

Πτυχιακή Εργασία

Μακρογιάννη Κων/νου

Επιβλέπων καθηγητής: Κλεφτούρης Δημήτριος



Αριστοτέλειο Τεχνολογικό Ίδρυμα Θεσσαλονίκης
Τμήμα Πληροφορικής

Περιεχόμενα

σελ.

ΚΕΦΑΛΑΙΟ 1. Cloud computing.

Ορισμός Cloud computing	2
Ο όρος Υπηρεσία	2
Software as Service	3
Platform as Service	3
Storage as Service	4
Hardware as Service	4
Database as Service	4
Ο όρος Virtualization	5
Πλεονεκτήματα	5
Μειονεκτήματα	5

ΚΕΦΑΛΑΙΟ 2. ΧΑΜΡΡ.

Ετυμολογία	7
Δυνατότητες και απαιτήσεις	7
Χρήση	7

ΚΕΦΑΛΑΙΟ 3. Bootstrap.

Γενικά	9
Επισκόπηση	10
HTML5 doctype	10
Containers	10
Σύστημα πλέγματος(Grid system) και ανταποκρίσιμος σχεδιασμός(responsive design)	10
Media Queries	11
Επιλογές πλέγματος(Grid Options)	12
Η κατανόηση του CSS style	13
Javascript στοιχεία	13
Επαναχρησιμοποιήσιμα συστατικά	14
Buttons	14
Επιλογές	14
Μεγέθη	14

Ενεργή κατάσταση	15
Ιδιότητες	15
Μη ενεργή κατάσταση	16
Χρήση	16
ΚΕΦΑΛΑΙΟ 4. JQuery.	
Χαρακτηριστικά	18
Χρησιμοποιώντας το JQuery File Upload σε PHP ιστοσελίδες	18
Χρησιμοποιώντας JQuery File Upload(UI έκδοση) με έναν προσαρμοσμένο(custom) server-side upload handler.....	19
Content-Type Negotiation	20
ΚΕΦΑΛΑΙΟ 5. AJAX προγραμματισμός.	
Τι είναι ο AJAX	23
Μοντέλο web εφαρμογών	23
Μοντέλο επικοινωνίας	24
Τι είναι το JSON	24
Δομή JSON	25
Παραδείγματα JSON	25
Μετατροπή σε Object	25
Το αντικείμενο XMLHttpRequest	25
Ιδιότητες του XMLHttpRequest	26
Μέθοδοι και γεγονότα του XMLHttpRequest	26
Καταστάσεις ready state	26
Αρχικοποίηση και παραδείγματα	26
ΚΕΦΑΛΑΙΟ 6. Περιγραφή εφαρμογής.	
Σκοπός εφαρμογής	30
Λειτουργία εφαρμογής	30
Επικοινωνία σε πραγματικό χρόνο και αρχιτεκτονική	33
Διεπαφή χρήστη - UI	33
ΚΕΦΑΛΑΙΟ 7. Ανάλυση κώδικα εφαρμογής.	
Αρχείο index.php	35

Αρχείο header.php	35
Αρχείο footer.php	35
Αρχείο database.php	36
Αρχείο auth_check.php	36
Αρχείο register.php	36
Αρχείο header_dahsboard.php	37
Αρχείο dashboard.php	38
Αρχείο settings.php	39
Αρχείο contactChat.php	40
Αρχείο logout.php	40
Αρχείο makros-diploma.sql	41
References	42

Cloud computing

Ορισμός

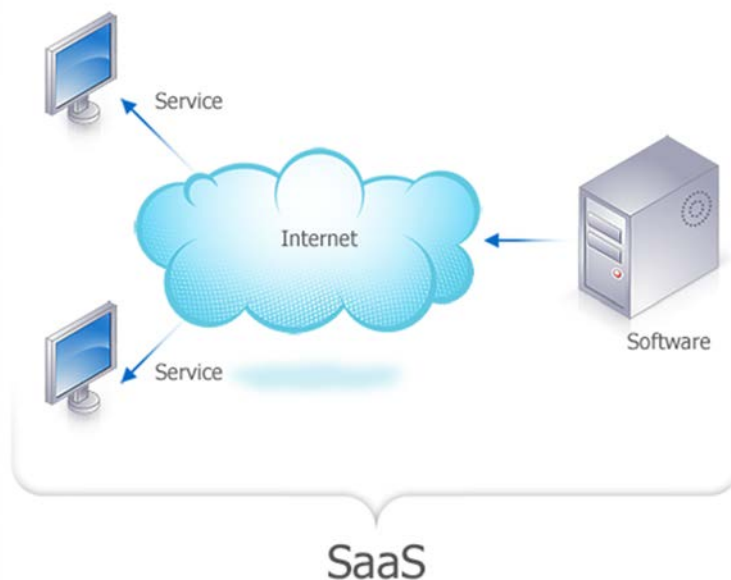
Το “**cloud computing**” είναι μία δομή, με την οποία μας δίνεται η δυνατότητα να έχουμε πρόσβαση και να χρησιμοποιούμε web εφαρμογές χωρίς να τις διαθέτουμε στον υπολογιστή μας ή σε κάποια άλλη συσκευή που είναι διασυνδεδεμένη με το ίντερνετ. Σε αυτή τη δομή η εφαρμογή βρίσκεται σε ένα server και εμείς τη χρησιμοποιούμε χωρίς να χρειάζεται να την εγκαταστήσουμε στον υπολογιστή μας.



Ο όρος “υπηρεσία” στο cloud computing.

“Υπηρεσία” είναι η δυνατότητα που μας παρέχει το cloud σύστημα να χρησιμοποιούμε κάποιο software μέσα σε αυτό το δίκτυο. Αυτή η δυνατότητα ονομάζεται “υπηρεσία”. Παρακάτω παρουσιάζονται κάποια βασικά είδη υπηρεσιών που μας προσφέρει το **cloud computing**.

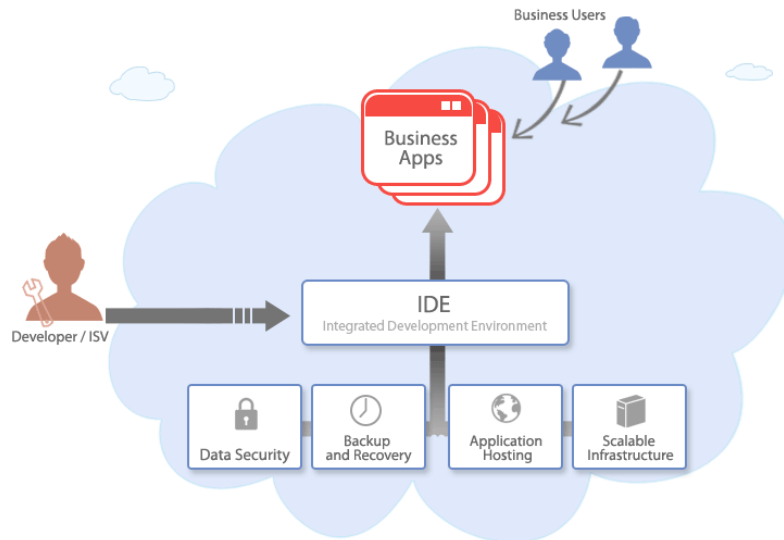
Software as Service (SaaS)



Σε αυτόν τον τύπο υπάρχει ένα application που βρίσκεται σε ένα **cloud server** και ο χρήστης μπορεί να έχει πρόσβαση σε αυτό μέσω μίας απλής σύνδεσης στο ίντερνετ. Το software αυτό ανήκει σε κάποιον κατασκευη και ο χρήστης το πληρώνει ανάλογα με την χρήση που του κάνει και τους πόρους που χρειάζεται. Το βασικό πλεονέκτημα του μοντέλου “**software as service**” είναι ότι ο κατασκευαστής αναλαμβάνει τα έξοδα συντήρησης του software καθώς και τη φιλοξενία του σε κάποιον cloud server. Ο χρήστης πληρώνει μόνο την χρήση που κάνει(αν και υπάρχουν και **cloud applications** που είναι δωρεάν). Επίσης το μοντέλο SaaS είναι δημιουργημένο με βασικό γνώμονα τη σωστή λειτουργία του software με χρήση **browser**. Όσον αφορά την ασφάλεια των διαφόρων εφαρμογών, συνήθως χρησιμοποιείται SSL (Secure Sockets Layer) το οποίο είναι παγκοσμίως αναγνωρισμένο. Έτσι ,οι χρήστες μπορούν με ασφάλεια να χρησιμοποιήσουν το cloud application.

