήκουν σε αυτή την κατηγορία σχεδόν αποκλείονται κοινωνικά και δεν θεωρούνται δημοφιλής ενώ οι επιχειρήσεις που δε διαθέτουν ισχυρή διαδικτυακή παρουσία δεν διαθέτουν το ανταγωνιστικό πλεονέκτημα. Και αυτό ισχύει και για τις άλλες πλατφόρμες κοινωνικής δικτύωσης.

Η γνώση είναι δύναμη και αποκτάται αφού μετατραπούν σε δεδομένα οι διάφορες πληροφορίες που προέρχονται από τις κοινωνικές πλατφόρμες, όπως είναι οι αναρτήσεις αλλά και οι αντίστοιχες ανταποκρίσεις είναι πολύ χρήσιμες και στις επιχειρήσεις, τα δημόσια πρόσωπα ή ακόμα και τους ιδιώτες όπως και για τη διερεύνηση της κοινωνικής μας συμπεριφοράς.

Όταν αυτές οι πληροφορίες που δημοσιεύονται στους διαδικτυακούς τόπους μετατρέπονται σε μετρήσιμα μεγέθη δείκτες γίνονται ένα ισχυρό εργαλείο μέτρησης αποδοχής, δημοφιλίας και γενικότερης επισκεψιμότητας ενός προφίλ και συγκεκριμένων αναρτήσεων.

Σκοπός της εργασίας είναι να διαπιστωθεί αν είναι μπορεί να δημιουργηθεί ένα εργαλείο το οποίο να αναζητά σε ένα απίστευτα μεγάλο αριθμό αναρτήσεων, εκείνες που μας ενδιαφέρουν εφόσον θέσουμε συγκεκριμένα κριτήρια όπως τα like και στη συνέχεια να τις προβάλλει σε επιλεγμένη σελίδα/site.

Το εργαλείο που δημιουργήθηκε χρησιμοποιεί τεχνολογίες, οι οποίες αρχικά θα αναλυθούν στο θεωρητικό τους πλαίσιο και θα εξηγηθεί ο λόγος για τον οποίο επιλέχθηκαν. Έπειτα, θα

περιγραφεί ο οδηγός εγκατάστασης των εργαλείων του προγράμματος και τέλος ακολουθεί η λειτουργία του λογισμικού με παράθεση ολόκληρου του κώδικα καθώς και στιγμιότυπων.

Μπορούμε να χωρίσουμε το πρακτικό μέρος της εργασίας σε 2 μέρη: το πρώτο αφορά τη σάρωση των αναρτήσεων από το group, και το δεύτερο την προώθηση αυτών των αναρτήσεων στο site του τμήματος της σχολής. Το τεχνικό πρόβλημα που προκύπτει από το 2^ο μέρος είναι ότι ο φοιτητής δεν έχει πρόσβαση στη σελίδα του τμήματος. Οι κυριότεροι μέθοδοι πρόσβασης στη διαχείριση μίας ιστοσελίδας είναι:

1. Σε επίπεδο αρχείων (FTP, File Transfer Protocol)
2. Σε επίπεδο ιστοσελίδας, (HTTP, Hyper Text Transfer Protocol)
3. Σε επίπεδο ασφαλούς κελύφους (SSH, Secure Shell)

Εφόσον δεν παρέχεται πρόσβαση, χρησιμοποιήθηκε η εναλλακτική εγκατάστασης τοπικού διακομιστή (server) στο μηχάνημα του φοιτητή. Έτσι, μία σελίδα (HTML page) χρησιμοποιείται για τη σάρωση, το φιλτράρισμα και τελικά την προώθηση των αναρτήσεων (posts), ενώ μία άλλη σελίδα χρησιμοποιείται για την εμφάνιση των προωθημένων αναρτήσεων. Το μοντέλο αυτό θεωρώ ότι μπορεί να δουλέψει και στο site του τμήματος. Προτιμήθηκε αυτή η εναλλακτική λύση για να αποδειχθεί ότι η προώθηση των αναρτήσεων του group από 2 διαφορετικές πηγές (cross origin/cross domain) έχει μελετηθεί και είναι εφικτή από το φοιτητή.

Η εφαρμογή που αναπτύχθηκε περιλαμβάνει τη χρήση 2 γλωσσών προγραμματισμού, την PHP για επίπεδο server και τη JavaScript για το επίπεδο πελάτη (client). Επίσης, χρησιμοποιήθηκε η γλώσσα σήμανσης HTML (Hyper Text Markup Language), τα φύλλα στυλ (CSS, Cascading Style Sheets) και το αντικείμενο σημείωση (JSON, JavaScript Notation Object).

Τέλος, για τη διασύνδεση (integration) της εφαρμογής με το Facebook είναι αναγκαία η δημιουργία ενός Facebook app στη σελίδα <https://developers.facebook.com>. Χωρίς την ύπαρξη Facebook app δεν είναι δυνατή η ανάγνωση και διαχείριση των αναρτήσεων.

Στις επόμενες ενότητες θα περιγραφεί η εγκατάσταση τοπικού server στο μηχάνημα του χρήστη και της γλώσσας PHP. Το περιβάλλον εκτέλεσης του λογισμικού είναι ο περιηγητής Ιστού (browser) του χρήστη. Έπειτα, θα περιγραφεί η δημιουργία του Facebook App, θα

εξηγηθεί ο κώδικας του χρήστη και θα επισυναφθούν στιγμιότυπα οθόνης (screenshots) από τη λειτουργία του προγράμματος.

1. Θεωρητική ανάλυση των τεχνολογιών του προγράμματος



1.1 Hypertext Transfer Protocol (HTTP)

Το HTTP είναι πρωτόκολλο εφαρμογών για διανεμημένα, συνεργατικά, υπερ-μεσικά πληροφοριακά συστήματα. Ανακαλύφθηκε ως το πρωτόκολλο για την επικοινωνία δεδομένων μεταξύ διαφορετικών συστημάτων.¹

Πρόκειται για ένα δομημένο κείμενο που χρησιμοποιεί λογικούς συνδέσμους (υπερ-συνδέσμους) μεταξύ κόμβων που περιέχουν κείμενο. Ουσιαστικά, είναι ένα πρωτόκολλο για ανταλλαγή και μεταφορά υπερ-κειμένου (HTML). Το HTTP λειτουργεί ως ένα πρωτόκολλο αιτήματος-απάντησης στο υπολογιστικό μοντέλο client-server.

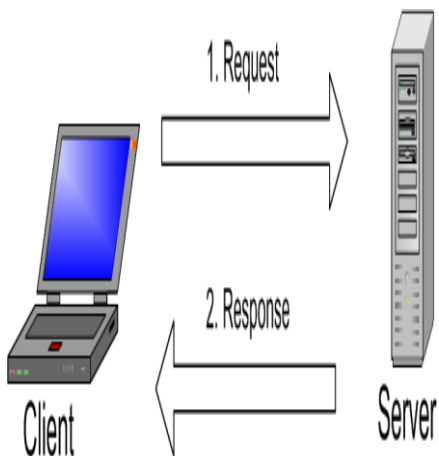
Ένας περιηγητής Ιστού, για παράδειγμα, είναι ο πελάτης, ενώ ο διακομιστής είναι μια εφαρμογή που «τρέχει» σε έναν υπολογιστή που φιλοξενεί την ιστοσελίδα της εφαρμογής. Ο πελάτης υποβάλλει HTTP αιτήματα προς το διακομιστή. Ο διακομιστής, ο οποίος παρέχει πόρους, όπως HTML αρχεία και άλλο περιεχόμενο ή εκτελεί άλλες λειτουργίες προς τον πελάτη, επιστρέφει ένα απαντητικό μήνυμα προς τον πελάτη.

Η απάντηση περιέχει πληροφορίες ολοκλήρωσης της κατάστασης σχετικά με το αίτημα και μπορεί να περιέχει αιτούμενο περιεχόμενο στο σώμα της απάντησης. Το πρωτόκολλο έχει για να επιτρέπει

¹ https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

ενδιάμεσα δικτυακά στοιχεία με σκοπό την ενεργοποίηση και τη βελτίωση της επικοινωνίας μεταξύ πελατών και διακομιστών.

1.2 Το μοντέλο client-server



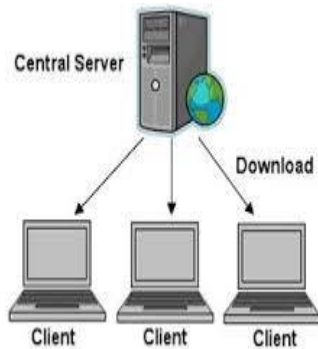
Η εφαρμογή που αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας και συγκεκριμένα οι τεχνολογίες που κάνει χρήση είναι δομημένες στο μοντέλο client-server. Πρόκειται για ένα υπολογιστικό μοντέλο που ενεργεί ως εφαρμογή-διανεμητής, η οποία τεμαχίζει τις εργασίες μεταξύ των παρόχων πόρων ή υπηρεσιών (servers – εξυπηρετητές) και αυτών που ζητάνε τις υπηρεσίες (clients – πελάτες). Στον εξυπηρετητή “τρέχουν” ένα ή περισσότερα προγράμματα τα οποία μοιράζονται πόρους με τους πελάτες.²

Ο πελάτης δεν μοιράζεται κανέναν από τους ιδίους πόρους, αλλά ζητά ή καλύτερα πραγματοποιεί αιτήματα για περιεχόμενο από τον εξυπηρετητή. Οι πελάτες εκκινούν συνεδρίες επικοινωνίας με τους εξυπηρετητές, οι οποίοι αναμένουν για εισερχόμενες αιτήσεις.

Το μοντέλο client-server περιγράφει τη σχέση που έχουν συνεργαζόμενα υπό-προγράμματα σε μία κύρια εφαρμογή. Τα υπό-προγράμματα είναι οι web τεχνολογίες που χρησιμοποιεί η εφαρμογή της εργασίας και θα εξηγηθούν σε επόμενες ενότητες.

² Wikipedia, <http://en.wikipedia.org/wiki/Client-server>

1.3 Το επίπεδο client



Ένας πελάτης (client) είναι τυπικά μία εφαρμογή υπολογιστή, όπως ένας περιηγητής ιστοσελίδων, που “τρέχει” σε τοπικούς σταθμούς εργασίας χρηστών και συνδέεται με έναν εξυπηρετητή όταν είναι απαραίτητο.

Οι λειτουργίες εκτελούνται στην πλευρά του client για το λόγο ότι απαιτούν πρόσβαση σε πληροφορία η οποία είναι διαθέσιμη στον πελάτη και όχι στον εξυπηρετητή. Τέτοιου είδους λειτουργίες είναι συνήθως η απόδοση κειμένου σε μία ιστοσελίδα και η αναμονή για εισαγωγή πληροφορίας από το χρήστη.³



1.4 HyperText Markup Language (HTML)

Η HTML “τρέχει” στην πλευρά του client και είναι η κύρια γλώσσα για την εμφάνιση ιστοσελίδων και άλλων πληροφοριών μέσω ενός περιηγητή. Συντάσσεται με τη μορφή HTML στοιχείων τα οποία αποτελούνται από ετικέτες (tags) που περικλείονται από γωνιακές αγκύλες, όπως `<html>`, εντός του περιεχομένου της ιστοσελίδας.⁴

Οι HTML ετικέτες συναντώνται σε ζεύγη διαθέτοντας ετικέτες αρχής και τέλους, όπως `<h1>` και `</h1>`, ενώ μερικά στοιχεία δεν έχουν ετικέτα τέλους, π.χ. το ``. Ανάμεσα στις ετικέτες τοποθετείται κείμενο, σχόλια και εμφωλευμένες ετικέτες. Ο περιηγητής ιστοσελίδων “διαβάζει” τα έγγραφα HTML και τις μετατρέπει σε

³ Wikipedia, http://en.wikipedia.org/wiki/Client_%28computing%29

⁴ <http://en.wikipedia.org/wiki/Html>

οπτικές ή ακουστικές ιστοσελίδες. Οι HTML ετικέτες δεν εμφανίζονται, αλλά ο περιηγητής τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της ιστοσελίδας.



1.5 Cascading Style Sheets (CSS)

Τα φύλλα ή στυλ CSS είναι μία γλώσσα κανόνων και χρησιμοποιείται για να περιγράψει τη μορφή των αρχείων που έχουν συνταχθεί με μία γλώσσα σήμανσης, όπως την HTML. Κύρια λειτουργία της είναι να μορφοποιεί έγγραφα ιστοσελίδων, προσθέτοντας γραμματοσειρές, χρώμα, στοίχιση και άλλες επιλογές.⁵

Τα στυλ CSS δημιουργήθηκαν για να διαφοροποιήσουν το περιεχόμενο των εγγράφων από την αντίστοιχη μορφοποίηση τους έτσι ώστε να παρέχουν μεγαλύτερη ευελιξία και έλεγχο στην παρουσίαση των στοιχείων των εγγράφων και δίνοντας τη δυνατότητα σε πολλαπλές ιστοσελίδες να μοιράζονται μορφοποιήσεις, μειώνοντας με αυτό τον τρόπο την πολυπλοκότητα και την επανάληψη κώδικα.

Τα CSS περιγράφουν ένα σχέδιο (scheme) προτεραιοτήτων για να καθορίσουν ποιοι κανόνες στυλ εφαρμόζονται σε ένα έγγραφο στην περίπτωση που ισχύουν περισσότεροι του ενός κανόνες. Οι προτεραιότητες, τα βάρη υπολογίζονται και ανατίθενται σε κανόνες και ισχύουν σε ένα έγγραφο διαδοχικά (cascade).