Platform as Service (PaaS)

Αυτό το μοντέλο μοιάζει πολύ με το προηγούμενο. Το βασικό του στοιχείο είναι ότι **παρέχει την πλατφόρμα** την οποία χρησιμοποιεί ένας χρήστης για να δημιουργήσει κάτι, για παράδειγμα ένα web application, χωρίς να εγκαταστήσει τίποτα.Το “platform as service” μοντέλο χρησιμοποιείται πιο πολύ για δημιουργία web interfaces, web εφαρμογών κλπ. Ένα σημαντικό πρόβλημα που υπάρχει με αυτό το μοντέλο είναι ότι αυτή η εφαρμογή που δημιουργούμε βασίζεται σε ένα συγκεκριμένο framework και υπάρχει πιθανότητα αν θελήσουμε να την μεταφέρουμε σε άλλο παροχέα cloud υπηρεσιών αυτή να μη λειτουργεί σωστά.



Storage as a service (StaaS)

Στο μοντέλο αυτό υπάρχει κάποιος πάροχος αποθηκευτικού χώρου online ο οποίος στην ουσία τον νοικιάζει έναντι κάποιας αμοιβής.

Hardware as Service (HaaS)

Ο προμηθευτής αυτής της cloud υπηρεσίας παρέχει στον χρήστη έναντι “ενοικίου”-αμοιβής το hardware που χρειάζεται όπως web servers, μνήμη CPU, αποθηκευτικό χώρο και ότι άλλο χρειάζεται ο χρήστης σε **επίπεδο hardware**. Τα χρήματα που πληρώνει κάποιος στο HaaS είναι αντίστοιχα της χρήσεως των πόρων του συστήματος που κάνει.

Database as Service (DaaS)

Σε αυτό το μοντέλο υπάρχει μία υπηρεσία online παρέχει την βάση δεδομένων την οποία μπορούμε να χρησιμοποιήσουμε με κάποιο web application. Σε αυτό το μοντέλο το βασικό πλεονέκτημα είναι ότι πληρώνουμε ανάλογα με την χρήση. Ουσιαστικά όσο πιο πολύ κόσμος χρησιμοποιεί την εφαρμογή μας τόσο περισσότερα χρήματα πληρώνουμε.

Ο όρος “Virtualization” στο cloud computing.

Η τεχνολογία virtualization είναι η κινητήριος δύναμη του cloud computing. Στην τεχνολογία αυτή μπορούμε να χωρίσουμε ένα φυσικό hardware, ένα webserver, σε πολλά κομμάτια που το κάθε ένα τρέχει το δικό του λειτουργικό. Έτσι επιτυγχάνεται άριστη λειτουργικότητα, ταχύτητα και απόλυτη αξιοποίηση των πόρων του συστήματος. Αυτά τα κομμάτια είναι σαν εικονικοί servers και ονομάζονται “**virtual machines**” ή **VMs**. Το **cloud** συνδέεται με την έννοια virtualization γιατί ουσιαστικά είναι ένα σύνολο συνδεδεμένων virtual machines. Αυτός είναι ο λόγος που τα cloud δίκτυα έχουν τόσες πολλές δυνατότητες και επεκτασιμότητα διότι μοιράζονται του πόρους διάφορων συνδεδεμένων virtual machines.

Πλεονεκτήματα του Cloud computing.

- **Οικονομία.**
Αυτό είναι από τα πιο βασικά πλεονεκτήματα του cloud computing. Το κόστος που μπορεί να έχει ένα λογισμικό ίσως να είναι απαγορευτικό για μία μικρή εταιρία. Με το “cloud” τα δεδομένα αυτά αλλάζουν καθώς η εταιρία δεν πληρώνει την εφαρμογή αλλά πληρώνει την χρήση της. Συνήθως σε cloud δίκτυα υπάρχουν πολλές δυνατότητες και “πακέτα” για την πληρωμή της χρήσης κάποιας εφαρμογής.
- **Μεγάλος Αποθηκευτικός Χώρος.**
Η αποθήκευση των διαφόρων πληροφοριών είναι θέμα υψίστης σημασίας. Με το **cloud computing** έχουμε συνήθως όσο αποθηκευτικό χώρο θα χρειαστούμε.
- **Πρόσβαση από οποιαδήποτε συσκευή διαθέτει σύνδεση στο ίντερνετ.**
- **Πολύ μεγάλη ευελιξία.**

Μειονεκτήματα του Cloud computing.

- **Ασφάλεια δεδομένων.**
Είναι λογικό κάποιες φορές να έχουμε συγκεκριμένα δεδομένα σε δικό μας τοπικό server και όχι στο cloud.
- **Αυξημένη πολυπλοκότητα.**
Αυτό συμβαίνει όταν έχουμε μία εφαρμογή αποθηκευμένη κάπου τοπικά, σε ένα δικό μας webserver και προσπαθούμε να την κάνουμε να επικοινωνήσει με μία άλλη στο cloud. Τα πράγματα εκεί γίνονται αρκετά περίπλοκα και πολλές φορές η λύση εκτός ότι δεν είναι προφανής, αποτυγχάνει.

XAMPP^[1]

Για τις ανάγκες της εφαρμογής χρησιμοποιήθηκε το **XAMPP**. Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοιχτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Ετυμολογία

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- **X** (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- **Apache http εξυπηρετητής**
- **MySQL**
- **PHP**
- **Perl**

Το XAMPP είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets.

Δυνατότητες και απαιτήσεις

Το XAMPP προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Το XAMPP έχει δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl. Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

Χρήση

Επίσημα οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί. Στην πράξη το XAMPP ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων. Υπάρχει ένα ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών και υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite.

Όταν εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Η σύνδεση στον localhost μέσω του FTP μπορεί να γίνει με το όνομα χρήστη «newuser» και το κωδικό «wampp». Για την βάση δεδομένων MySQL υπάρχει ο χρήστης «root» χωρίς κωδικό πρόσβασης.

Bootstrap^{[2][3]}

Γενικά

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις Javascript. Έχει το πιο δημοφιλές πρόγραμμα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. Τον Αύγουστο του 2011 κυκλοφόρησε Twitter Bootstrap ως λογισμικό ανοιχτού κώδικα. Τον Φεβρουάριο του 2012, ήταν το πιο δημοφιλές έργο ανάπτυξης GitHub.

Το Bootstrap αποτελείται ουσιαστικά από μια σειρά styles(stylesheets) που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα style που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους.

Προσαρμογές είναι δυνατές σε περιορισμένη έκταση μέσω ενός κεντρικού style διαμόρφωσης. Η χρήση γλώσσας style επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς (operators), ένθετους επιλογείς, γνωστά και ως μείγματα (mixins). Από την έκδοση 2.0, η διαμόρφωση του Bootstrap έχει επίσης μία ειδική επιλογή "Προσαρμογή" στην τεκμηρίωση (documentation). Επιπλέον, ο σχεδιαστής του έργου επιλέγει σε μια φόρμα τα επιθυμητά συστατικά και τα προσαρμόζει, εάν είναι αναγκαίο, σε τιμές διαφόρων εναλλακτικών λύσεων για τις ανάγκες του. Στη συνέχεια δημιουργείται ένα πακέτο που περιλαμβάνει ήδη το προ-χτισμένο CSS style.

HTML5 doctype

Το Bootstrap, όπως αναφέρθηκε παραπάνω, περιέχει HTML και CSS για τις μορφές τυπογραφίας. Επομένως είναι απαραίτητη η χρήση του *HTML doctype* στην αρχή κάθε εργασίας (project).

```
<!DOCTYPE html>
<html lang="el">
  ...
</html>
```

Containers

Στο Bootstrap υπάρχει η απαίτηση ένα containing στοιχείο να "τυλίξει" τα περιεχόμενα της ιστοσελίδας μας και να στεγάσει όλο το διασυνδεδεμένο σύστημα πλέγματος (Grid System). Υπάρχει το ενδεχόμενο να γίνει η χρήση δύο τέτοιων στοιχείων σε μία εργασία (project). Σ' αυτήν την περίπτωση αξίζει να σημειωθεί ότι, δεδομένου του padding, κανένα container στοιχείο δεν μπορεί να φωλιάσει στο άλλο.

Ακολουθούν δύο παραδείγματα. Στο πρώτο χρησιμοποιώντας κλάση `.container` δημιουργούμε ένα container μεταβλητού πλάτους, ενώ στο δεύτερο χρησιμοποιώντας κλάση `.container-fluid` δημιουργούμε ένα σταθερό container σε ολόκληρο το παράθυρο θέασής μας. Στην εφαρμογή χρησιμοποιήθηκε κλάση `.container`.

```
<div class="container">
  ...
</div>
```

```
<div class="container-fluid">
  ...
</div>
```

Σύστημα πλέγματος(Grid System) και ανταποκρίσιμος σχεδιασμός(responsive design)

Για τις ανάγκες της εφαρμογής χρησιμοποιήθηκε πλέγμα 1140 pixels, έχοντας αφαιρέσει 30 pixels padding αριστερά και δεξιά από το κύριο container. Το Bootstrap έρχεται σταθερό με 768 pixel πλάτος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη μέσω του ανταποκριτικού

σχεδιασμού του Bootstrap, η οποία μεγαλώνει κατάλληλα μέχρι δώδεκα στήλες καθώς μεγαλώνει η συσκευή ή το παράθυρο θέασης. Και για τις δύο περιπτώσεις container, η εργαλειοθήκη έχει τέσσερις παραλλαγές για χρήση σε διαφορετικές αναλύσεις οθόνης και τύπους συσκευών όπως κινητά τηλέφωνα, tablets και υπολογιστές με χαμηλή και υψηλή ανάλυση.