⁵ Wikipedia, <http://en.wikipedia.org/wiki/Css>



1.6 Twitter Bootstrap

Το Twitter Bootstrap είναι μία ελεύθερη και ανοιχτού κώδικα βιβλιοθήκη για προγραμματισμό του σχεδιαστικού μέρους των ιστοσελίδων και διαδικτυακών εφαρμογών και δημιουργήθηκε από το Twitter.⁶ Περιέχει HTML και CSS σχεδιαστικά πρότυπα για τυπογραφία, φόρμες, κουμπιά, πλοήγηση και άλλα συστατικά γραφικής διεπαφής, αλλά και επεκτάσεις Javascript. Ο σκοπός της βιβλιοθήκης είναι να επιταχύνει και να διευκολύνει σημαντικά την ανάπτυξη εφαρμογών.⁷



1.7 Javascript

Η γλώσσα client Javascript είναι μία αντικειμενοστραφής δυναμική, χαλαρών τύπων γλώσσα σεναρίων βασισμένη στο Prototype (οι κλάσεις δεν είναι ορατές στον προγραμματιστή) και υποστηρίζει διαδικαστικά και συναρτησιακά προγραμματιστικά στυλ. Το Javascript κώδικα εκτελούν οι περιηγητές ιστού και χρησιμοποιείται για να προσδώσει σε ένα έγγραφο ιστοσελίδας εμπλουτισμένη διεπαφή με το χρήστη και δυναμικότητα.⁸

Η δημοτικότητα της Javascript ήρθε με τον καιρό καθώς ολοένα και περισσότεροι προγραμματιστές επιλέγουν να τρέχουν σενάρια ιστοσελίδων στον υπολογιστή του χρήστη μοιράζοντας έτσι το φόρτο

⁶ [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

⁷ <http://getbootstrap.com/getting-started/>

⁸ Wikipedia ,<http://en.wikipedia.org/wiki/JavaScript>

εργασίας ανάμεσα στον server και στον client δεδομένου ότι η ισχύς και οι δυνατότητες των υπολογιστών από τη μεριά των χρηστών συνεχώς αυξάνονται.

Η συγκεκριμένη γλώσσα σε μερικές περιπτώσεις καθίσταται απαραίτητη, για το λόγο ότι το γραφικό περιβάλλον χρήστη, το οποίο χειρίζεται η πλειοψηφία των συναρτήσεων, καθίσταται όλο και πιο σημαντικό, αφού σε πολλές περιπτώσεις μία ιστοσελίδα “κρίνεται” όχι μόνο μέσα από το περιεχόμενο αλλά και από την εμφάνιση.



1.8 jQuery

Το 99% του κώδικα της εφαρμογής στη διατριβή χρησιμοποιεί τη βιβλιοθήκη της Javascript, την jQuery. Πρόκειται για μία ελεύθερη, ανοικτού κώδικα γλώσσα που σχεδιάστηκε το 2006 για να απλοποιήσει τη γραφή σεναρίων στην πλευρά του client. Στατιστικά χρησιμοποιείται από το 55% των 10.000 ιστοσελίδων με τη μεγαλύτερη επισκεψιμότητα. Ένα από τα πλεονεκτήματά της είναι η ερμηνεία των σεναρίων το ίδιο από όλους τους περιηγητές, δίνοντας λύση στο πρόβλημα οι εντολές της Javascript (χωρίς τη βιβλιοθήκη jQuery) να ερμηνεύονται διαφορετικά.⁹

Η jQuery βιβλιοθήκη σταδιακά γίνεται πιο επιθυμητή από τους προγραμματιστές σε σχέση με την απλή Javascript για το λόγο ότι προσφέρει μεγαλύτερη ευελιξία και έλεγχο στα στοιχεία της HTML, των συμβάντων, τη δημιουργία animations και γενικότερα καλύτερη εμπειρία χρήσης και διαδραστικότητα. Η συγγραφή σεναρίων

⁹ <http://jquery.com/>

θεωρείται υψηλού επιπέδου και είναι περισσότερο κατανοητή από τον προγραμματιστή, απαιτεί λιγότερο κώδικα, ο οποίος είναι και ευκολότερα συντηρήσιμος είναι μερικά από τα πλεονεκτήματα της βιβλιοθήκης.

Παρέχει δυνατότητες για επέκταση βιβλιοθηκών αλλά και για δημιουργία plugins (παρόμοια με τις έτοιμες κλάσεις που συναντώνται σε άλλες γλώσσες προγραμματισμού), τα οποία απλοποιούν ορισμένες επιλογές της Javascript αποκρύπτοντας το δυσνόητο κώδικα για τον προγραμματιστή.¹⁰



1.9 Asynchronous JavaScript and XML (AJAX)

Το ακρωνύμιο AJAX είναι ένα σύνολο αλληλοσυνδεδεμένων τεχνικών ανάπτυξης στο Διαδίκτυο που χρησιμοποιείται στην πλευρά του client για τη δημιουργία ασύγχρονων web εφαρμογών. Χάρη στο AJAX οι web εφαρμογές δύνανται να στείλουν και να λάβουν δεδομένα από ένα server ασύγχρονα (στο παρασκήνιο) χωρίς να επηρεάσουν την παρουσίαση και τη συμπεριφορά μίας ιστοσελίδας.¹¹

Η πιο συχνή χρήση τους αφορά στην ανταλλαγή δεδομένων από τον client στο server και αντίστροφα χωρίς την απαίτηση για ανανέωση της ιστοσελίδας. Επιτρέπει τη διενέργεια αιτημάτων (requests) προς τον διακομιστή από τον περιηγητή ιστού στο παρασκήνιο.

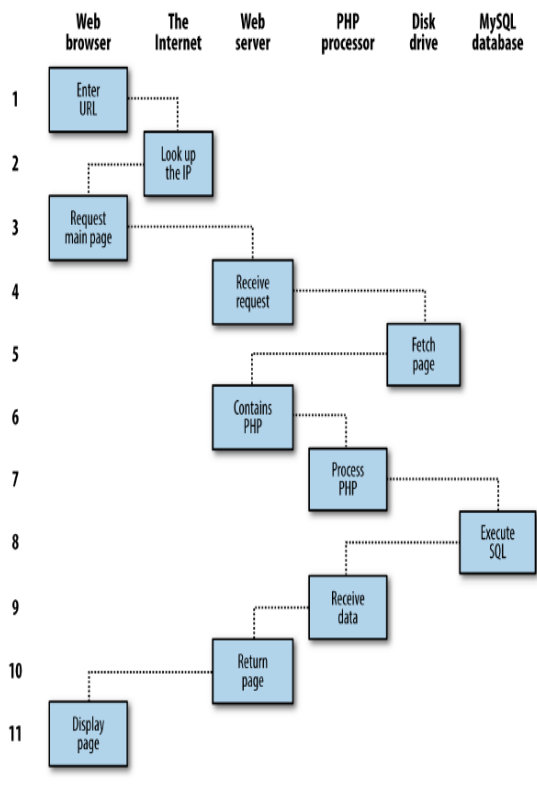
Θεωρείται μία σημαντική τεχνική που χρησιμοποιείται ευρέως από τα web προγράμματα: προσδίδει δυναμικότητα σε μια ιστοσελίδα,

¹⁰ Bibeault B. et al (2015) *jQuery in Action* - Second Edition, Manning Publications

¹¹ <http://en.wikipedia.org/wiki/Ajax>

εξοικονομεί χρόνο και στο χρήστη και στον προγραμματιστή ανταλλάσσοντας μόνο τα χρήσιμα δεδομένα και αυξάνει την παραγωγικότητα του τελευταίου.

1.10 Το επίπεδο server



Το επίπεδο server περιγράφει τις λειτουργίες που εκτελούνται από έναν εξυπηρετητή σε ένα μοντέλο client-server στη διασύνδεση υπολογιστών. Τυπικά, ένας server είναι ένα λογισμικό, το οποίο προσπελάζεται από σταθμούς εργασίας χρηστών.

Οι διάφορες λειτουργίες επιλέγονται να γίνουν στην πλευρά του server, επειδή απαιτούν πρόσβαση σε πληροφορία ή λειτουργικότητα, τα οποία είτε δεν είναι διαθέσιμα στην **πλευρά του πελάτη** είτε η **πλευρά client** δε θεωρείται αξιόπιστη στην προκειμένη περίπτωση.¹²

Πρόκειται για λειτουργίες που περιλαμβάνουν επεξεργασία και αποθήκευση δεδομένων από έναν client σε ένα server και διασύνδεση με βάση δεδομένων, τα αποτελέσματα των οποίων παρουσιάζονται στους χρήστες με τη μορφή ιστοσελίδων.

¹² Wikipedia, <http://en.wikipedia.org/wiki/Client-server>



1.11 Hypertext Processor (PHP)

Μία γλώσσα προγραμματισμού server αποτελεί και η PHP με τα στατιστικά χρήσης της σε ιστοσελίδες να είναι πάνω από το 77% για το 2012. Είναι μία ανοικτού κώδικα, αντικειμενοστραφής, διαδικαστική, ανεξάρτητη από πλατφόρμα, γενικού σκοπού γλώσσα σεναρίων που σχεδιάστηκε για την ανάπτυξη διαδικτυακών και δυναμικών εφαρμογών.¹³

Ο κώδικάς της μεταγλωττίζεται σε έναν web server που διαθέτει μία υπό-μονάδα PHP επεξεργαστή, ο οποίος παράγει το τελικό αποτέλεσμα σε μία ιστοσελίδα.

Κύριες λειτουργίες της είναι η διαχείριση αρχείων και η εισαγωγή από το χρήστη, αλλά και η σύνδεση με βάσεις δεδομένων. Ενσωματώνεται σε έγγραφα HTML, είναι “ελαφριά” στη χρήση και σε πολλά σημεία γρηγορότερη σε σχέση με τους ανταγωνιστές της, όπως ASP και Coldfusion.



1.12 Περιηγητές Ιστού (Web browsers)

Ένας περιηγητής ιστού είναι μία εφαρμογή για ανάκτηση, παρουσίαση και διανομή πληροφοριακών πηγών στο World Wide Web. Μία πληροφοριακή πηγή προσδιορίζεται από ένα καθολικό αναγνωριστικό πηγής ή τοποθεσίας (Uniform Resource Identifier - URL/URI), το οποίο μπορεί να είναι μία ιστοσελίδα, εικόνα, βίντεο ή οποιοδήποτε άλλο κομμάτι που περιέχει πληροφορία.

¹³ Wikipedia, <http://en.wikipedia.org/wiki/Php>

Οι υπερσύνδεσμοι (hyperlinks) που εντοπίζονται σε αυτές τις πηγές επιτρέπουν στους χρήστες να περιηγηθούν σε άλλες τοποθεσίες και αυτό δείχνει το συσχετισμό των τοποθεσιών ή κόμβων μεταξύ τους.

Παρόλο που οι περιηγητές (ιστού) προορίζονται κυρίως για το World Wide Web, μπορούν επίσης να χρησιμοποιηθούν για πρόσβαση σε πληροφορίες που παρέχονται από τους εξυπηρετητές σε ιδιωτικά δίκτυα (private networks) ή αρχείων σε συστήματα αρχείων (file systems). Ο περιηγητής ιστού είναι το λογισμικό που επεξεργάζεται και προβάλλει τις πληροφορίες που παρουσιάζονται στην πτυχιακή εργασία.¹⁴



1.13 HTML5

Η HTML5 είναι μία γλώσσα σήμανσης που χρησιμοποιείται για προβολή πληροφορίας στο World Wide Web και είναι μία θεμελιώδης τεχνολογία του Διαδικτύου. Βρίσκεται στην πέμπτη του HTML προτύπου και θεωρείται υποψήφια για υιοθέτηση το World Wide Consortium (W3C).¹⁵

Οι κύριοι στόχοι της τεχνολογίας έχουν να κάνουν με τη βελτίωση της γλώσσας και για υποστήριξη των τελευταίων πολυμέσων, ενώ παράλληλα να διατηρεί την άνετη αλληλεπίδραση τόσο από ανθρώπους όσο και από υπολογιστές, όπως οι περιηγητές ιστού.

Σχεδόν όλοι οι περιηγητές τον υποστηρίζουν, αλλά από συγκεκριμένες εκδόσεις και πάνω, ενώ υπάρχει ολοένα αυξανόμενη υποστήριξη της νέας τεχνολογίας. Ο λόγος αναφοράς της HTML5 τεχνολογίας είναι ότι

¹⁴ https://en.wikipedia.org/wiki/Web_browser

¹⁵ <http://en.wikipedia.org/wiki/HTML5>

η εφαρμογή που αναπτύχθηκε υποστηρίζεται μόνο από HTML5 browsers. Σε παλαιότερες εκδόσεις ενδέχεται το πρόγραμμα να λειτουργεί με προβλήματα.



1.14 JSON

Το JSON (Javascript Object Notation) είναι μία ανοικτού προτύπου αναφορά, η οποία χρησιμοποιεί κείμενο κατανοητό από τον άνθρωπο για τη μετάδοση αντικειμένων που περιέχουν δεδομένα, τα οποία αποτελούνται από ζευγάρια ιδιοτήτων-τιμών.

Είναι η πιο συχνά χρησιμοποιούμενη αναφορά δεδομένων που εκμεταλλεύεται προς χρήση για ασύγχρονη επικοινωνία μεταξύ περιηγητών και διακομιστών (AJAX).

Η προέλευσή της είναι η γλώσσα Javascript, αλλά πλέον είναι ανεξάρτητη γλώσσας προγραμματισμού. Η μορφή ενός αντικειμένου JSON μοιάζει αρκετά με δημοφιλείς γλώσσες, όπου το αριστερό μέρος (ιδιότητα ή μεταβλητή) ισούται με μια τιμή. Επιπλέον, μια ιδιότητα υπερ-σύνολο μπορεί να ισούται με ένα υπο-σύνολο ζευγαριών ιδιοτήτων-τιμών, μορφή που μοιάζει με τις μήτρες (arrays) στην Javascript.¹⁶

Η έλευση αυτής της αναφοράς έλυσε αρκετά προβλήματα ασύγχρονης φόρτωσης δεδομένων ιστοσελίδων, αφού το μέγεθος της αναφοράς είναι συνήθως μικρό σε μέγεθος. Η τεχνολογία JSON χρησιμοποιείται επίσης από Javascript Facebook Applications.

¹⁶ <http://en.wikipedia.org/wiki/JSON>



1.15 Facebook

Facebook είναι μία ιστοσελίδα κοινωνικής δικτύωσης, που επιτρέπει την εύκολη διαδικτυακή επικοινωνία μεταξύ οικογένειας και φίλων. Η ιστοσελίδα, επίσης, παρέχει υπηρεσίες τόσο σε απλούς χρήστες όσο και σε εταιρείες και προγραμματιστές εφαρμογών.

Οι εταιρείες μπορούν να καταναλώσουν τις υπηρεσίες για διαδικτυακή προβολή αυτών και των προϊόντων καταλόγων που διαθέτουν, ενώ οι προγραμματιστές μπορούν να αξιοποιήσουν δημόσια στοιχεία άλλων χρηστών για χρήσεις όπως κοινωνική ανάλυση και καταναλωτικών προτιμήσεων.¹⁷



1.16 Facebook Apps

Οι εφαρμογές Facebook αναπτύσσονται για κατανάλωση των βασικών τεχνολογιών που διαθέτει για δημιουργία επεκτάσιμων κοινωνικών μέσων του πλαισίου εργασίας που παρέχει η Facebook πλατφόρμα. Οι εφαρμογές αυτές ενσωματώνουν ροές ειδήσεων, ειδοποιήσεις και διάφορα άλλα κοινωνικά κανάλια και χαρακτηριστικά για την προσέλκυση νέων και υπαρχόντων χρηστών προς αυτές.¹⁸

Πολλές εφημερίδες και ειδησεογραφικά πρακτορεία τροφοδοτούν με ειδήσεις και άρθρα τις Facebook σελίδες και ομάδες που διαθέτουν προς το κοινό με σκοπό την επέκταση σε νέες

¹⁷ <http://www.webopedia.com/TERM/F/Facebook.html>

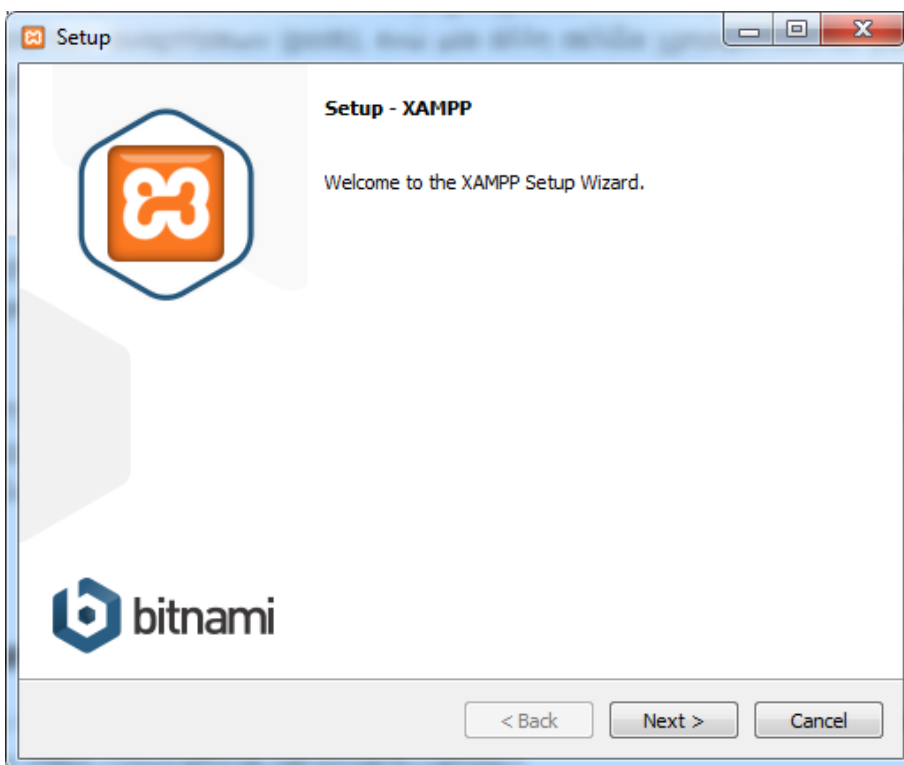
¹⁸ <http://code.tutsplus.com/tutorials/build-your-first-facebook-app--net-471>

αγορές. Στην παρούσα εργασία γίνεται χρήση της ροής ειδήσεων του Facebook Group: [enikos.gr](https://www.facebook.com/enikos.gr).

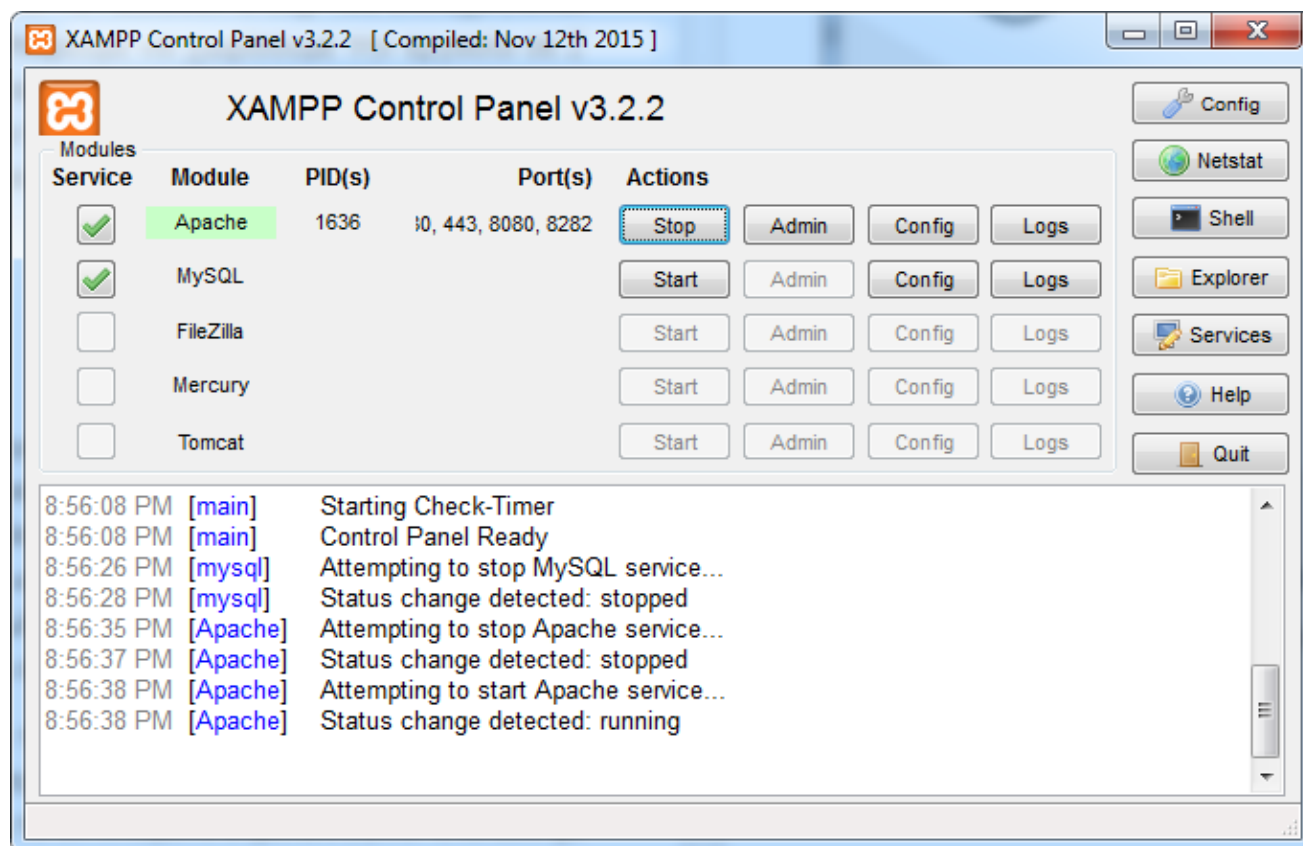
2. Ανάλυση του πρακτικού μέρους του προγράμματος

2.1 Εγκατάσταση σουίτας XAMPP: Apache server και PHP language

Θα κατεβάσουμε το πρόγραμμα XAMPP, το οποίο αυτοματοποιεί την εγκατάσταση Apache server και PHP σε λειτουργικό Windows (<https://sourceforge.net/projects/xampp/>). Ακολουθούμε τον οδηγό εγκατάστασης. Ανοίγουμε το πρόγραμμα και εκκινούμε (Start) τον Apache.



Στιγμιότυπο 1: Εγκατάσταση XAMPP



Στιγμιότυπο 2: Εκτέλεση XAMPP και εκκίνηση Apache server

Θα χρειαστούμε τώρα ένα πρόγραμμα το οποίο θα γράψουμε τον κώδικα του προγράμματος. Επιλέγουμε το Notepad++ (<https://notepad-plus-plus.org/>), το οποίο μορφοποιεί τον κώδικα για να είναι πιο ευανάγνωστος.

2.2 Δημιουργία του Facebook App: UniversityOfPeloponnese

Η δημιουργία και η χρήση ενός Facebook app είναι αρκετά απλή διαδικασία που πραγματοποιείται με λίγα clicks. Προϋποθέτει ότι ο χρήστης έχει λογαριασμό στο Facebook. Απαιτούνται μόνο η επιλογή του τύπου του app (επελέγη το Website) και το όνομα που θα έχει το app. Παρακάτω, βλέπουμε τον πίνακα ελέγχου (Dashboard) του app. Αυτά που μας ενδιαφέρουν και θα χρησιμοποιήσουμε στη δική μας εφαρμογή είναι το App ID και το App Secret.

The screenshot shows the Facebook App Dashboard for an application named "UniversityOfPeloponnese". On the left is a navigation sidebar with the following items: Dashboard (selected), Settings, App Review, App Details, Roles, Open Graph, Alerts, Localize, Canvas Payments, Audience Network, Test Apps, WebHooks, and Analytics. The main content area displays the app's profile, including its logo (a blue square with a white atomic symbol), the name "UniversityOfPeloponnese", and a status "This app is in development mode and can only be used by app admins, developers and testers [?]". Below this, the API Version is listed as "v2.5" and the App ID as "686497771493476". The App Secret is masked with dots and has a "Show" button. Three main sections are visible: "Get Started with the Facebook SDK" with a "Choose a Platform" button, "Facebook Analytics for Apps" with a "Get Analytics and Trends" section and a "Try It Now" button, and "Facebook Login".

Στιγμιότυπο 3: Πίνακας ελέγχου Facebook App

2.3 Ανάλυση index.html

```
<html>
<head>
<title>Facebook: Προώθηση επίμαχων αναρτήσεων</title>
<meta charset="UTF-8">
<link
href='https://fonts.googleapis.com/css?family=Ubuntu:400,300,300italic,400italic,500,500
italic,700&subset=greek,latin' rel='stylesheet' type='text/css'>
<link rel="stylesheet" type="text/css" href="style.css">
<script src="jquery.js"></script>
<script src="index.js"></script>
</head>
<body>

<div align="center">
  <h3>Προώθηση Επίμαχων Αναρτήσεων από Facebook Group: Enikos.gr</h3>
  <input type="button" id="scan-btn" disabled="disabled" value="Σάρωση" />
</div>

<div id="filters">
  <h3>Φίλτρα</h3>

  <div class="filter">
    <span>1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:</span>
    <input type="text" id="filter-title" />
  </div>

  <div class="filter">
    <span>2. Η ανάρτηση έχει πάνω από:</span>
    <select id="filter-like">
```

```

        <option value="">-</option>
        <option value="5">5</option>
        <option value="10">10</option>
        <option value="15">15</option>
        <option value="20">20</option>
        <option value="50">50</option>
    </select>
    <span>likes</span>
</div>

<div class="filter">
    <span>3. Η ανάρτηση έχει πάνω από:</span>
    <select id="filter-comment">
        <option value="">-</option>
        <option value="5">5</option>
        <option value="10">10</option>
        <option value="15">15</option>
        <option value="20">20</option>
    </select>
    <span>comments</span>
</div>

<div class="filter">
    <span>4. Η ανάρτηση έχει πάνω από:</span>
    <select id="filter-share">
        <option value="">-</option>
        <option value="5">5</option>
        <option value="10">10</option>
        <option value="15">15</option>
        <option value="20">20</option>
    </select>
    <span>shares</span>
</div>

```

```

        </select>
        <span>shares</span>
    </div>

    <div class="filter">
        <span>5. Το like έγινε από:</span>
        <input type="text" id="filter-like-name" />
    </div>

    <div class="filter">
        <span>6. Το comment έγινε από:</span>
        <input type="text" id="filter-comment-name" />
    </div>

    <div class="filter">
        <span>7. Το comment περιέχει το κείμενο:</span>
        <input type="text" id="filter-comment-message" />
    </div>

    <input type="button" id="filter-btn" value="Προώθηση" />
    <div id="message">Επιτυχής προώθηση</div>

</div>

<div class="wrap wrap-1">
    <h4>Αποτελέσματα σάρωσης</h4>
    <table id="scan-results">
        <thead>
            <tr>

```

```

    <th class="aa">A/A</th>
    <th class="title">Τίτλος</th>
    <th class="desc">Προεπισκόπηση</th>
    <th class="type">Τύπος</th>
    <th class="pic">Εικόνα</th>
    <th class="likes_cnt">Likes</th>
    <th class="comments_cnt">Σχόλια</th>
    <th class="shares_cnt">Shares</th>
  </tr>
</thead>
<tbody></tbody>
</table>
</div>

```

```

<div class="wrap wrap-2">
  <h4>Ανάλυση Likes</h4>
  <table id="like-results">
    <thead>
      <tr>
        <th class="aa">A/A</th>
        <th class="name">Χρήστης</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>
</div>

```

```

<div class="wrap wrap-3">
  <h4>Ανάλυση Comments</h4>
  <table id="comment-results">

```

```

        <thead>
            <tr>
                <th class="aa">Α/Α</th>
                <th class="name">Χρήστης</th>
                <th class="message">Σχόλιο</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
</div>

</body>
</html>

```

Το πρώτο αρχείο που θα γράψουμε είναι το index.html. Η ονομασία του αρχείου επιλέγη να είναι αυτή, καθώς είναι το πρώτο αρχείο που εκτελεί ο server. Ένα αρχείο HTML οργανώνεται σε κόμβους, οι οποίοι έχουν ιδιότητες και περιεχόμενο. Κάθε κόμβος περιέχει ένθετους κόμβους και ανοίγουν με γωνιακές αγκύλες (<>) και κλείνουν με πάλι με γωνιακές αγκύλες και το σύμβολο slash (/) εντός των αγκύλων.