Παρακάτω φαίνεται πως λειτουργεί το Σύστημα πλέγματος:

- Για την κατάλληλη στοίχιση ή το padding οι γραμμές πρέπει να τοποθετηθούν μέσα σε ένα container τύπου `.container` ή `.container-fluid`. Στη δική μας περίπτωση `.container`.
- Χρησιμοποίηση των γραμμών για τη δημιουργία οριζόντιων ομάδων από στήλες.
- Το περιεχόμενο πρέπει να το τοποθετηθεί μέσα στις στήλες.
- Μόνο οι στήλες είναι άμεσα παιδιά (children) των γραμμών.
- Προκαθορισμένες κλάσεις όπως η `.row` και η `.col-xs-4` είναι διαθέσιμες για γρήγορη δημιουργία διατάξεων από πλέγματα.
- Οι στήλες δημιουργούν κενά μεταξύ τους, τα λεγόμενα *gutters*. Αυτό συμβαίνει λόγω του `padding`. Το padding αντισταθμίζεται μέσω του αρνητικού περιθωρίου στη `.row` για την πρώτη και την τελευταία στήλη.
- Υπάρχουν διαθέσιμες έως και δώδεκα στήλες για τη διάταξη του περιεχομένου μας. Ανάλογα με τις ανάγκες μας επιλέγουμε αν θα χρησιμοποιήσουμε και τις δώδεκα ή θα τις "συμπτίξουμε". Για παράδειγμα, τρεις ίσες στήλες θα χρησιμοποιήσουν τρεις κλάσεις `.col-xs-4`.
- Εάν μέσα σε μία γραμμή τοποθετηθούν περισσότερες από δώδεκα στήλες, κάθε επιπλέον ομάδα από στήλες θα μετακινηθεί σε νέα γραμμή (σαν μία μονάδα).
- Οι κλάσεις πλέγματος του Bootstrap έχουν εφαρμογή σε οθόνες ίσες ή μεγαλύτερες από αυτές που ορίζουν τα οριακά σημεία (breakpoints) των συσκευών μας και παρακάμπτουν κλάσεις που στοχεύουν σε μικρότερες συσκευές. Ως εκ τούτου, εφαρμόζοντας μία κλάση `.col-md` σε μία μεσαία συσκευή θα επηρεάσει τη διάταξη των περιεχομένων, όπως επίσης θα επηρεάσει τη διάταξη και σε μία μεγάλη, εάν δεν υπάρχει η κλάση `.col-lg`.

Media Queries

Χρησιμοποιούμε τις παρακάτω εντολές για τη δημιουργία των οριακών σημείων(breakpoints) του Συστήματος πλέγματος (Grid System) και για να απευθυνθούμε στην αντίστοιχη ανάλυση της κάθε συσκευής.

```
/* Σε πολύ μικρές συσκευές (τηλέφωνα, λιγότερο από 768 pixels) */  
/* Δε χρειάζεται media query, γιατί αυτό είναι το στάνταρντ στο Bootstrap
```

```
/* Μικρές συσκευές (tablets, 768 pixels και πάνω) */  
@media (min-width: @screen-sm-min) { ... }
```

```
/* Μεσαίες συσκευές (υπολογιστές γραφείου, 992px και πάνω) */  
@media (min-width: @screen-md-min) { ... }
```

```
/* Μεγάλες συσκευές (μεγάλοι υπολογιστές, 1200px και πάνω) */
@media (min-width: @screen-lg-min) { ... }
```

Υπάρχουν φορές που χρησιμοποιούμε το max-width στα Media Queries για να περιοριστεί το CSS σε μικρότερη ομάδα συσκευών.

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

Επιλογές πλέγματος (Grid Options)

	Πολύ μικρές συσκευές τηλέφωνα(<768px)	Μικρές συσκευές tablets(>=768px)	Μεσαίες συσκευές υπολογιστής(>=992px)	Μεγάλες συσκευές υπολογιστής(>=1200px)
Πλάτος container	Κανένα (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# αριθμός στηλών	12			
Πλάτος στήλης	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of the column)			
Nestable	Yes			
Offsets	Yes			
Ταξινόμηση στηλών	Yes			

Ακολουθούν παραδείγματα σε κινητά, tablets και υπολογιστές.

σε κινητά:

.col-xs-12 .col-sm-6 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4
.col-xs-6 .col-sm-4	

σε tablets:

.col-xs-12 .col-sm-6 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4
.col-xs-6 .col-sm-4	

σε υπολογιστές:

.col-xs-6 .col-sm-4 .col-md-8	.col-xs-6 .col-sm-4	
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4

```
<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
<div class="row">
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <!-- Optional: clear the XS cols if their content doesn't match in height -->
  <div class="clearfix visible-xs-block"></div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>
```

Η κατανόηση του CSS style

Το Bootstrap παρέχει ένα σύνολο από style με βασικούς ορισμούς style για όλα τα βασικά στοιχεία HTML. Αυτά δημιουργούν ενιαία, σύγχρονη εμφάνιση για πίνακες, μορφοποίηση κειμένου, καθώς και στοιχεία μιας φόρμας.

JavaScript στοιχεία

Το Bootstrap έρχεται με πολλά συστατικά JavaScript σε μια μορφή jQuery plugin. Παρέχουν πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρουσέλ. Μπορούν επίσης να επεκτείνουν τη λειτουργικότητα ορισμένων υφιστάμενων στοιχείων της διασύνδεσης, όπως για παράδειγμα μια αυτόματη πλήρη λειτουργία για πεδία εισαγωγής. Στην έκδοση 2.0, υποστηρίζονται τα ακόλουθα JavaScript plugins: Modal, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel και Typeahead.

Επαναχρησιμοποιήσιμα συστατικά

Εκτός από τα βασικά HTML στοιχεία, το Bootstrap περιέχει και άλλα στοιχεία περιβάλλοντος που χρησιμοποιούνται συχνά. Αυτά περιλαμβάνουν κουμπιά με προηγμένα χαρακτηριστικά (π.χ. ομαδοποίηση κουμπιών ή drop -down επιλογή, οριζόντιες και κάθετες καρτέλες, πλοήγηση, σελιδοποίηση, κ.λπ.), ετικέτες, προηγμένες τυπογραφικές δυνατότητες, εικονίδια, προειδοποιητικά μηνύματα και μια γραμμή προόδου.

Buttons

Επιλογές

Χρήση των διαθέσιμων κλάσεων κουμπιών για δημιουργία κουμπιών με style.

Παράδειγμα



```
<!--στάνταρντ κουμπί -->
<button type="button" class="btn btn-default">Default</button>

<!--παρέχει έξτρα οπτικό βάρος και δείχνει τη βασική λειτουργία(?) σε ένα σετ από κουμπιά -->
<button type="button" class="btn btn-primary">Primary</button>

<!-- δείχνει μία επιτυχής ή θετική δράση -->
<button type="button" class="btn btn-success">Success</button>

<!-- κουμπί για πληροφοριακά μηνύματα ειδοποίησης (alert) -->
<button type="button" class="btn btn-info">Info</button>

<!-- δείχνει την προσοχή που απαιτείται για μία πράξη -->
<button type="button" class="btn btn-warning">Warning</button>

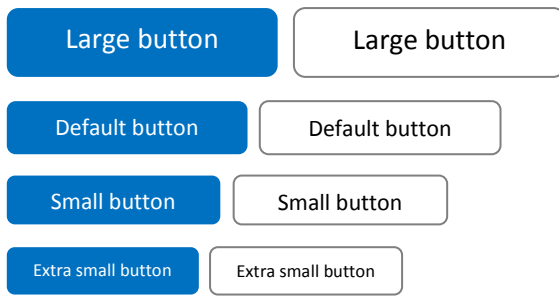
<!-- δείχνει μια επικίνδυνη ή πιθανά αρνητική πράξη -->
<button type="button" class="btn btn-danger">Danger</button>

<!-- ξετονίζει ένα κουμπί δείχνοντας το σαν ένα link , διατηρώντας όμως τις λειτουργίας που έχει ένα κουμπί -->
<button type="button" class="btn btn-link">Link</button>
```

Μεγέθη

Για μεγαλύτερα κουμπιά χρησιμοποιούμε την `.btn-lg` και για μικρότερα την `.btn-sm`, ενώ για επιπλέον μεγέθη χρησιμοποιούμε την `.btn-xs`.

Παράδειγμα



```
<p>  
  <button type="button" class="btn btn-primary btn-lg">Large button</button>  
  <button type="button" class="btn btn-default btn-lg">Large button</button>  
</p>  
<p>  
  <button type="button" class="btn btn-primary">Default button</button>  
  <button type="button" class="btn btn-default">Default button</button>  
</p>  
<p>  
  <button type="button" class="btn btn-primary btn-sm">Small button</button>  
  <button type="button" class="btn btn-default btn-sm">Small button</button>  
</p>  
<p>  
  <button type="button" class="btn btn-primary btn-xs">Extra small button</button>  
  <button type="button" class="btn btn-default btn-xs">Extra small button</button>  
</p>
```

Ενεργή κατάσταση κουμπιού (Active state)

Τα κουμπιά παρουσιάζονται πιεσμένα (με ένα πιο σκούρο φόντο, πιο σκούρο πλαίσιο και εσωτερική σκίαση) όταν είναι ενεργά. Η τροποποίηση αυτών των ιδιοτήτων `<button>` γίνεται μέσω `:active`. Για `<a>` ιδιότητες γίνεται μέσω `.active`. Ωστόσο μπορεί να χρησιμοποιηθεί η `.active` σε ένα `<button>`s - απαιτείται όμως η αναπαραγωγή της ενεργής κατάστασης μέσω προγραμματισμού.

Ιδιότητες κουμπιού (Button element)

Αν απαιτείται όμοια εμφάνιση κουμπιών δε χρειάζεται να προσθέσουμε την `:active`, καθώς είναι ψευδο-κλάση, αλλά χρησιμοποιούμε την `.active`.

Παράδειγμα



```
<button type="button" class="btn btn-primary btn-lg active">Primary button</button>  
<button type="button" class="btn btn-default btn-lg active">Button</button>
```

Μη ενεργή κατάσταση κουμπιού (Disable state)

Ένα κουμπί δίνει την αίσθηση του ανενεργού αν μειωθεί η φωτεινότητά του κατά 50%. Προσθήκη της **disabled** ιδιότητας στο κουμπί `<button>`.

Παράδειγμα

Primary button

Button

Χρήση

Για να χρησιμοποιηθεί το Bootstrap σε μια σελίδα HTML, ο σχεδιαστής του έργου κάνει λήψη του style CSS Bootstrap και περιλαμβάνει μια σύνδεση στο αρχείο HTML. Αν ο προγραμματιστής θέλει να χρησιμοποιήσει τα στοιχεία JavaScript, θα πρέπει να αναφέρονται μαζί με τη βιβλιοθήκη jQuery στο HTML έγγραφο.

JQuery^[4]

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

Η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT.

Χαρακτηριστικά

- DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών *Sizzle*.
- DOM διάσχιση και τροποποίηση (υποστηρίζοντας CSS 1-3)
- χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
- Events
- Εφέ και κινητά στοιχεία
- AJAX
- Επεκτασιμότητα μέσω plug-ins
- Εργαλεία όπως πληροφορίες user-agent, ανίχνευση χαρακτηριστικών.
- Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προγράμματα περιήγησης.
- Υποστήριξη πολλαπλών φυλλομετρητών.

Το JQuery File Upload πέρα από τη χρήση του σε PHP ιστοσελίδες μπορεί να χρησιμοποιηθεί και με *Google App Engine*, με *Node.js* και με έναν προσαρμοσμένο (*custom*) *server-side upload handler*.

Χρησιμοποιώντας το JQuery File Upload σε PHP ιστοσελίδες

Το παρακάτω παράδειγμα λειτουργεί πέρα από τα καθιερωμένα και χρειάζεται ένα μόνο βήμα για να προσθέσουμε το JQuery File Upload plugin σε PHP ιστοσελίδες.

1. Κατεβάζουμε το συμπιεσμένο plugin, το αποσυμπιέζουμε και το ανεβάζουμε στον server. Αν επιθυμούμε μπορούμε να μετονομάσουμε τον αποσυμπιεσμένο φάκελο.

Επισκεπτόμαστε το directory που ανεβάσαμε το φάκελο. Θα πρέπει να δούμε την ίδια File Upload διεπαφή με αυτή του demo, επιτρέποντάς μας να ανεβάσουμε αρχεία στην ιστοσελίδα μας. Εάν αυτή η λειτουργία δεν είναι εφικτή, πρέπει να σιγουρευτούμε ότι στα δικαιώματα του php/files directory επιτρέπεται η δυνατότητα εγγραφής (από τον server).

Επισήμανση:

Πρέπει να έχουμε υπόψιν μας τυχών παράγοντες ασφαλείας όταν τρέχουμε στον server ένα File Upload Handler.

Χρησιμοποιώντας jQuery File Upload(UI έκδοση) με έναν προσαρμοσμένο(custom) server-side upload handler.

1. Ενσωματώνουμε έναν file-upload handler στην πλατφόρμα μας (Ruby, Python, Java κ.α.) που διαχειρίζεται απλά file uploads και τον ανεβάζουμε στον server μας.
2. Κατεβάζουμε το συμπιεσμένο plugin και το αποσυμπιέζουμε.
3. Τροποποιούμε το url της main.js με το url του προσαρμοσμένου File Upload Handler. Εναλλακτικά μπορούμε να επεξεργαστούμε το Index.html και να τροποποιήσουμε την ιδιότητα Action (της HTML φόρμας μας) με το url του προσαρμοσμένου μας File Upload Handler.
Αν ο File Upload Handler χρειάζεται διαφορετική αρχικοποίηση από αυτήν του files[] για το ανέβασμα αρχείων, μπορούμε να αλλάξουμε είτε τη μεταβλητή file input name είτε την επιλογή paramName.
4. Ανεβάζουμε το φάκελο jQuery File-Upload στον server μας.
5. Τροποποιούμε τον προσαρμοσμένο(custom) Upload Handler του server μας, ώστε να επιστρέφει μία JSON (JavaScript Object Notation) απάντηση όμοια με την παρακάτω:

```
{ "files": [  
  {  
    "name": "picture1.jpg",  
    "size": 902604,  
    "url": "http://example.org/files/picture1.jpg",  
    "thumbnailUrl": "http://example.org/files/thumbnail/picture1.jpg",  
    "deleteUrl": "http://example.org/files/picture1.jpg",  
    "deleteType": "DELETE"  
  },  
  {  
    "name": "picture2.jpg",  
    "size": 841946,  
    "url": "http://example.org/files/picture2.jpg",  
    "thumbnailUrl": "http://example.org/files/thumbnail/picture2.jpg",  
    "deleteUrl": "http://example.org/files/picture2.jpg",  
    "deleteType": "DELETE"  
  }  
]  
}]
```

Αν θέλουμε να επιστρέφει και να εμφανίζει σφάλματα, προσθέτουμε την ιδιότητα error σε κάθε ένα από τα αντικείμενα-αρχεία.

```
{ "files": [  
  {  
    "name": "picture1.jpg",  
    "size": 902604,  
    "error": "Filetype not allowed"  
  },  
  {  
    "name": "picture2.jpg",  
    "size": 841946,  
    "error": "Filetype not allowed"  
  }  
]  
}]
```

Για κάθε αντικείμενο-αρχείο που σβήνεται χρησιμοποιώντας το κουμπί "delete", ο κώδικας θα πρέπει να είναι ο εξής:

```
{ "files": [
  {
    "picture1.jpg": true
  },
  {
    "picture2.jpg": true
  }
]}
```

Αξίζει να σημειωθεί ότι η απάντηση που επιστρέφεται θα πρέπει να είναι ένα αντικείμενο JSON που περιέχει έναν πίνακα αρχείων, ακόμα και όταν αυτός ο πίνακας έχει μόνο ένα αντικείμενο-αρχείο.

Πλέον αν επισκεφτούμε το φάκελο που ανεβάσαμε στον server μας θα πρέπει να μας εμφανιστεί η διεπαφή όπως αυτή στο demo, επιτρέποντάς μας να ανεβάσουμε αρχεία στην ιστοσελίδα μας.

Content-Type Negotiation

Το plugin File-Upload χρησιμοποιεί ένα μηχανισμό μεταφοράς τύπου Iframe για φυλλομετρητές όπως ο Microsoft Internet Explorer και ο Opera, οι οποίοι δεν υποστηρίζουν ακόμα ανέβασμα αρχείων τύπου XMLHttpRequest.

Ο μηχανισμός Iframe απαιτεί η απάντηση JSON να οριστεί σαν text/plain ή text/html, αλλιώς ο ορισμός της ως application/json θα έχει την εμφάνιση ενός μη επιθυμητού παράθυρου διαλόγου.

Μπορούμε να χρησιμοποιήσουμε την κεφαλίδα(header) Accept για την απόδοση διαφορετικών ειδών περιεχομένου σε μία απάντηση όταν ανεβάζουμε ένα αρχείο.

Παράδειγμα (PHP):

```
<?php
header('Vary: Accept');
if (isset($_SERVER['HTTP_ACCEPT']) &&
    (strpos($_SERVER['HTTP_ACCEPT'], 'application/json') !== false)) {
    header('Content-type: application/json');
} else {
    header('Content-type: text/plain');
}
?>
```

Χάρης στην επιλογή content_type η κεφαλίδα έχει αποδοθεί σωστά σε όλους τους φυλλομετρητές.