Ο αρχικός κόμβος είναι ο <html>. Ακολουθεί ο κόμβος <head> που περιέχει οδηγίες για τη σελίδα, δεν εμφανίζουν περιεχόμενο. Αυτός περιέχει το <title>Facebook: Προώθηση επίμαχων αναρτήσεων</title>, που είναι ο τίτλος της σελίδας.

Μετά ακολουθεί το <meta charset="UTF-8">. Είναι οδηγία προς τον browser έτσι ώστε να αναγνωρίσει το περιεχόμενο της σελίδας ως ελληνικά. Ο κόμβος επίσης περιέχει 2 κόμβους <link>, ο πρώτος χρησιμοποιείται για την εισαγωγή μιας γραμματοσειράς της Google, την Ubuntu. Είναι μία γραμματοσειρά που δεν υπάρχει εγκατεστημένη στους browsers, γι' αυτό και πρέπει να την εισάγουμε.

Ο 2^{ος} κόμβος εισάγει ένα αρχείο φύλλου στυλ (style.css). Σε αυτό το αρχείο έχει γραφτεί τη μορφοποίηση της σελίδας. Ο κόμβος <head> τελειώνει με την εισαγωγή 2 JavaScript βιβλιοθηκών (libraries). Πρόκειται για κώδικα, γραμμένο από προγραμματιστές και βοηθά

αλλά και επιταχύνει τη δημιουργία και τη συντήρηση του λογισμικού. Αυτές οι βιβλιοθήκες εισάγονται στους κόμβους `<script>`.

Η πρώτη βιβλιοθήκη που εισάγεται είναι η jQuery. Είναι η πιο γνωστή βιβλιοθήκη τύπου JavaScript, η οποία είναι βοηθητική στη ανάπτυξη κώδικα.

Η επόμενη βιβλιοθήκη είναι κώδικας χρήστη (index.js) και είναι το μέρος που διασυνδέεται με τον κώδικα του Facebook. Ο κόμβος `<body>` δείχνει περιεχόμενο στη σελίδα και καταλαμβάνει το μεγαλύτερο μέρος HTML.

Ο πρώτος ένθετος κόμβος του `<body>` είναι ένας κόμβος `<div>`. Έχει ως ιδιότητα τη στοίχιση του περιεχομένου στο κέντρο της σελίδας: `align="center"`. Αυτός περιέχει έναν κόμβο `<h3>` με τον υπότιτλο της σελίδας (Προώθηση Επίμαχων Αναρτήσεων από Facebook Group: Enikos.gr) και ένα `n` κόμβο κουμπιού `<input>`, που ο χρήστης το πατάει για να σαρώσει περιεχόμενο από το Facebook Group.

Ακολουθούν τα φίλτρα, `<div id="filters">`, που δομείται από τον τίτλο κάθε φίλτρου και το αντίστοιχο στοιχείο ελέγχου. Συνολικά έχουμε 7 φίλτρα. Το πρώτο φίλτρο έχει ως στοιχείο ελέγχου τον τίτλο της ανάρτησης να περιέχει το κείμενο που εισάγει ο χρήστης. Επομένως, ο κόμβος `<input type="text">`, είναι ο επιθυμητός κόμβος για αυτή τη δουλειά. Το 2^ο φίλτρο επιβάλλει τις αναρτήσεις να έχουν ένα αριθμό likes και άνω για να προωθηθούν.

Οι επιλογές είναι: 5, 10, 15, 20, 50. Συμπερασματικά, αναρτήσεις με λιγότερα από 5 likes δεν προωθούνται εάν έχει επιλέξει αυτό το φίλτρο ο χρήστης. Η επιλογή του φίλτρου γίνεται με τον κόμβο `<select>` και έναν ή περισσότερους ένθετους κόμβους `<option>`. Η τιμή κάθε επιλογής `<option>` εισάγεται στην ιδιότητα `value`.

Το ίδιο μοτίβο συναντούμε και στο φίλτρο των comments. Εάν επιλεγεί από το χρήστη, θα προωθηθούν αναρτήσεις με αριθμό 10 comments και άνω σε επιλογή 10. Το ίδιο συμβαίνει και με το 4^ο φίλτρο, το οποίο εάν το επιλέξει ο χρήστης θα προωθήσει αναρτήσεις που έχουν shares, δηλαδή έχουν μοιραστεί από χρήστες, από τον επιλεγμένο αριθμό και άνω.

Τα 3 τελευταία φίλτρα είναι περισσότερο εξειδικευμένα και εάν επιλεγούν, απαιτούν είσοδο κειμένου από το χρήστη. Το 5^ο φίλτρο καθορίζει ότι το like έγινε από κάποιο χρήστη (όνομα χρήστη), το 6^ο φίλτρο καθορίζει ότι το comment έγινε από κάποιο χρήστη (όνομα χρήστη) και το 7^ο φίλτρο απαιτεί το comment να περιέχει το κείμενο, ή μέρος του, που εισάγει το χρήστης.

Η σελίδα τελειώνει με 3 πίνακες. Οι πίνακες στην HTML αντιπροσωπεύονται με τον κόμβο <table>. Ο πίνακας μπορεί να έχει επικεφαλίδα <thead>, σώμα <tbody>, τα οποία περιέχουν γραμμές <tr> και κελιά <td>. Ο 1^{ος} πίνακας δείχνει τα αποτελέσματα σάρωσης από το Facebook Group. Έχει 8 στήλες:

1. Αύξων αριθμός
2. Τίτλος (ανάρτησης)
3. Προεπισκόπηση (ανάρτησης)
4. Τύπος (ανάρτησης)
5. Εικόνα (ανάρτησης)
6. Likes (αριθμός)
7. Comments (αριθμός)
8. Shares (αριθμός)

Αρχικά ο πίνακας είναι κενός και αποκτά περιεχόμενο όταν ο χρήστης πατήσει το κουμπί Σάρωση. Θα εμφανιστούν οι τελευταίες 100 αναρτήσεις ταξινομημένες ανά χρονική σειρά (περισσότερο πρόσφατη με λιγότερο πρόσφατη). Ο 2^{ος} πίνακας περιέχει όλα τα likes μαζί με τους χρήστες από τους οποίους έγιναν για την επιλεγμένη ανάρτηση.

Το περιεχόμενο του πίνακα είναι κενό και ο χρήστης πρέπει να πατήσει πάνω σε έναν αριθμό likes σε μία ανάρτηση έτσι ώστε το περιεχόμενο να συμπληρωθεί. Το αντίστοιχο ισχύει και για τον 3^ο πίνακα, ο οποίος εμφανίζει τα comments και τους χρήστες από τα οποία έγιναν για κάθε ανάρτηση. Ο χρήστης πρέπει να πατήσει στον αριθμό των comments κάθε ανάρτησης.

2.4 Ανάλυση styles.css

Τα φύλλα στυλ καθορίζουν τη μορφοποίηση του περιεχομένου της ιστοσελίδας. Πρώτα καθορίζεται το/α στοιχείο/α επιλογής και δεύτερον καθορίζεται ο κανόνας στυλ.

```
body, input{
    font-family: 'Ubuntu', sans-serif;
}
```


Επιλογή του κόμβου <body> και του κόμβου <input> να έχουν την Ubuntu γραμματοσειρά, η οποία έχει εισαχθεί στη σελίδα index.html.

```
h4{
    text-align: center;
}
```

Ο κόμβος <h4> στοιχίζει το περιεχόμενό του στο κέντρο της σελίδας.

```
.filter input{
    width: 250px;
    height: 30px;
    padding: 5px;
}
```

Η κλάση filter βρίσκεται σε κάθε κόμβο <div> που περιέχει φίλτρο. Στον άνω κανόνα αυτός ο κόμβος ονομάζεται πατέρας, ενώ ο ένθετος κόμβος <input> που περιέχεται εντός του <div>, ονομάζεται παιδί. Στη συγκεκριμένη περίπτωση ο κόμβος επιλογής είναι <input> που είναι παιδί του κόμβου <div class="filter">, με τον πρώτο να έχει σταθερό μήκος, ύψος και εσωτερικό περιθώριο.

```
select{
    width: 50px;
}
```

Ορισμός σταθερού μήκους για όλους τους κόμβους <select>.

```
table td{
    text-align: center;
    border: 1px solid #c0c0c0;
}
```

Ορισμός όλων των κελιών <td> που βρίσκονται σε όλους τους πίνακες <table> να στοιχίζονται στο κέντρο και τα 4 όρια τους να έχουν το δεκα-εξαδικό χρώμα: #c0c0c0.

```
.title{
    max-width: 200px;
}
```

Αυτή η κλάση βρίσκεται σε κάθε κελί <td> του πίνακα και φιλοξενεί όλους τους τίτλους των αναρτήσεων. Ορίζεται το μέγιστο μήκος του κελιού, καθώς το μήκος των κελιών ενός πίνακα διαμορφώνεται ως προεπιλογή από το μέγεθος του περιεχομένου.

```
.aa{
    width: 40px;
}
```

Η κλάση που βρίσκεται σε κελιά πίνακα που φιλοξενούν τον αύξοντα αριθμό κάθε ανάρτησης έχει ως κανόνα το σταθερό μήκος.

```
.desc{
    max-width: 500px;
}
```

Η κλάση που βρίσκεται σε κελιά πίνακα που φιλοξενούν την προεπισκόπηση κάθε ανάρτησης έχει ως κανόνα το μέγιστο μήκος του κελιού.

```
.pic img{
    width: 180px;
}
```

Τα κελιά που περιέχουν εικόνα έχουν την κλάση pic. Αυτά έχουν παιδιά τις εικόνες κόμβους που σαρώνονται από το Facebook Group. Όλες οι εικόνες που είναι παιδιά αυτής της κλάσης έχουν σταθερό μήκος.

```
.user{
    width: 160px;
}
```

Η κλάση αυτή εισάγεται δυναμικά μέσω κώδικα JavaScript και χαρακτηρίζει κελί πίνακα που περιλαμβάνει όνομα χρήστη για τους πίνακες likes και comments.

```
.message{
    max-width: 200px;
}
```

Τη κλάση message τη συναντούμε στον πίνακα των comments και ειδικότερα στο κελί του κειμένου κάθε comment.

```
.wrap{
    float: left;
    display: none;
}
```

Η κλάση wrap είναι άμεσος πατέρας κάθε πίνακα. Αρχικά, κατά τη φόρτωση της σελίδας, οι πίνακες δεν εμφανίζονται. Όλοι οι πίνακες είναι τοποθετημένοι αριστερά.

```
.wrap-1{
    width: 50%;
    max-width: 500px;
}
```

Αυτή η κλάση είναι ο πατέρας κόμβος του πρώτου πίνακα. Ορίζεται σταθερό μήκος ως ποσοστό τοις εκατό του άμεσου πατέρα κόμβου που είναι το <body> και μέγιστο μήκος.

```
.wrap-2{
    margin-left: 25%;
    width: 15%;
    max-width: 200px;
}
```

Ο πατέρας κόμβος του 2^{ου} πίνακα. Το εξωτερικό αριστερό του περιθώριο είναι 25% ως προς τον άμεσο πατέρα <body>. Επίσης έχει ποσοστιαίο δυναμικό μήκος και μέγιστο μήκος.

```
.wrap-3{
    margin-left: 25%;
    width: 40%;
    max-width: 400px;
}
```

Ο πατέρας κόμβος του 3^{ου} πίνακα. Έχουμε παρόμοιους κανόνες που συναντήσαμε στον πατέρα κόμβο του 2^{ου} πίνακα.

```
#filters{
    margin-bottom: 30px;
}
```

Ο κόμβος αυτός περιέχει όλα τα φίλτρα της σελίδας και έχει εξωτερικό κάτω περιθώριο.

```
.filter{
    margin-bottom: 20px;
}
```

Είναι ο πατέρας κάθε φίλτρου και έχει εξωτερικό κάτω περιθώριο.

```
#message{
    margin-top: 20px;
    opacity: 0;
}
```

Κρυφό μήνυμα που εμφανίζεται δυναμικά σε κάθε πάτημα του κουμπιού Προώθηση. Έχει διαφάνεια 0 και εξωτερικό άνω περιθώριο.

```
#total{
    color: #FF8C00;
}
```

Εμφανίζει το σύνολο των προωθημένων αναρτήσεων. Είναι ένας κόμβος που συναντάμε στη σελίδα posts.html. Η σελίδα αυτή περιέχει όλες τις προωθημένες αναρτήσεις και μοιράζεται το ίδιο αρχείο φύλλων στυλ με τη σελίδα index.html.