Για τα πρακτικά αυτή είναι η to_fileupload μέθοδος:

```
def to_fileupload(attachment_name, attachment_style)
```

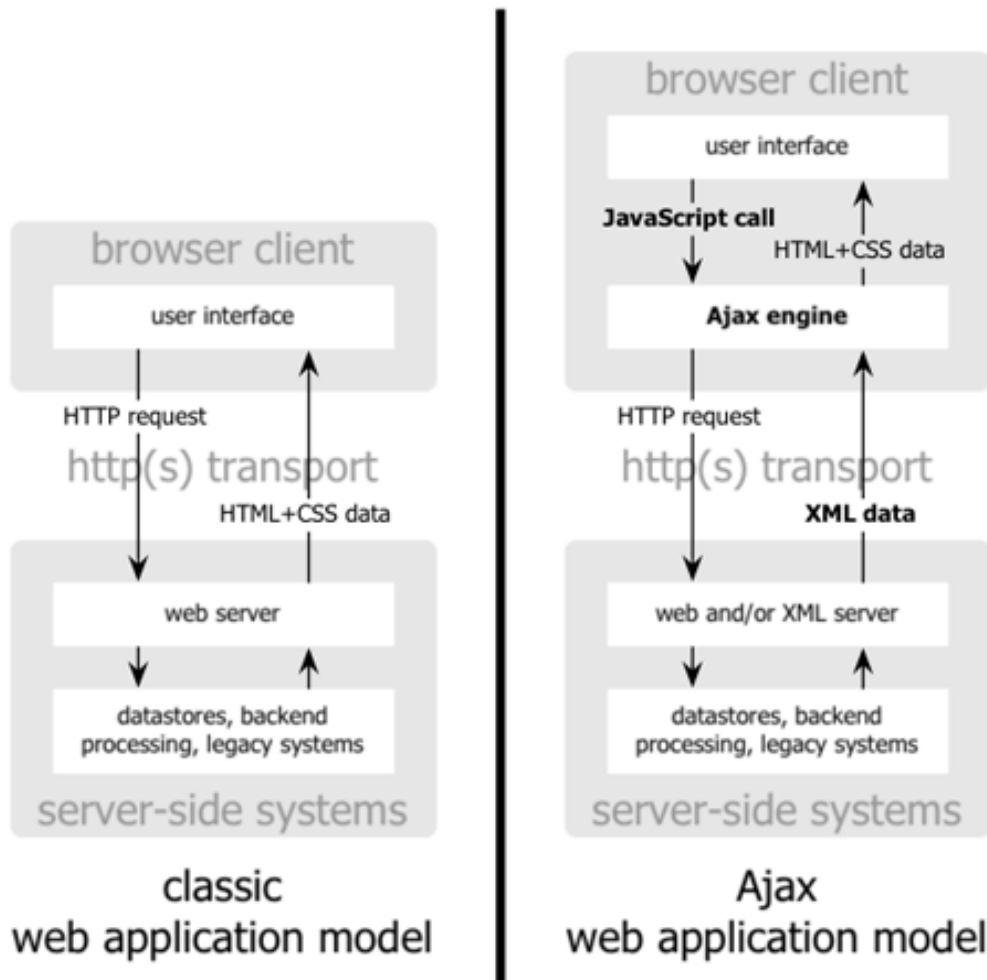
```
{
  files: [
    {
      id: read_attribute(:id),
      name: read_attribute("#{attachment_name}_file_name"),
      type: read_attribute("#{attachment_name}_content_type"),
      size: read_attribute("#{attachment_name}_file_size"),
      url: send(attachment_name).url(attachment_style)
    }
  ]
}
end
```

AJAX προγραμματισμός

Τι είναι ο AJAX

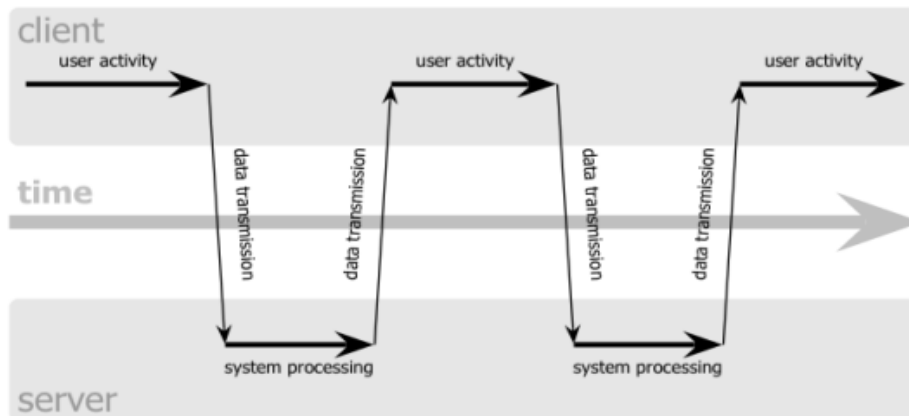
Ο AJAX (Asynchronous Javascript And Xml) είναι μια ασύγχρονη επικοινωνία μεταξύ client (browser – αρχικοποίηση επικοινωνίας μέσω συμβάντων και javascript) και server (χρήση XML για την αποστολή των δεδομένων). Δεν είναι νέα γλώσσα προγραμματισμού, αλλά είναι ένας νέος τρόπος να χρησιμοποιούμε υπάρχοντα πρότυπα (Javascript, XML, HTML, DOM, CSS).

Μοντέλο web εφαρμογών

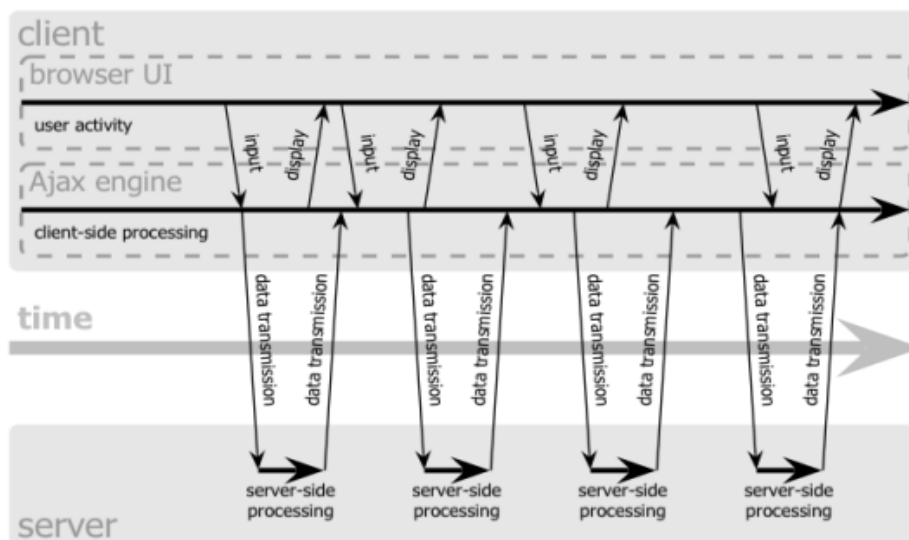


Μοντέλο επικοινωνίας

classic web application model (synchronous)



Ajax web application model (asynchronous)



Τι είναι το JSON

Το JSON (Javascript Object Notation) είναι ένα lightweight data-interchange format εύκολο για τους ανθρώπους να το διαβάσουν και να το γράψουν και εύκολο για τις μηχανές να το αναλύσουν και να δημιουργήσουν. Βασίζεται σε ένα υποσύνολο της JavaScript Programming Language(Standard ECMA-262 3rd Edition - December 1999) με μορφή κειμένου που είναι εντελώς ανεξάρτητη από την γλώσσα αλλά χρησιμοποιεί συμβάσεις που είναι γνωστές στους προγραμματιστές γλωσσών όπως C, C++, C#, Java, JavaScript, Perl, Python κ.α.

Δομή JSON

Το JSON αποτελείται από δύο δομές. Η πρώτη είναι μια συλλογή από ζεύγη name/value όπου σε διάφορες γλώσσες αυτό γίνεται αντιληπτό ως object, record, struct, dictionary, hash table, keyed list, ή associative array και η δεύτερη είναι μία ταξινομημένη λίστα τιμών όπου στις περισσότερες γλώσσες αυτό γίνεται αντιληπτό σαν array, vector, list ή sequence.