2.5 Ανάλυση index.js

Το Facebook δίνει ένα βασικό κομμάτι κώδικα (code snippet) για κάθε γλώσσα προγραμματισμού που υποστηρίζει να δημιουργούνται apps που διασυνδέεται με αυτό. Εμείς θα χρησιμοποιήσουμε τα παρακάτω κομμάτια κώδικα, για το λόγο ότι χρησιμοποιούμε JavaScript. Αυτά τα εισάγουμε στην αρχή του αρχείου index.js και είναι απαραίτητα για τη σάρωση των αναρτήσεων.

```
window.fbAsyncInit = function(){
  FB.init({
    appId      : '686497771493476',
    xfbml      : true,
    status     : true,
    cookie     : true,
    version    : 'v2.5'
  });
};
```

```
(function(d, s, id){
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/en_US/sdk.js";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));
```

Το πρώτο κομμάτι είναι μια συνάρτηση/μέθοδος (function) του Facebook, η οποία ορίζει και αρχικοποιεί κάποιες μεταβλητές μαζί με το App Id που είδαμε σε προηγούμενες ενότητες. Το δεύτερο κομμάτι είναι ακόμα μία συνάρτηση του Facebook και αυτό που κάνει ουσιαστικά είναι ή φόρτωση κώδικα βιβλιοθήκης JavaScript από το server του Facebook και η εισαγωγή του στον κόμβο <head> της σελίδας index.html, εκεί όπου ορίζονται μεταξύ άλλων αρχεία κώδικα.

```
var json = {}, posts = {};
```

Αρχικοποιούνται οι μεταβλητές που θα αποθηκεύσουν αντικείμενα, json και posts.

```
$(document).ready(function(){ });
```

Μέσα σε αυτό το block, δηλαδή ο κώδικας που περιέχεται σε αυτό εκτελείται υπό συνθήκες που ορίζονται έξω από το block, περιέχεται όλος ο υπόλοιπος κώδικας χρήστη. Αυτό το κάνουμε επειδή θέλουμε πρώτα να έχουν φορτώσει όλα τα στοιχεία/κόμβοι της σελίδας έτσι ώστε να είμαστε σε θέση να τα χειριστούμε. Σε άλλη περίπτωση, η JavaScript δεν αναγνωρίζει τους κόμβους της σελίδας με αποτέλεσμα να δημιουργείται σφάλμα εκτέλεσης κώδικα.

```
setTimeout(function(){ $('#scan-btn').removeAttr('disabled'); }, 1500);
```

Το παραπάνω κομμάτι κώδικα θα εκτελέσει τον ένθετο κώδικα σε 1.5 δευτερόλεπτα (1500 milliseconds). Συγκεκριμένα, κατά τη φόρτωση της σελίδας το κουμπί Σάρωση δεν είναι clickable. Μετά από 1.5 δευτερόλεπτα, το κουμπί καθίσταται clickable, διαγράφοντας την ιδιότητα disabled. Αυτό το κάνουμε διότι μερικές φορές ο κώδικας του Facebook, που είδαμε πιο πάνω, αργεί να φορτώσει, με αποτέλεσμα ο χρήστης να πατάει το κουμπί Σάρωση και να μη συμβαίνει τίποτα.

```
posts = getJSON();
if ( posts == null || posts == undefined ){
    posts = { filters:[], id:[], title:[], likes:[], comments:[], shares:[] };
}
```

```

var URL = window.location.href;
if ( URL.indexOf('posts') != -1 ){
    var data = "";
    for(var i = 0; i < posts.filters.length; i++){
        data += '<tr>';
        data += '<td class="aa">'+(i+1)+'</td>';
        data += '<td class="title">'+posts.title[i]+'</td>';
        data += '<td class="likes_cnt">'+posts.likes[i]+'</td>';
        data += '<td class="comments_cnt">'+posts.comments[i]+'</td>';
        data += '<td class="shares_cnt">'+posts.shares[i]+'</td>';
        data += '<td>'+posts.filters[i]+'</td>';
        data += '</tr>';
    }
    $('#posts tbody').append(data);
    $('#total').text(posts.filters.length);
}

```

Η μεταβλητή posts θα λάβει το αποτέλεσμα/επιστροφή της συνάρτησης getJSON, την οποία θα αναλύσουμε πιο κάτω. Έπειτα, γίνεται έλεγχος εάν το αποτέλεσμα είναι null ή undefined, δηλαδή η επιστροφή της συνάρτησης είναι κενή ή άκυρη. Σε αυτή την περίπτωση η μεταβλητή posts θα αρχικοποιηθεί με 6 ιδιότητες τύπου πίνακα.

Αυτές οι ιδιότητες περιέχουν στοιχεία αναρτήσεων: φίλτρο, id, τίτλο, αριθμό likes, αριθμό comments και αριθμό shares. Σε αντίθετη περίπτωση, η μεταβλητή posts είναι ήδη συμπληρωμένη και περιέχει υπάρχουσες αναρτήσεις.

Μετά, παίρνουμε το διεύθυνση (URL) της σελίδας στην οποία βρισκόμαστε. Εάν η διεύθυνση περιέχει το κείμενο posts, δηλαδή βρισκόμαστε στη σελίδα των προωθημένων αναρτήσεων, τότε ο κώδικας θα εκτελεστεί μόνο σε αυτή τη σελίδα. Θα αρχικοποιηθεί η μεταβλητή data που θα περιέχει τελικά μία συμβολοσειρά (string) με το HTML περιεχόμενο του πίνακα που περιέχει τις προωθημένες αναρτήσεις.

Το HTML κείμενο δημιουργείται δυναμικά και αποθηκεύεται στο τέλος (append) της προηγούμενης αποθηκευμένης συμβολοσειράς στη μεταβλητή data εντός ενός βρόγχου που θα εκτελεστεί συνολικά τόσες φορές όσο είναι το μήκος μιας (οποιασδήποτε) ιδιότητας του αντικειμένου/μεταβλητής posts. Δηλαδή εάν η ιδιότητα filters έχει μήκος 5, τότε ο βρόγχος for θα εκτελεστεί 5 φορές. Το αποτέλεσμα είναι το περιεχόμενο του πίνακα που περιέχει τις ήδη προωθημένες αναρτήσεις, δηλαδή ο πίνακας δημιουργείται δυναμικά στην μνήμη του browser.

Μετά το πέρας της τελευταίας εκτέλεσης του βρόγχου, το HTML περιεχόμενο εισάγεται στον κόμβο <tbody> που είναι παιδί του κόμβου με ιδιότητα id posts (η ιδιότητα id συμβολίζεται με τη δίεση: #). Η εισαγωγή γίνεται με την εντολή append, έχοντας επιλέξει το στοιχείο εισαγωγής (#posts tbody) και το HTML περιεχόμενο, που το έχει η μεταβλητή data.

Το στοιχείο που έχει το id total αποκτά κι αυτό δυναμικό περιεχόμενο. Εδώ αποκτά έναν αριθμό που εκφράζει το σύνολο των προωθημένων αναρτήσεων για πληροφόρηση του χρήστη. Ο αριθμός αυτός εξάγεται από το μήκος μιας ιδιότητας του αντικειμένου posts.

```
$(this).on('click', '#scan-btn', function(){
    var $that = $(this);
    $(this).attr('disabled', 'disabled');
    $(this).val('Φόρτωση αποτελεσμάτων...');

    FB.api('/enikosgr/feed?fields=name,description,comments,shares,likes,full_picture
,message,type&limit=100',
        function (response){
            $that.val('Σάρωση');
            $that.removeAttr('disabled');
            var data = "";
            if ( response && !response.error ){
                json = response;
                for(var i = 0; i < 100; i++){
```



```

data += '<tr>';
data += '<td class="aa">'+(i+1)+'</td>';
data += '<td class="title">'+response.data[i].name+'</td>';

if ( response.data[i].description != undefined ){
    data += '<td class="desc">'+response.data[i].description+'</td>';
} else{
    data += '<td class="desc"></td>';
}

data += '<td class="type">'+response.data[i].type+'</td>';

if ( response.data[i].full_picture != undefined ){
    data += '<td class="pic"></td>';
}

if ( response.data[i].likes != undefined ){
    data += '<td class="likes_cnt"><a href="#" data-
index="'+i+'">'+response.data[i].likes.data.length+'</a></td>';
} else{
    data += '<td class="likes_cnt">0</td>';
}

if ( response.data[i].comments != undefined ){
    data += '<td class="comments_cnt"><a href="#"
data-index="'+i+'">'+response.data[i].comments.data.length+'</a></td>';
}

```

```

        } else{
            data += '<td class="comments_cnt">0</td>';
        }

        if ( response.data[i].shares != undefined ){
            data += '<td class="shares_cnt">'+response.data[i].shares.count+'</td>';
        } else{
            data += '<td class="shares_cnt">0</td>';
        }

        data += '</tr>';
    }
    $('#scan-results tbody').append(data);
    $('#div.wrap-1').fadeIn(500);
} else{
    alert('Συνέβη κάποιο σφάλμα. Προσπαθήστε αργότερα.');
```

Το παραπάνω block κώδικα θα εκτελεστεί σε ένα συμβάν (event) click, δηλαδή ο χρήστης θα πατήσει ένα κουμπί στη σελίδα. Εδώ ο χρήστης θα πατήσει το κουμπί Σάρωση. Ο κώδικας αυτός αναμένει για ένα συμβάν click για να εκτελεστεί.

Το κουμπί Σάρωση έχει την ιδιότητα id scan-btn. Αποθηκεύουμε στη μεταβλητή \$that το αντικείμενο που καλείται, το κουμπί σάρωσης. Ορίζουμε μετά το κουμπί να μην είναι clickable και να λάβει τιμή-κείμενο: Φόρτωση αποτελεσμάτων..., για να πληροφορήσουμε το χρήστη ότι έχοντας πατήσει το κουμπί πρέπει να περιμένει κάποια δευτερόλεπτα για να φορτώσουν τα αποτελέσματα της σάρωσης, δηλαδή οι αναρτήσεις από το Facebook Group.

Υπάρχει συγκεκριμένος τρόπος ανάγνωσης των αναρτήσεων από ένα group και συμβολίζεται με τη μορφή URL με παραμέτρους. Συνολικά, οι παράμετροι είναι πάνω 50, εμείς έχουμε δώσει μόνο κάποιες από αυτές.

```
/enikosgr/feed?fields=name,description,comments,shares,likes,full_picture,message,type&limit=100
```

1. Θέτουμε το όνομα του group θέλουμε να σαρώσουμε: enikosgr
2. Ορίζουμε τι θέλουμε να κάνουμε το group, εμείς θέλουμε τις αναρτήσεις: feed
3. Θέλουμε για κάθε ανάρτηση να γνωρίζουμε (εάν υπάρχουν) τον τίτλο (name), την προεπισκόπηση (description), τα σχόλια (comments), τα shares, τα likes, την εικόνα που επισυνάφθηκε (full_picture), το μήνυμα κάθε σχολίου (message), τον τύπο (type) κάθε ανάρτησης
4. Το όριο, δηλαδή το σύνολο των αναρτήσεων που θα επιστραφεί, το μέγιστο είναι 100: limit

Το αποτέλεσμα της εντολής `FB.api` είναι ένα αντικείμενο JSON, το οποίο θα το μετατρέψουμε για να το εμφανίσουμε στη σελίδα `index.html`. Επαναφέρουμε το κουπί Σάρωση στην αρχική του κατάσταση και μετά γίνεται έλεγχος εάν το αποτέλεσμα είναι έγκυρο και δεν περιέχει λάθη. Διαφορετικά, θα εμφανιστεί μήνυμα λάθος στο χρήστη με την εντολή `alert('Συνέβη κάποιο σφάλμα. Προσπαθήστε αργότερα.')`.

Το αποτέλεσμα αποθηκεύεται στη μεταβλητή `json` για χρήση αργότερα. Ακολουθεί ένα βρόγχος `for` που θα εκτελεστεί 100 φορές, δηλαδή τόσες όσο είναι το μήκος του αντικειμένου που επιστέφει το Facebook. Η δουλειά του βρόγχου είναι να δημιουργήσει δυναμικά ένα HTML κείμενο και συγκεκριμένα το κείμενο με τις 100 πιο πρόσφατες αναρτήσεις. Θα δημιουργηθούν 100 γραμμές με κάθε γραμμή να περιέχει 8 στήλες:

1. Αύξων αριθμός: `(i+1)`, η τρέχουσα εκτέλεση του βρόγχου (μεταβλητή `i`) αυξημένη κατά 1 για να εμφανίζεται η πρώτη ανάρτηση ξεκινώντας από τον αριθμό 1.
2. Τίτλος: `response.data[i].name`
3. Προεπισκόπηση: `response.data[i].description`, γίνεται πρώτα έλεγχος εάν η ανάρτηση έχει περιγραφή/προεπισκόπηση (`response.data[i].description != undefined`)

4. Τύπος: `response.data[i].type`
5. Εικόνα: `response.data[i].full_picture`, γίνεται πρώτα έλεγχος εάν η ανάρτηση περιέχει εικόνα (`response.data[i].full_picture != undefined`)
6. Likes: `response.data[i].likes.data.length`, γίνεται πρώτα έλεγχος εάν η ανάρτηση έχει likes (`response.data[i].likes != undefined`)
7. Comments: `response.data[i].comments.data.length`, γίνεται πρώτα έλεγχος εάν η ανάρτηση έχει comments (`response.data[i].comments != undefined`)
8. Shares: `response.data[i].shares.count`, γίνεται πρώτα έλεγχος εάν η ανάρτηση έχει διαμοιραστεί (`response.data[i].shares != undefined`)

Το τελικό διαμορφωμένο HTML κείμενο εισάγεται στον πρώτο πίνακα της σελίδας και συγκεκριμένα στον κόμβο `<tbody>` (`#scan-results tbody`). Ο πίνακας αρχικά είναι κρυφός και εμφανίζεται σταδιακά μέσα σε μισό δευτερόλεπτο (500 milliseconds) με τη μορφή εφέ `fade in`, για να δείξει την ομαλή μετάβαση από κενό περιεχόμενο πίνακα σε γεμάτο περιεχόμενο πίνακα.

```
$(this).on('click', 'td.likes_cnt a', function(){
    var cnt = parseInt($(this).text());
    var index = parseInt($(this).attr('data-index'));
    var data = "";
    $('#like-results tbody').html("");
    for(var i = 0; i < cnt; i++){
        data += '<tr>';
        data += '<td class="aa">'+(i+1)+'</td>';
        data += '<td class="user">'+json.data[index].likes.data[i].name+'</td>';
        data += '</tr>';
    }
    $('#like-results tbody').append(data);
    $('div.wrap-3').fadeOut(100);
    $('div.wrap-2').fadeIn(500);
});
```

```
});
```

Παραπάνω είναι ένα συμβάν για αναμονή click στο πάτημα ενός συνδέσμου (link) με πατέρα το κελί που έχει την κλάση likes (η κλάση συμβολίζεται με την τελεία: .) που βρίσκεται στον πίνακα των αναρτήσεων.

Ο σύνδεσμος χαρακτηρίζει τον αριθμό των likes και είναι clickable. Αρχικά, θα αποθηκευτεί σε μια μεταβλητή ο αριθμός των likes της ανάρτησης που περιέχεται στο περιεχόμενο του κελιού. Θα αποθηκευτεί σε μια άλλη μεταβλητή η σειρά της ανάρτησης αναφορικά με το JSON αντικείμενο που επιστρέφεται από το Facebook για επαλήθευση ότι τα likes συσχετίζονται με τη ανάρτηση που επέλεξε ο χρήστης για την ανάλυση των likes. Η μεταβλητή data θα λάβει HTML κείμενο και θα εισαχθεί στον κρυφό πίνακα των likes.

Ο βρόγχος που διαμορφώνει το κείμενο θα εκτελεστεί τόσες όσες φορές όσα είναι τα likes της ανάρτησης. Θα χρησιμοποιηθεί η μεταβλητή json, η οποία έχει αποθηκεύσει τις αναρτήσεις προηγούμενα όπως είδαμε. Τα likes αναλύονται έχοντας τις στήλες: αύξων αριθμός και το όνομα του χρήστη που έκανε το like. Θα διαγράψουμε το όποιο περιεχόμενο έχει ο πίνακας των likes, για το λόγο ότι ο χρήστης έχει επιλέξει να δει τα likes μιας άλλης ανάρτησης πρωτίτερα, οπότε δε θέλουμε να δούμε ταυτόχρονα την ανάλυση των likes για 2 ή παραπάνω αναρτήσεις.

Μετά το πέρας του βρόγχου for, θα εισαχθεί το HTML string στον κόμβο <tbody> στον πίνακα των likes και θα εμφανιστεί τελικά ο πίνακας με το εφέ fade in.

```
$(this).on('click', 'td.comments_cnt a', function(){
    var cnt = parseInt($(this).text());
    var index = parseInt($(this).attr('data-index'));
    var data = "";
    $('#comment-results tbody').html("");
    for(var i = 0; i < cnt; i++){
        data += '<tr>';
        data += '<td class="aa">'+(i+1)+'</td>';
        data += '
            +='
            '
            <td
            class="user">'+json.data[index].comments.data[i].from.name+'</td>';
```

```

        data += '<td
class="message">'+json.data[index].comments.data[i].message+'</td>';
        data += '</tr>';
    }
    $('#comment-results tbody').append(data);
    $('#div.wrap-2').fadeOut(100);
    $('#div.wrap-3').fadeIn(500);
});

```

Ο παραπάνω κώδικας αναμένει για click στον αριθμό των comments κάθε ανάρτησης για να εκτελεστεί. Έχει παρόμοια λογική με τον κώδικα της ανάλυσης των likes, αλλά αυτό που αλλάζει είναι ο πίνακας επιλογής και εισαγωγής (πίνακας των comments), η ανάγνωση των comments από τη μεταβλητή json και το γεγονός ότι εκτός από τον αύξοντα αριθμό και το όνομα χρήστη από τον οποίο έγινε το comment, έχουμε επιπλέον και το περιεχόμενο του comment.

```

$(this).on('click', '#filter-btn', function(){
    if ( $.isEmptyObject(json) ){
        alert('Πατήστε πρώτα τη Σάρωση για να φιλτράρετε αναρτήσεις!');
    } else{
        var title = $('#filter-title').val();
        var likes = parseInt($('#filter-like').val());
        var comments = parseInt($('#filter-comment').val());
        var shares = parseInt($('#filter-share').val());
        var like_user = $('#filter-like-name').val();
        var comment_user = $('#filter-comment-name').val();
        var comment_message = $('#filter-comment-message').val();

        for(var i = 0; i < 100; i++){
            if ( title != "" &&
json.data[i].name.toLowerCase().indexOf(title.toLowerCase()) != -1 ){

```

```

        setPosts(posts, i, 1);
    }

    if ( likes != " " && json.data[i].likes != undefined && likes <
parseInt(json.data[i].likes.data.length) ){
        setPosts(posts, i, 2);
    }

    if ( comments != " " && json.data[i].comments != undefined
&& comments < parseInt(json.data[i].comments.data.length) ){
        setPosts(posts, i, 3);
    }

    if ( shares != " " && json.data[i].shares != undefined && shares
< parseInt(json.data[i].shares.count) ){
        setPosts(posts, i, 4);
    }

    if ( json.data[i].likes != undefined && like_user != " " ){
        for(var j = 0; j < json.data[i].likes.data.length; j++){
            if ( json.data[i].likes.data[j].name.toLowerCase() == like_user.toLowerCase() ){
                setPosts(posts, i, 5);
            }
        }
    }

    if ( json.data[i].comments != undefined && comment_user !=
" " ){

```

```

        for(var j = 0; j < json.data[i].comments.data.length;
j++){
            if (
json.data[i].comments.data[j].from.name.toLowerCase() == comment_user.toLowerCase()
){
                setPosts(posts, i, 6);
            }
        }
    }

    if ( json.data[i].comments != undefined &&
comment_message != ""){
        for(var j = 0; j < json.data[i].comments.data.length;
j++){
            if (
json.data[i].comments.data[j].message.toLowerCase().indexOf(comment_message.toLower
rCase()) != -1 ){
                setPosts(posts, i, 7);
            }
        }
    }

    $.ajax({
        type: 'POST',
        dataType: 'json',
        url: 'posts.php',
        data: {'data': posts},
    });

```



```
$('#message').stop().animate({ opacity: 1 },  
500).delay(3000).animate({ opacity: 0 }, 500);  
    }  
});
```

Εδώ έχουμε ένα click event όταν ο χρήστης πατήσει το κουμπί Προώθηση. Προϋποθέτει ότι ο χρήστης έχει χρησιμοποιήσει τουλάχιστον ένα από τα επτά φίλτρα. Σε αντίθετη περίπτωση, δε γίνεται προώθηση καμίας ανάρτησης.

Επιπλέον, γίνεται έλεγχος εάν ο χρήστης έχει επιλέξει να σαρώσει τις αναρτήσεις από το Facebook Group, ελέγχοντας εάν η μεταβλητή json έχει αποθηκεύσει περιεχόμενο, δηλαδή η μεταβλητή είναι κενή (\$.isEmptyObject(json)). Εάν δεν έχει περιεχόμενο, τότε το επόμενο μήνυμα εμφανίζεται στο χρήστη: Πατήστε πρώτα τη Σάρωση για να φιλτράρετε αναρτήσεις. Αρχικά, θα αποθηκεύσουμε την τιμή του κάθε φίλτρου σε αντίστοιχες μεταβλητές. Η τιμή, δηλαδή το περιεχόμενο του κόμβου <input> και του κόμβου <select> χρησιμοποιώντας την εντολή val() επί του στοιχείου επιλογής. Στην περίπτωση που αναμένουμε μία τιμή να είναι αριθμός, δηλαδή να προέρχεται από κόμβο <select>, μετατρέπουμε (type cast) την τιμή σε αριθμό, διότι ως προεπιλογή μία τιμή στην JavaScript χρησιμοποιώντας την εντολή val() αντιπροσωπεύεται ως συμβολοσειρά.

Στη συνέχεια, θα εκτελέσουμε ένα βρόγχο 100 φορές, οπότε θα ψάξει για εφαρμοζόμενα φίλτρα για κάθε ανάρτηση. Πρώτα, ελέγχουμε για κάθε φίλτρο εάν αυτό έχει οριστεί και δεν είναι κενό. Επομένως, έχουμε τους εξής επιπλέον ελέγχους για τα παρακάτω φίλτρα αναφορικά με κάθε ανάρτηση:

1. Μέρος του τίτλου της ανάρτησης υπάρχει στο κείμενο που έχει εισάγει ο χρήστης χρησιμοποιώντας το 1^ο φίλτρο. Η τιμή του φίλτρου και το κείμενο που αναζητείται μετατρέπονται σε πεζούς χαρακτήρες, καθώς στο προγραμματισμό ένα πεζό γράμμα x είναι διαφορετικό από το κεφαλαίο γράμμα X.
2. Έλεγχος εάν η τιμή του φίλτρου των likes είναι μικρότερη από το αριθμό των likes της ανάρτησης. Επίσης, ελέγχεται εάν όντως υπάρχουν likes για την τρέχουσα ανάρτηση του βρόγχου.

3. Έλεγχος εάν η τιμή του φίλτρου των comments είναι μικρότερη από το αριθμό των comments της ανάρτησης. Επίσης, ελέγχεται εάν όντως υπάρχουν comments για την τρέχουσα ανάρτηση του βρόγχου.
4. Έλεγχος εάν η τιμή του φίλτρου των shares είναι μικρότερη από το αριθμό των shares της ανάρτησης. Επίσης, ελέγχεται εάν όντως υπάρχουν shares για την τρέχουσα ανάρτηση του βρόγχου.
5. Εάν ο χρήστης έχει εισάγει τιμή στο 5^ο φίλτρο, ελέγχεται εάν όντως υπάρχουν likes για την τρέχουσα ανάρτηση του βρόγχου και θα εκτελεστεί ένας ένθετος βρόγχος τόσες φορές όσα είναι και τα likes της ανάρτησης. Εντός του ένθετου βρόγχου, θα γίνει έλεγχος εάν το όνομα του χρήστη που έκανε το like είναι ίδιο με την τιμή του φίλτρου που εισήγαγε ο χρήστης. Φυσικά, θα μετατραπούν και οι δύο συμβολοσειρές σε πεζά, όπως εξηγήθηκε στο 1^ο φίλτρο.
6. Εάν ο χρήστης έχει εισάγει τιμή στο 6^ο φίλτρο, ελέγχεται εάν όντως υπάρχουν comments για την τρέχουσα ανάρτηση του βρόγχου και θα εκτελεστεί ένας ένθετος βρόγχος τόσες φορές όσα είναι και τα comments της ανάρτησης. Εντός του ένθετου βρόγχου, θα γίνει έλεγχος εάν το όνομα του χρήστη που έκανε το comment είναι ίδιο με την τιμή του φίλτρου που εισήγαγε ο χρήστης. Φυσικά, θα μετατραπούν και οι δύο συμβολοσειρές σε πεζά, όπως εξηγήθηκε στο 1^ο φίλτρο.
7. Εάν ο χρήστης έχει εισάγει τιμή στο 7^ο φίλτρο, ελέγχεται εάν όντως υπάρχουν comments για την τρέχουσα ανάρτηση του βρόγχου και θα εκτελεστεί ένας ένθετος βρόγχος τόσες φορές όσα είναι και τα comments της ανάρτησης. Εντός του ένθετου βρόγχου, θα γίνει έλεγχος εάν το κείμενο κάθε comment περιέχεται σε μέρος της τιμής του φίλτρου που εισήγαγε ο χρήστης. Φυσικά, θα μετατραπούν και οι δύο συμβολοσειρές σε πεζά, όπως εξηγήθηκε στο 1^ο φίλτρο.

Εάν ένας ή περισσότεροι έλεγχοι είναι αληθείς, θα εκτελεστεί η μέθοδος setPosts δίνοντας ως παραμέτρους το αντικείμενο posts, την τρέχουσα θέση της ανάρτησης και τον αριθμό του αντίστοιχου φίλτρου. Αυτή η μέθοδος θα εξηγηθεί στη συνέχεια.

Ακολουθεί ένα αίτημα (request) στον server που θα διεκπεραιωθεί από την PHP. Πρόκειται για έναν τρόπο επικοινωνίας μεταξύ διαφορετικών τεχνολογιών, που καλείται Ajax (Asynchronous JavaScript XML).

Ο σκοπός του αιτήματος είναι η αποθήκευση των προωθημένων αναρτήσεων με βάση τα φίλτρα που επέλεξε ο χρήστης σε ένα αρχείου τύπου JSON για ανάγνωση είτε από τη σελίδα index.html (πριν γίνει το αίτημα, το λεγόμενο αρχείο διαβάζεται ως προς το περιεχόμενό του και αυτό αποθηκεύεται στη μεταβλητή posts, έτσι ώστε να κρατηθούν οι παλαιότερες προωθημένες αναρτήσεις) είτε από τη σελίδα posts.html, όπου εμφανίζονται όλες οι αποθηκευμένες προωθημένες αναρτήσεις.