Παραδείγματα JSON

Javascript Object

```
{
  "statCode" : 200,
  "statText" : "Ok",
  "interval" : ["hour", "day", "week"],
  "ids": [10, 30, 45],
  "readings": [
    {
      "timestamp": 1398816000000,
      "reading": 26.720923959946468
    },
    {
      "timestamp": 1398902400000,
      "reading": 27.733928534021615
    }
  ]
}
```

JSON (text)

```
{'+
  "statCode" : 200, '+
  "statText" : "Ok", '+
  "interval" : ["hour", "day", "week"], '+
  "ids": [10, 30, 45], '+
  "readings": ['+
    {'+
      "timestamp": 1398816000000, '+
      "reading": 26.720923959946468'+
    }, '+
    {'+
      "timestamp": 1398902400000, '+
      "reading": 27.733928534021615'+
    } '+
  ] '+
}'
```

Μετατροπή σε Object

Η μετατροπή σε Object γίνεται με την eval() (δε συνίσταται για λόγους ασφαλείας), όπου το κείμενο πρέπει να γίνει wrap σε παρενθέσεις, με την εγγενή υποστήριξη του browser (εφόσον διατίθεται), με τη json_parse() από τη βιβλιοθήκη <https://github.com/douglascrockford/JSON-js> και με την υποστήριξη που προσφέρουν βιβλιοθήκες όπως jQuery, Yahoo UI, Prototype, Dojo, ECMAScript 1.5.

Το αντικείμενο XMLHttpRequest

Όλοι οι νέοι browsers υποστηρίζουν το Javascript αντικείμενο XMLHttpRequest με εξαίρεση τις εκδόσεις του IE πριν την 7, οι οποίες παρέχουν υποστήριξη μέσω αντικειμένου ActiveX. Το αντικείμενο XMLHttpRequest διαθέτει ένα σύνολο ιδιοτήτων, μεθόδων και γεγονότων τα οποία χρησιμοποιεί ο προγραμματιστής για την υλοποίηση της ασύγχρονης επικοινωνίας

Ιδιότητες του XMLHttpRequest

- **readyState**: Επιστρέφει την τρέχουσα κατάσταση της λειτουργίας του αιτήματος
- **responseText**: Επιστρέφει το σώμα της απόκρισης σαν αλφαριθμητικό.
- **responseXML**: Επιστρέφει το σώμα της απόκρισης σαν ένα αντικείμενο XML Document.
- **status**: Επιστρέφει τον HTTP κωδικό κατάστασης του αιτήματος.
- **statusText**: Επιστρέφει την φιλική μορφή (κείμενο) της HTTP κατάστασης του αιτήματος.

Μέθοδοι και γεγονότα του XMLHttpRequest

- **abort()**: Απορρίπτει το xmlhttp αίτημα.
- **getAllResponseHeaders()**: Επιστρέφει τις τιμές όλων των επικεφαλίδων σαν αλφαριθμητικό.
- **getResponseHeader(header)**: Επιστρέφει την τιμή της συγκεκριμένης επικεφαλίδας.
- **open(Method,URL,Async)**: Δίνει τιμές σε παραμέτρους τις επικοινωνίας όπου θα χρησιμοποιηθεί το αντικείμενο XMLHttpRequest.
- **send(postData)**: Στέλνει το HTTP αίτημα στον server.
- **setRequestHeader(header,value)**: Προσθέτει επιπλέον HTTP επικεφαλίδες στο αίτημα.
- **onreadystatechange**: καθορίζει τον χειριστή γεγονότων που θα κληθεί όταν αλλάξει η κατάσταση ready state.

Καταστάσεις ready state

- 0 (Uninitialized)
The object has been created, but not initialized (the **open** method has not been called).
- 1 (Open)
The object has been created, but the **send** method has not been called.
- 2 (Sent)
The **send** method has been called. **responseText** is not available.
- 3 (Receiving)
Some data has been received. **responseText** is not available.
- 4 (Loaded)
All the data has been received. **responseText** is available. responseBody is available.

Αρχικοποίηση

```
var xmlhttp;  
if (window.XMLHttpRequest){  
    // κώδικας για IE7+, Firefox, Chrome, Opera, Safari  
    xmlhttp=new XMLHttpRequest();  
}
```



```

else if (window.ActiveXObject){ // κώδικας για IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
else { alert("Your browser does not support XMLHTTP!");
}

```

ένα πρώτο απλό παράδειγμα:

```

...
xmlhttp.open("GET","simple1_.htm",true);
xmlhttp.onreadystatechange=function() { rstatech(xmlhttp) };
xmlhttp.send(null);
}

function rstatech(xmlhttpobj){
    alert(xmlhttpobj.readyState);

    if(xmlhttpobj.readyState==4)
        alert(xmlhttpobj.responseText);
}

```

παράδειγμα δεύτερο:

Παράδειγμα χρήσης Ajax για παροχή εισηγήσεων στον χρήστη

Όνομα χρήστη:

Εεκινήστε να γράψετε το όνομα στο παραπάνω πεδίο.

fanis
fani

```

<div id="suggestionel" style="display:none; position:absolute; margin-top:
2px; border:1px solid black; padding: 1px; background-color: white">DUMMY
TEXT</div>
<p>Παράδειγμα χρήσης Ajax για παροχή εισηγήσεων στον χρήστη </p>
<p>Όνομα χρήστη: <input type="Text" name="inptxt" onkeyup="ajaxcall(this)"
onblur="document.getElementById('suggestionel').style.display='none'">
</p>
<p> Εεκινήστε να γράφετε το όνομα στο παραπάνω πεδίο. </p>

```

παράδειγμα δεύτερο-JavaScript:

```

function ajaxcall(par){
...init ...
    var qstr="?name="+par.value;
    var url_="example2.php"+qstr;
    xmlhttp.open("GET",url_,true);
    xmlhttp.onreadystatechange=function() { rstatech(xmlhttp, par) };
    xmlhttp.send(null);
}

```

```

}

function rstatech(Xobj,par1){
  if(Xobj.readyState==4){
    var _suggel = document.getElementById("suggestionel");
    _suggel.style.top = par1.offsetTop + 20 + "px";
    _suggel.style.left = par1.offsetLeft + "px";
    _suggel.innerHTML=Xobj.responseText;
    _suggel.style.display='block';
  }
}

```

παράδειγμα τρίτο - XML Response:

```

<?
header('Content-type: text/xml; charset=ISO-8859-7');
$nameear = array( 'fanis', 'vasilis', 'danai', 'flora', 'fani', 'denny' );

$name = iconv("UTF-8", "ISO-8859-7", $_GET['name']);
echo "<?xml version=\"1.0\" encoding=\"ISO-8859-7\"?>\n";
echo "<rootel>";
foreach($nameear as $si){
  if(strpos($si,$name)===0)
    echo "<suggname>".$si."</suggname>";
}
echo "</rootel>";
?>

```

παράδειγμα POST - Javascript:

```

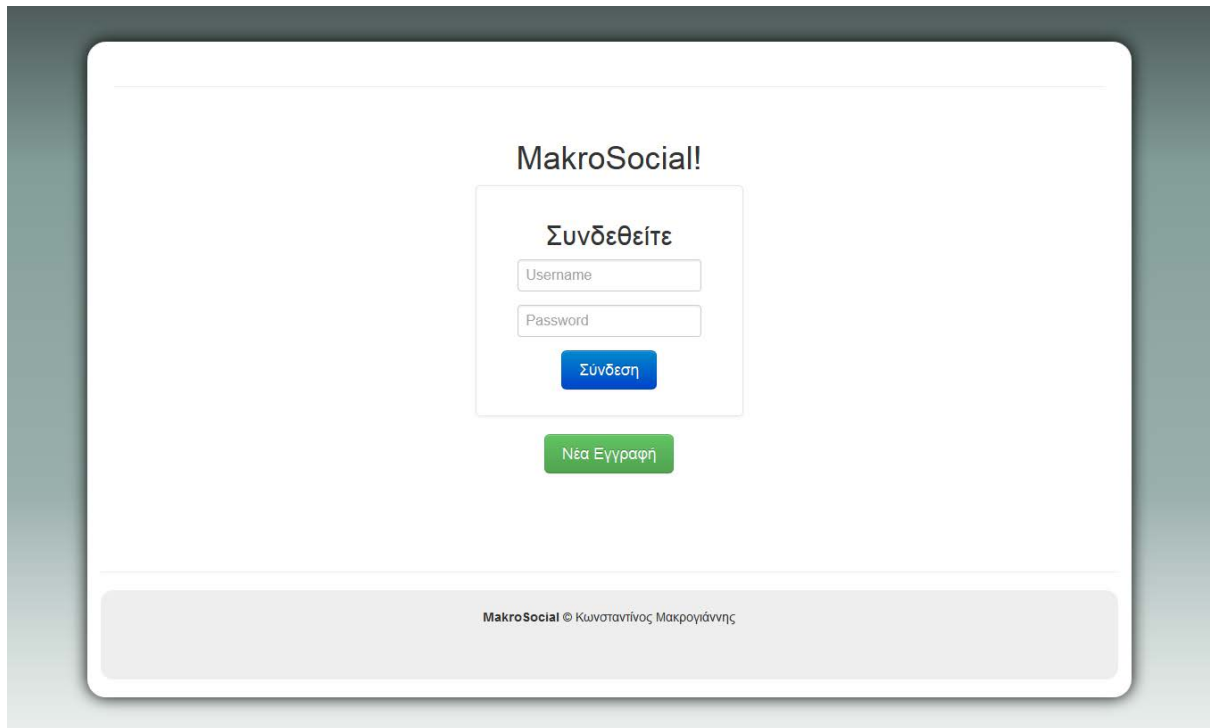
function ajaxcall(par){
... init ...
  var qstr="name="+par.value;
  var url_="example4.php";
  xmlhttp.open("POST",url_,true);

xmlhttp.setRequestHeader('Content-Type','application/x-www-form-urlencoded;
charset=ISO-8859-7');
  xmlhttp.setRequestHeader('Content-length', qstr.length);
  xmlhttp.onreadystatechange=function() { rstatech(xmlhttp, par) };
  xmlhttp.send(qstr);
}

```

MakroSocial Application

Η εφαρμογή MakroSocial είναι μία cloud εφαρμογή όπου η χρήση της γίνεται μέσα από έναν οποιοδήποτε web browser και δεν απαιτείται η εγκατάσταση κανενός άλλου εργαλείου ή εφαρμογής στον υπολογιστή μας για να έχουμε πρόσβαση σε αυτήν. Πρόκειται για μια εφαρμογή SaaS στην οποία μπορούν να δημιουργήσουν ασφαλές λογαριασμό και να διαμοιράζονται τα αρχεία τους χρήστες από οποιαδήποτε τοποθεσία.



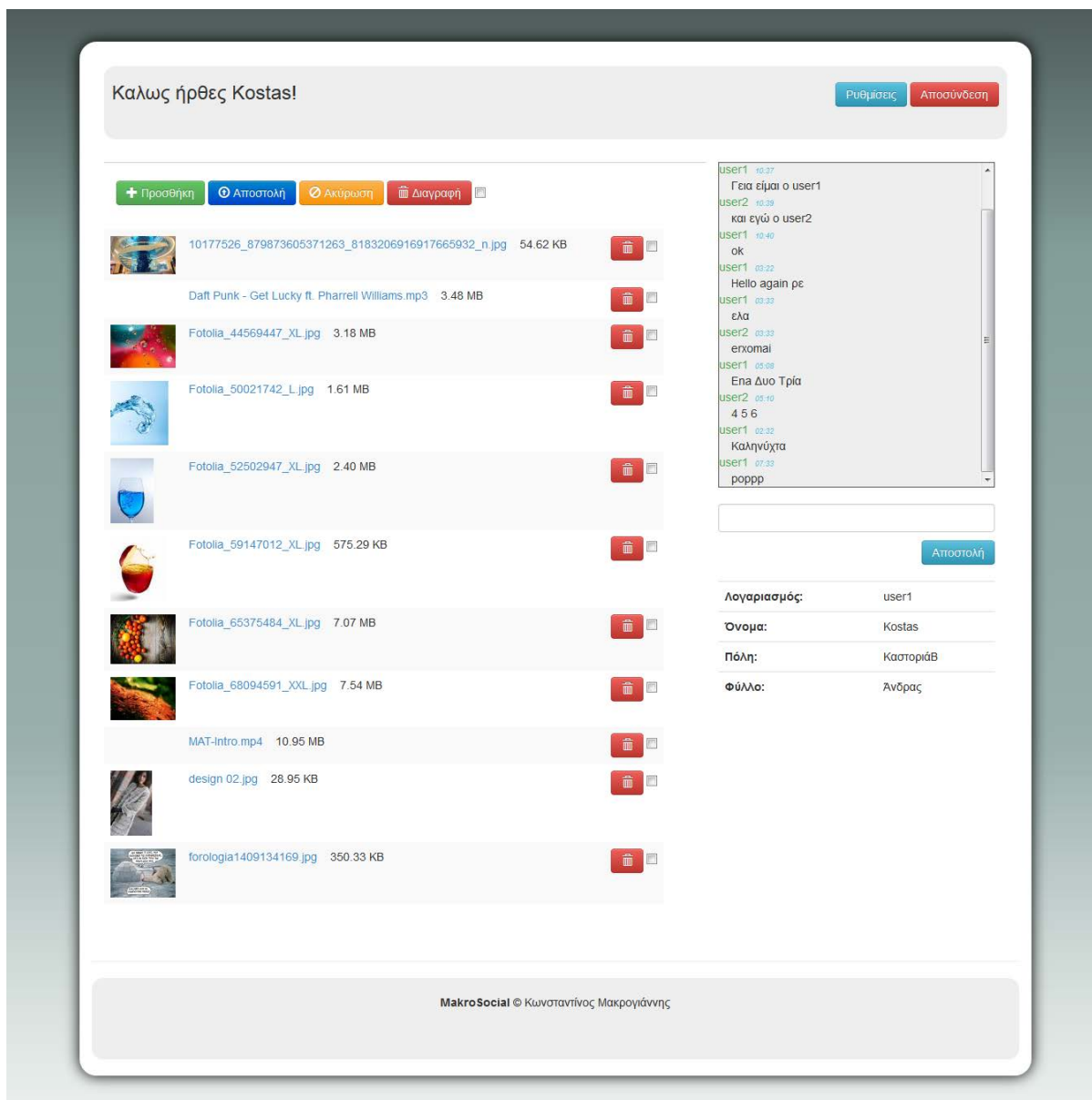
η αρχική σελίδα της εφαρμογής

Σκοπός

Δημιουργήθηκε με σκοπό την αποθήκευση, διαχείριση και επεξεργασία δεδομένων, όπως φωτογραφίες, βίντεο και αρχεία ήχου από τους χρήστες του καθώς και για τη επικοινωνία μεταξύ αυτών σε πραγματικό χρόνο. Τα δεδομένα μένουν αποθηκευμένα στον server ακόμα και αν αποσυνδεθούμε. Έτσι είναι διαθέσιμα για όλους οποιαδήποτε στιγμή και αν συνδεθούμε ξανά στην εφαρμογή. Στόχος αυτής της λειτουργίας είναι ο συντονισμός των πόρων και αρχείων και η επικοινωνία μεταξύ μελών της ίδιας κοινότητας, όπως μια εταιρεία, ένα γραφείο, μια ομάδα κοινού ενδιαφέροντος.

Λειτουργία

Ο κάθε χρήστης, αφού κάνει εγγραφή και δημιουργήσει ένα νέο προσωπικό λογαριασμό στη βάση δεδομένων της εφαρμογής μέσω της αρχικής σελίδας, μπορεί μέσω του κεντρικού panel (dashboard) να διαχειριστεί τα αρχεία που είναι ήδη αποθηκευμένα στον server από την υπόλοιπη ομάδα, να ανεβάσει καινούρια αρχεία ή να διαγράψει όποια δε χρειάζονται. Από την εφαρμογή υποστηρίζονται όλοι οι τύποι αρχείων και το ανέβασμά τους γίνεται είτε πατώντας το κουμπί “Προσθήκη”, όπου στη συνέχεια αναδύεται ένα παράθυρο για την επιλογή του αρχείου, είτε μέσω drag-n-drop λειτουργίας.



η κεντρική σελίδα της εφαρμογής

Ο χρόνος που απομένει για την ολοκλήρωση του upload φαίνεται στη μπάρα προόδου (progress-bar) που εμφανίζεται όταν πραγματοποιείται μία τέτοια διαδικασία. Υπάρχει φυσικά η δυνατότητα ακύρωσης μέσω του κουμπιού “Ακύρωση”. Όσον αφορά τη διαγραφή αρχείου/αρχείων αυτή επιτυγχάνεται είτε διαγράφοντας κάθε αρχείο μεμονομένα είτε με μαζική εκτέλεση επιλέγοντας ποια αρχεία επιθυμούμε να διαγράψουμε και πατώντας το κουμπί “Διαγραφή”. Η προσθήκη επεξεργασία και διαγραφή των αρχείων γίνεται με AJAX λειτουργία.

Εμφανή στην κεντρική σελίδα είναι και τα στοιχεία του συνδεδεμένου χρήστη, τα οποία μπορεί ανά πάσα στιγμή να τα επεξεργαστεί από την επιλογή “Settings”.

Καλως ήρθες Kostas!