Επομένως, θα οριστούν οι εξής οδηγίες για το αίτημα:

1. Ο τύπος/μέθοδος του αιτήματος είναι POST. Θα μπορούσε να είναι GET, αλλά επιλέγουμε τον πρώτο τύπο, καθώς επιτρέπει περισσότερα bytes και άρα μεγαλύτερο μήκος περιεχομένου να σταλεί στην PHP.
2. Ο τύπος των δεδομένων είναι JSON.
3. Το URL, δηλαδή η διεύθυνση του αρχείου PHP είναι το posts.php.
4. Τα δεδομένα που αποστέλλονται στο server, εδώ οι προωθημένες αναρτήσεις που βρίσκονται αποθηκευμένες στην μεταβλητή posts.

Τελικά, θα εκτελεστεί ένα animation, το οποίο θα δείξει για 3 δευτερόλεπτα το προηγούμενο κρυφό μήνυμα: Επιτυχής προώθηση και έπειτα θα εξαφανιστεί. Αυτό επιτυγχάνεται ελέγχοντας τη διαφάνεια του κόμβου με ιδιότητα message, δηλαδή για την εμφάνιση του μηνύματος το opacity θα γίνει σταδιακά ίσο με 1, ενώ για την εξαφάνισή του θα γίνει σταδιακά ίσο με 0.

```
function getJSON(){
    var obj = null;
    $.ajax({
        type: 'POST',
        dataType: 'json',
        async: false,
```

```

        url: 'posts.json',
        complete: function(response){
            obj = $.parseJSON(response.responseText);
        }
    });
    return obj;
}

```

Η συνάρτηση αυτή θα διαβάσει το τοπικό αρχείο posts.json κατά τη φόρτωση της σελίδας. Το αρχείο αυτό περιέχει προηγούμενες προωθημένες αναρτήσεις. Αρχικοποιούμε μία μεταβλητή με όνομα obj με τιμή null (κενή). Στη συνέχεια θα εκτελεστεί ένα αίτημα προς το server με σκοπό ο τελευταίος να επιτρέψει την ανάγνωση του τοπικού αρχείου δίνοντας τις εξής παραμέτρους στο αίτημα:

1. Τύπος/μέθοδος: POST
 2. Τύπος δεδομένων: JSON
 3. Σύγχρονο αίτημα και όχι ασύγχρονο όπως είδαμε στο προηγούμενο αίτημα.
 4. Η διαδρομή του αρχείου για το οποίο γίνεται αίτηση: posts.json.
 5. Μέθοδος που θα εκτελεστεί μετά την ολοκλήρωση του αιτήματος (callback function) με όνομα: complete. Εδώ αποθηκεύεται το περιεχόμενο του αρχείου, αφού πρώτα μετατραπεί σε αντικείμενο τύπου JSON (\$.parseJSON), καθώς ως προεπιλογή ο server επιστρέφει συμβολοσειρά σε μορφή JSON, αλλά όχι αντικείμενο.
- Η συνάρτηση επιστρέφει το περιεχόμενο της μεταβλητής obj στον καλούντα.

```

function setPosts(posts, index, filters){
    var id = json.data[index].id;
    if ( $.inArray(id, posts.id) == -1 ){
        posts.id.push(id);
        posts.title.push(json.data[index].name);
        posts.filters.push(filters);
        if ( json.data[index].likes != undefined ){

```

```

        posts.likes.push(json.data[index].likes.data.length);
    } else{
        posts.likes.push(0);
    }
    if ( json.data[index].comments != undefined ){
        posts.comments.push(json.data[index].comments.data.length);
    } else{
        posts.comments.push(0);
    }
    if ( json.data[index].shares != undefined ){
        posts.shares.push(json.data[index].shares.count);
    } else{
        posts.shares.push(0);
    }
    } else{
        var key = $.inArray(id, posts.id);
        posts.title[key] = json.data[index].name;
        posts.filters[key] = filters;
        if ( json.data[index].likes != undefined ){
            posts.likes[key] = json.data[index].likes.data.length;
        }
        if ( json.data[index].comments != undefined ){
            posts.comments[key]
            json.data[index].comments.data.length;
        }
        if ( json.data[index].shares != undefined ){
            posts.shares[key] = json.data[index].shares.count;
        }
    }
}

```

Η παραπάνω μέθοδος καλείται όπως είδαμε σε προηγούμενη ενότητα, κατά το φιλτράρισμα και την προώθηση των αναρτήσεων. Ορίζεται, αρχικά, μια μεταβλητή `id`, η οποία έχει δυναμικό περιεχόμενο, δηλαδή αποθηκεύει το `id` κάθε ανάρτησης που εξάγεται από τις παραμέτρους της συνάρτησης. Το `id` της ανάρτησης χρησιμοποιείται για να διαπιστωθεί εάν μια προωθημένη ανάρτηση έχει ήδη αποθηκευτεί, έτσι ώστε να μην υπάρχουν διπλότυπες αναρτήσεις.

Εάν μια ανάρτηση φιλτραριστεί για προώθηση, τότε το περιεχόμενο και τα στοιχεία της ανάρτησης γράφονται πάνω (`overwrite`) στην παλαιά ανάρτηση, αφού πρώτα αναζητηθεί (`$.inArray(id, posts.id)`) εάν το τρέχον `id` υπάρχει στη μεταβλητή `posts`, η οποία έχει αποθηκευμένες όλες τις προωθημένες αναρτήσεις.

Αυτό το κάνουμε επειδή υπάρχουν περιπτώσεις που ο χρήστης έχει ήδη προωθήσει μια ανάρτηση στο παρελθόν και επιλέξει να επαναπροωθήσει την ίδια ανάρτηση για την οποία έχει μεταβληθεί είτε ο αριθμός των likes είτε ο αριθμός των comments είτε ο αριθμός των shares είτε ο τίτλος της ανάρτησης. Επίσης, αποθηκεύουμε και το νέο φίλτρο το οποίο χρησιμοποιήσαμε για την προώθηση της ανάρτησης.

Σε περίπτωση που η ανάρτηση είναι μοναδική, δηλαδή δεν έχει αποθηκευτεί στο παρελθόν, αποθηκεύουμε στη μεταβλητή `posts` τις παρακάτω ιδιότητες (τύπου πίνακα) αυτού του αντικειμένου:

1. Το `id`
2. Τον τίτλο
3. Το φίλτρο το οποίο χρησιμοποιήθηκε
4. Τον αριθμό των likes
5. Τον αριθμό των comments
6. Τον αριθμό των shares

Υπάρχουν περιπτώσεις όπου κάποιες αναρτήσεις δεν έχουν likes, comments, shares. Γι' αυτό ελέγχουμε εάν για την υπό προώθηση ανάρτηση υπάρχουν αντίστοιχα likes, comments, shares. Σε αντίθετη περίπτωση, αποθηκεύουμε τον αριθμό 0 στη μεταβλητή

posts. Η μεταβλητή `index` που λαμβάνεται από τις παραμέτρους της συνάρτησης θα καθορίσει τη θέση στο αντικείμενο/πίνακα `json`.

2.6 Ανάλυση posts.php

```
<?php  
file_put_contents('posts.json', json_encode($_POST['data']));
```

Είναι η μοναδική εντολή του αρχείου PHP που βρίσκεται στο server και για το οποίο γίνεται αίτημα όπως περιγράφηκε παραπάνω. Ένα αρχείο PHP πρέπει να ξεκινά με την ετικέτα `<?php`. Έπειτα, ακολουθούν οι PHP εντολές. Η παραπάνω γραμμή είναι ουσιαστικά 2 εντολές. Οι εντολές εκτελούνται με προτεραιότητα ξεκινώντας από δεξιά ως προς τη σειρά ανάγνωσης.

Η εντολή `json_encode` μετατρέπει μια συμβολοσειρά σε αντικείμενο JSON. Η συμβολοσειρά είναι όρισμα της προηγούμενης εντολής και βρίσκεται αποθηκευμένη στη μεταβλητή `$_POST['data']`, η οποία έχει λάβει τις παραμέτρους του αιτήματος μέσω μεθόδου AJAX, όπως είδαμε σε προηγούμενες ενότητες. Στην περίπτωσή μας έχουμε αποστείλει στον server μόνο μία παράμετρο, με όνομα `data`. Αυτή η παράμετρος περιέχει τη συμβολοσειρά με περιεχόμενο τις προωθημένες αναρτήσεις. Η αριστερότερη εντολή είναι η `file_put_contents` και δέχεται 2 ορίσματα: το όνομα του αρχείου που θα αποθηκεύσει στο server και τα δεδομένα του αρχείου προς αποθήκευση.

Η συγκεκριμένη εντολή θα αποθηκεύσει στο αρχείο `posts.json` τις προωθημένες αναρτήσεις για χρήση από τις 2 HTML σελίδες της εφαρμογής.

2.7. Ανάλυση posts.html

```
<html>  
<head>  
<title>Facebook: Επίμαχες αναρτήσεις</title>  
<meta charset="UTF-8">
```

```

<link
href='https://fonts.googleapis.com/css?family=Ubuntu:400,300,300italic,400italic,500,500
italic,700&subset=greek,latin' rel='stylesheet' type='text/css'>
<link rel="stylesheet" type="text/css" href="style.css">
<script src="jquery.js"></script>
<script src="index.js"></script>
</head>
<body>
<div align="center">
  <h3>Επίμαχες Αναρτήσεις από Facebook Group: Enikos.gr</h3>
  <h4>Total: <span id="total"></span></h4>
  <br/>
  <table id="posts">
    <thead>
      <tr>
        <th class="aa">A/A</th>
        <th class="title">Τίτλος</th>
        <th class="likes_cnt">Likes</th>
        <th class="comments_cnt">Σχόλια</th>
        <th class="shares_cnt">Shares</th>
        <th>Εφαρμοζόμενα φίλτρα</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>
</div>
</body>
</html>

```


Το αρχείο αυτό είναι η σελίδα που εμφανίζει τις προωθημένες αναρτήσεις από το Facebook Group. Οι περισσότεροι κόμβοι έχουν το ίδιο περιεχόμενο που είδαμε στη σελίδα index.html. Ορίζεται οι βασικοί κόμβοι <html>, <body>, <meta>, <head>, <script>, <link>. Διαφέρει ο κόμβος <title>, όπου ορίζεται ο τίτλος της ιστοσελίδας: Facebook: Επίμαχες αναρτήσεις.

Ορίζεται, επίσης, ο υπότιτλος της σελίδας: Επίμαχες Αναρτήσεις από Facebook Group: Epikos.gr εντός του κόμβου <h3>. Ο κόμβος <h4> περιέχει το σύνολο των ήδη προωθημένων αναρτήσεων. Το κυρίως περιεχόμενο της σελίδας είναι ο πίνακας <table> με id: posts, ο οποίος έχει τις παρακάτω στήλες:

1. Αύξων αριθμός
2. Τίτλος
3. Likes
4. Σχόλια
5. Εφαρμοζόμενα φίλτρα

Το περιεχόμενο του πίνακα καθορίζεται δυναμικά, όπως είδαμε μέσω της JavaScript στο αρχείο index.js και συγκεκριμένα από τα δεδομένου του αρχείο posts.json.

3. Στιγμιότυπα από τη λειτουργία χρήσης

Πρώθηση Επίμαχων Αναρτήσεων από Facebook Group: Enikos.gr

Σάρωση

Φίλτρα

1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:
2. Η ανάρτηση έχει πάνω από: likes
3. Η ανάρτηση έχει πάνω από: comments
4. Η ανάρτηση έχει πάνω από: shares
5. Το like έγινε από:
6. Το comment έγινε από:
7. Το comment περιέχει το κείμενο:

Πρώθηση

Στιγμιότυπο 1: Πρώτη φόρτωση της σελίδας index.html. Ο χρήστης βλέπει τα φίλτρα για την προώθηση των αναρτήσεων, αλλά δε βλέπει μέχρι στιγμής της αναρτήσεις από το Facebook Group.

Σάρωση

Φίλτρα

1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:

2. Η ανάρτηση έχει πάνω από: likes

3. Η ανάρτηση έχει πάνω από: comments

4. Η ανάρτηση έχει πάνω από: shares

5. Το like έγινε από:

6. Το comment έγινε από:

7. Το comment περιέχει το κείμενο:

Προώθηση

Αποτελέσματα σάρωσης

| A/A | Τίτλος | Προεπισκόπηση | Τύπος | Εικόνα | Likes | Σχόλια | Shares |
|-----|--|---|-------|--------|-------|--------|--------|
| 1 | Κυριακή στη Realnews: Η εκδίκηση της γυφτιάς(CD), Παραμύθια της Μικράς Ασίας, Taste & 10x5000€ | Η ΕΚΔΙΚΗΣΗ ΤΗΣ ΓΥΦΤΙΑΣ Ο Μανώλης Ρασούλης ο Νίκος Ξυδάκης και ο Νίκος Παπάζογλου σε ένα από τα σημαντικότερα άλμπουμ της ελληνικής δισκογραφίας, ΤΑ ΛΑΪΚΑ ΠΑΡΑΜΥΘΙΑ ΤΗΣ ΜΙΚΡΑΣ ΑΣΙΑΣ Για πρώτη φορά κορυφαίοι λαογράφοι καταγράφουν σε ένα συλλεκτικό βιβλίο τα παραμύθια από τις ... | link | | 0 | 0 | 0 |
| 2 | Μπακογιάννη: Δεν προκύπτει ότι το ΔΝΤ θέλει να... | «Η κυβέρνηση δια του κ. Σκουρλέτη λέει ότι στη συνομιλία Τόμσεν – Βελκουλέσκου θέλουν να πάνε την Ελλάδα σε χρεοκοπία, ενάντια στο δικό μου... | link | | 4 | 1 | 1 |


Στιγμιότυπο 2: Ο χρήστης μόλις πάτησε το κουμπί Σάρωση και σε δευτερόλεπτα εμφανίστηκαν οι αναρτήσεις από το Facebook Group.

5. Το like έγινε από:

6. Το comment έγινε από:

7. Το comment περιέχει το κείμενο:

Αποτελέσματα σάρωσης

| A/A | Τίτλος | Προεπισκόπηση | Τύπος | Εικόνα | Likes | Σχόλια | Shares |
|-----|--|---|-------|---|-------|--------|--------|
| 1 | Κυριακή στη Realnews: Η εκδίκηση της γυφτιάς(CD), Παραμύθια της Μικράς Ασίας, Taste & 10x5000€ | Η ΕΚΔΙΚΗΣΗ ΤΗΣ ΓΥΦΤΙΑΣ Ο Μανώλης Ρασούλης ο Νίκος Ξυδάκης και ο Νίκος Παπάζογλου σε ένα από τα σημαντικότερα άλμπουμ της ελληνικής δισκογραφίας, ΤΑ ΛΑΪΚΑ ΠΑΡΑΜΥΘΙΑ ΤΗΣ ΜΙΚΡΑΣ ΑΣΙΑΣ Για πρώτη φορά κορυφαίοι λαογράφοι καταγράφουν σε ένα συλλεκτικό βιβλίο τα παραμύθια από τις ... | link |  | 0 | 0 | 0 |
| 2 | Μπακογιάννη: Δεν προκύπτει ότι το ΔΝΤ θέλει χρεοκοπία της Ελλάδας - BINTEO | «Η κυβέρνηση δια του κ. Σκουρλέτη λέει ότι στη συνομιλία Τόμσεν – Βελκουλέσκου θέλουν να πάνε την Ελλάδα σε χρεοκοπία, εμένα στη δικιά μου ανάγνωση αυτό δεν μου προκύπτει. Προκύπτει η ανάλυση ότι οι Έλληνες, όταν φτάνουν στο παρά πέντε ... | link |  | 4 | 1 | 1 |
| 3 | Σκουρλέτης: Το ΔΝΤ σχεδίαζε τη χρεοκοπία της χώρας - BINTEO | «Το πρώτο που πρέπει κανείς να επισημάνει είναι ότι δεν έχει διαψευστεί το περιεχόμενο» τόνισε ο υπουργός Περιβάλλοντος και Ενέργειας, Πάνος Σκουρλέτης μιλώντας στην εκπομπή του Νίκου Χατζηνικολάου, «στον ενικό» αναφορικά με τις αποκαλύψεις του Wikileaks. «Αρα μιλάμε για μία ... | link |  | 5 | 22 | 2 |

Ανάλυση Comments

| A/A | Χρήστης | Σχόλιο |
|-----|---------------------|---|
| 1 | Theo-Jess Ligs | ΤΑ ΕΜΒΟΛΙΑ???? |
| 2 | Μαίρη Λεβεντοπούλου | Πολύ μεγάλο είναι ο κυριότερος παράγοντας |

Στιγμιότυπο 3: Ο χρήστης πάτησε σε ένα σύνδεσμο των comments μιας ανάρτησης για να δει αναλυτικά και να διαβάσει τα comments γι' αυτήν.

1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:

2. Η ανάρτηση έχει πάνω από: likes

3. Η ανάρτηση έχει πάνω από: comments

4. Η ανάρτηση έχει πάνω από: shares



5. Το like έγινε από:

6. Το comment έγινε από:

7. Το comment περιέχει το κείμενο:

Πρώθηση

Αποτελέσματα σάρωσης

| A/A | Τίτλος | Προεπισκόπηση | Τύπος | Εικόνα | Likes | Σχόλια | Shares |
|-----|--|---|-------|---|-------|--------|--------|
| 1 | Κυριακή στη Realnews: Η εκδίκηση της γυφτιάς(CD), Παραμύθια της Μικράς Ασίας, Taste & 10x5000€ | Η ΕΚΔΙΚΗΣΗ ΤΗΣ ΓΥΦΤΙΑΣ Ο Μανώλης Ρασούλης ο Νίκος Ξυδάκης και ο Νίκος Παπάζογλου σε ένα από τα σημαντικότερα άλμπουμ της ελληνικής δισκογραφίας. ΤΑ ΛΑΪΚΑ ΠΑΡΑΜΥΘΙΑ ΤΗΣ ΜΙΚΡΑΣ ΑΣΙΑΣ Για πρώτη φορά κορυφαίοι λαογράφοι καταγράφουν σε ένα συλλεκτικό βιβλίο τα παραμύθια από τις ... | link |  | 0 | 0 | 0 |
| 2 | Μπακογιάννη: Δεν προκύπτει ότι το ΔΝΤ θέλει χρεοκοπία της Ελλάδας - BINTEO | «Η κυβέρνηση δια του κ. Σκουρλέτη λέει ότι στη συνομιλία Τόμσεν – Βελκουλέσκου θέλουν να πάνε την Ελλάδα σε χρεοκοπία, εμένα στη δικιά μου ανάγνωση αυτό δεν μου προκύπτει. Προκύπτει η ανάλυση ότι οι Έλληνες, όταν φτάνουν στο παρά πέντε ... | link |  | 4 | 1 | 1 |
| | | «Το πρώτο που πρέπει κανείς | | | | | |

Ανάλυση Likes

| A/A | Χρήστης |
|-----|---------------------|
| 1 | Giannis Batsoulis |
| 2 | Cristine Exarchakou |
| 3 | Giorgos Keramidas |
| 4 | Jon Varian King |

Στιγμιότυπο 4: Ο χρήστης πάτησε σε ένα σύνδεσμο των likes μιας ανάρτησης για να δει τα ονόματα των χρηστών που έκαναν like.

Φίλτρα

1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:
2. Η ανάρτηση έχει πάνω από: likes
3. Η ανάρτηση έχει πάνω από: comments
4. Η ανάρτηση έχει πάνω από: shares
5. Το like έγινε από:
6. Το comment έγινε από:
7. Το comment περιέχει το κείμενο:

Επιτυχής προώθηση

Στιγμιότυπο 5: Ο χρήστης έχει εισάγει 2 φίλτρα (κείμενο τίτλου ανάρτησης) και likes ανάρτησης πάνω από 15 και μόλις πάτησε το κουμπί Προώθηση και εμφανίστηκε το μήνυμα προς ενημέρωσή του.

Φίλτρα

1. Ο τίτλος της ανάρτησης περιέχει το κείμενο:

2. Η ανάρτηση έχει πάνω από: likes

3. Η ανάρτηση έχει πάνω από: comments

4. Η ανάρτηση έχει πάνω από: shares

5. Το like έγινε από:

6. Το comment έγινε από:

7. Το comment περιέχει το κείμενο:

Στιγμιότυπο 6: Ο χρήστης έχει εισάγει 3 φίλτρα: η ανάρτηση να έχει πάνω από 15 shares, το comment έγινε από το χρήστη με όνομα Giorgos και ένα ή περισσότερα comments περιέχουν τη λέξη μπάσκετ.

Επίμαχες Αναρτήσεις από Facebook Group: Enikos.gr

Total: 41

| A/A | Τίτλος | Likes | Σχόλια | Shares | Εφαρμοζόμενα φίλτρα |
|-----|---|-------|--------|--------|---------------------|
| 1 | Ολυμπιακός - Ρεάλ 99-84 (Τελικό) | 25 | 1 | 4 | 2 |
| 2 | Ολυμπιακός - Ρεάλ 73-54 (Γ' Δεκάλεπτο) | 11 | 1 | 0 | 2 |
| 3 | Ολυμπιακός - Ρεάλ 53-33 (ΗΜ.) | 12 | 0 | 2 | 2 |
| 4 | Κρατίδιο της Γερμανίας έτοιμο να δεχτεί πρόσφυγες από την Ειδομένη | 11 | 1 | 3 | 2 |
| 5 | Το μήνυμα των προσφύγων στην παρέλαση της Μυτιλήνης - ΦΩΤΟ | 25 | 17 | 5 | 2 |
| 6 | «Χρυσός» ο Πετρούνιας στο Παγκόσμιο | 25 | 10 | 95 | 2 |
| 7 | Συγκινούν οι 4 μαθητές που παρέλασαν στη Γαύδο - ΒΙΝΤΕΟ | 25 | 0 | 6 | 2 |
| 8 | Βομβιστής αυτοκτονίας αιματοκύλησε την Βαγδάτη - Τουλάχιστον 29 νεκροί | 1 | 0 | 7 | 4 |
| 9 | Βρέθηκε χάρτης του αεροδρομίου των Βρυξελλών στη γιάφκα του Αμπαούντ στο Παγκράτι | 2 | 3 | 2 | 6 |
| 10 | Μεγάλο πάρτι στην Αβάνα από τους Rolling Stones - ΦΩΤΟ | 1 | 1 | 0 | 6 |
| 11 | Κουρουμπλής: Οι πρόσφυγες θα γίνουν πρεσβευτές της Ελλάδας όπου κι αν βρεθούν | 20 | 25 | 7 | 2 |
| 12 | Οι 8 πιο όμορφες παραλίες στην Ελλάδα για το 2016 | 11 | 0 | 1 | 2 |
| 13 | Συναγερμός στο Βέλγιο - Δολοφονήθηκε φύλακας πυρηνικής εγκατάστασης | 7 | 0 | 1 | 2 |
| 14 | Έτσι αποχαιρέτησαν τον Κρόιφ στο "Αμστερνταμ | 7 | 0 | 1 | 2 |

Στιγμιότυπο 7: Η σελίδα posts.html δείχνει τις επίμαχες αναρτήσεις, δηλαδή αυτές που έχει επιλέξει ο χρήστης μέσα από τη σελίδα index.html μέσα από τα φίλτρα που χρησιμοποίησε. Βλέπουμε πληροφορίες για τις επίμαχες αναρτήσεις. Στο άνω στιγμιότυπο, το χρήστη τον ενδιαφέρουν περισσότερο οι αναρτήσεις με τουλάχιστον 10 likes.

Επίλογος

Συνοψίζοντας, η απάντηση στο ερώτημα για το αν είναι δυνατή η δημιουργία ενός εργαλείου το οποίο να προωθεί επίμαχες αναρτήσεις από το Facebook προς site είναι θετική. Το εργαλείο αυτό δημιουργήθηκε και είναι και αρκετά αποτελεσματικό. Μπορεί να αξιοποιηθεί τόσο εμπορικά όσο και κοινωνικά. Ακόμα έχει εφαρμογές στην εκπαίδευση, στην ενημέρωση και στη διασκέδαση.

Υπάρχουν αρκετοί τρόποι να γίνει αυτό και στη προκειμένη εργασία προτάθηκε ένας από αυτούς. Υπάρχουν πολλά εργαλεία και πολλές γλώσσες προγραμματισμού για να επιτευχθεί ο στόχος και στο μέλλον, με τους ρυθμούς που μεταβάλλονται τα δεδομένα στον κυβερνοχώρο είναι πολύ πιθανό να γίνεται λιγότερο σύνθετα και πολύ πιο γρήγορα.

Ίσως στο μέλλον το εργαλείο αυτό να αναπτυχθεί και να προσφέρεται από το Facebook αυτούσιο. Έχει μεγάλες δυνατότητες όσον αφορά τη χρήση του και δεν θα ήτανε υπερβολή να πούμε ότι μελλοντικά ίσως να είναι απαραίτητο εργαλείο διαχείρισης ενός προφίλ στο Facebook ή σε άλλες πλατφόρμες.

Βιβλιογραφία

1. Ελληνική Βιβλιογραφία

- Γιακουμάκης Ε., Διαμαντίδης Ν. (2009) *Τεχνολογία Λογισμικού*, Εκδόσεις Αθ. Σταμούλης, Αθήνα

2. Ξενόγλωσση Βιβλιογραφία

- Cochran D. (2012) *Twitter Bootstrap Web Development How-To*, Packt Publishing, Birmingham
- Bibeault B. et al (2015) *jQuery in Action - Second Edition*, Manning Publications, Shelter Island, NY
- Zakas N.C. (2009) *Professional JavaScript for Web Developers - 3rd Edition*, Willey Publishing Inc, Indianapolis

3. Ιστοσελίδες

- **Bootstrap Documentation**, <http://getbootstrap.com/getting-started/>
- **Twitter Bootstrap**, [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- **HTTP**, https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- **PHP**, <http://www.php.net/>
- **PHP**, Wikipedia, <http://en.wikipedia.org/wiki/Php>
- **Jquery**, <http://jquery.com/>
- **JavaScript**, Wikipedia, <http://en.wikipedia.org/wiki/JavaScript>
- **Ajax**, Wikipedia, <http://en.wikipedia.org/wiki/Ajax>

- **HTML**, Wikipedia, <http://en.wikipedia.org/wiki/Html>
- **HTML5**, <http://en.wikipedia.org/wiki/HTML5>
- **JSON**, <http://en.wikipedia.org/wiki/JSON>
- **Web service**, http://en.wikipedia.org/wiki/Web_service
- **Open-source software**, http://en.wikipedia.org/wiki/Open-source_software
- **Cascading Style Sheets**, Wikipedia, <http://en.wikipedia.org/wiki/Css>
- **Client–server model**, Wikipedia, <http://en.wikipedia.org/wiki/Client-server>
- **Client (computing)**, Wikipedia, http://en.wikipedia.org/wiki/Client_%28computing%29
- **Server (computing)**, Wikipedia, http://en.wikipedia.org/wiki/Server_%28computing%29
- **Web browsers**, https://en.wikipedia.org/wiki/Web_browser
- **Facebook apps**, <https://developers.facebook.com>
- **Facebook**, <http://www.webopedia.com/TERM/F/Facebook.html>
- **XAMPP**, <https://en.wikipedia.org/wiki/XAMPP>
- **Facebook apps in Javascript**, <http://code.tutsplus.com/tutorials/build-your-first-facebook-app--net-471>