[Ρυθμίσεις](#)
[Αποσύνδεση](#)

+ Προσθήκη
⏏ Αποστολή
⏏ Ακύρωση
🗑 Διαγραφή

8.24 Mbit/s | 00:08:05 | 20.46 % |
128.52 MB / 628.08 MB

10177526_879873605371263_8183206916917665932_n.jpg
54.62 KB
🗑

Daft Punk - Get Lucky ft. Pharrell Williams.mp3
3.48 MB
🗑

Fotolia_44569447_XL.jpg
3.18 MB
🗑

Fotolia_50021742_L.jpg
1.61 MB
🗑

Fotolia_52502947_XL.jpg
2.40 MB
🗑

Fotolia_59147012_XL.jpg
575.29 KB
🗑

Fotolia_65375484_XL.jpg
7.07 MB
🗑

Fotolia_68094591_XXL.jpg
7.54 MB
🗑

MAT-intro.mp4
10.95 MB
🗑

design_02.jpg
28.95 KB
🗑

forologia1409134169.jpg
350.33 KB
🗑

MOV_0104.mp4
628.08 MB

⏏ Αποστολή
⏏ Ακύρωση

user1 10:37
Για είμαι ο user1

user2 10:39
και εγώ ο user2

user1 10:40
ok

user1 03:22
Hello again ρε

user1 03:33
ελα

user2 03:29
erxomai

user1 03:08
Ena Διο Τρία

user2 03:10
4 5 6

user1 02:32
Καληνύχτα

user1 07:33
porpp

Αποστολή

Λογαριασμός: user1

Όνομα: Kostas

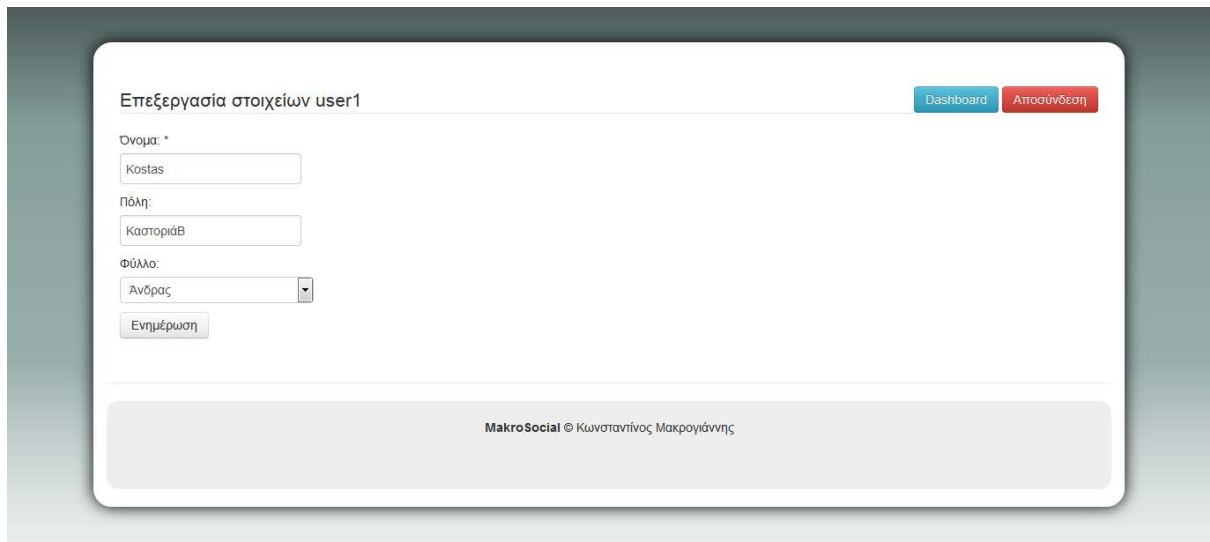
Πόλη: ΚαστοριάB

Φύλλο: Άνδρας

MakroSocial © Κωνσταντίνος Μακρογιάννης

progress-bar

Κατά το ανέβασμα αρχείων στο server προσφέρεται προεπισκόπηση του αρχείου προς ανέβασμα αν το αρχείο είναι εικόνα, ηχητικό αρχείο ή κάποιο βίντεο. Η προεπισκόπηση προσφέρεται μόνο κατά το ανέβασμα για τα αρχεία ήχου και βίντεο. Αυτό αποτελεί ένα μειονέκτημα του jQuery File Upload Plugin και ήδη βρίσκεται σε συζήτηση με την ομάδα BlueIMP, η ανάπτυξη λειτουργίας που θα προσφέρει προεπισκόπηση ήχου και βίντεο από το server της εφαρμογής προς τον χρήστη για τα αντίστοιχα αρχεία. Η ανάπτυξη λειτουργίας streaming αποτελεί επίσης ένα πεδίο που προσφέρεται προς αναβάθμιση της εφαρμογής.



σελίδα settings της εφαρμογής

Επικοινωνία σε πραγματικό χρόνο και αρχιτεκτονική

Κεντρικά στην εφαρμογή βρίσκεται και το πλαίσιο επικοινωνίας με κείμενα σε πραγματικό χρόνο μεταξύ των χρηστών. Η επικοινωνία μπορεί να γίνεται είτε μεταξύ δυο χρηστών είτε περισσότερων ακόμη και ολόκληρης της κοινότητας. Στο ανάλογο παράθυρο εμφανίζεται το όνομα του χρήστη και η χρονική στιγμή που στάλθηκε το μήνυμα. Όλα μηνύματα που γράφτηκαν και στάλθηκαν αποθηκεύονται στον server μέσα σε βάση δεδομένων με AJAX λειτουργία, μέσω μηνυμάτων JSON χωρίς να απαιτείται κάποια προσκόμιση ή εγγραφή πάνω σε txt αρχείο ή άλλα αρχεία κειμένου. Το αποτέλεσμα αυτής της λειτουργίας είναι η αποτελεσματικότητα της εφαρμογής, η ταχύτητα ανεξάρτητα του μεγέθους και του πλήθους μηνυμάτων της συνομιλίας, καθώς και η ασφάλεια των συνομιλιών εφόσον αυτή αποθηκεύεται μέσα στη βάση δεδομένων η οποία προστατεύεται από κωδικό.

Διεπαφή χρήστη - UI

Το MacroSocial είναι ένα εργαλείο που βοηθάει χρήστες που βρίσκονται στο ίδιο group ενός δικτύου να ανταλλάσσουν μηνύματα και αρχεία μεταξύ τους είτε αυτοί είναι συνδεδεμένοι μέσω σταθερού υπολογιστή/laptop, είτε μέσω tablet ή ακόμα και μέσω του smartphone τους. Η διεπαφή χρήστη (User Interface - UI), λόγω του Bootstrap framework που χρησιμοποιήθηκε είναι responsive για οποιοδήποτε μέγεθος οθόνης, που σημαίνει ότι η εφαρμογή προσαρμόζεται αυτόματα κατάλληλα σε όλες τις οθόνες για καλύτερη απόδοση, απεικόνιση χωρίς οριζόντιο scrolling και εύκολη διαχείριση. Επίσης η λειτουργία AJAX που προσφέρεται στη διαχείριση της εφαρμογής ελαχιστοποιεί τη συχνότητα ανανέωσης του παραθύρου του browser του χρήστη. Τα παραπάνω εκφράζουν τη φιλικότητα της εφαρμογής προς τον χρήστη.

Ανάλυση κώδικα

index.php

Περιγραφή: Αυτή είναι η αρχική σελίδα της εφαρμογής.

Λειτουργία:

```
include ('header.php'); ?>
```

Εισαγωγή του header.php.

```
if($message == 1)
```

```
else if($message == 2)
```

Εδώ εμφανίζονται τα alerts, όπως λάθος όνομα χρήστη ή κωδικός καθώς και όταν γίνει αποσύνδεση.

```
<form action="auth_check.php" method="post" class="form-signin">
```

Φόρμα για τη σύνδεση στην εφαρμογή δίνοντας το username και το password.

```
<a class="btn btn-large btn-success" href="register.php">Νέα Εγγραφή</a>
```

Κουμπί για δημιουργία χρήστη – Register.

```
<?php include ('footer.php'); ?>
```

Εισαγωγή του footer.php.

header.php

Περιγραφή: Κοινό Head section για όλα τα αρχεία HTML της εφαρμογής.

Λειτουργία:

```
if (basename($_SERVER['REQUEST_URI']) == 'dashboard.php') {
```

Φορτώνουμε την πλατφόρμα επικοινωνίας μόνο αν είμαστε στο dashboard.

```
echo '<body onload="javascript:startChat();">';
```

Με το που φορτώσουμε το header.php, αν είμαστε στο dashboard, καλείτε η `startChat()`;

```
echo '<body>';
```

αλλιώς όχι.

footer.php

Λειτουργία: Το footer της εφαρμογής.

database.php

Περιγραφή: Κώδικας σύνδεσης στη βάση και αναφορά σφαλμάτων από osCommerce Platform - (Opensource).

Λειτουργία:

error_reporting(E_ERROR | E_WARNING | E_PARSE);

Αναφορά όλων των PHP σφαλμάτων.

function db_connect

Σύνδεση στη βάση.

auth_check.php

Περιγραφή: Το αρχείο αυτό ελέγχει αν ο χρήστης που πάει να κάνει login υπάρχει στη βάση.

Λειτουργία:

include ('database.php');

Παίρνει τα login credentials απο τη login form, όπως το username και το password (το password είναι σε md5 κωδικοποίηση).

\$sql="SELECT * FROM users WHERE username='\$username' and password='\$password';"

Αναζητεί στη βάση αν υπάρχει χρήστης με το username και password που δώσαμε.

\$result=mysql_query(\$sql);

Επιστρέφει το αποτέλεσμα της αναζήτησης.

\$count=mysql_num_rows(\$result);

Επιστρέφει έναν αριθμό σύμφωνα με την παραπάνω αναζήτηση.

if(\$count==1)

Αν το count ισούται με 1 τότε έχουμε match.

register.php

Περιγραφή: Αρχείο εγγραφής νέου χρήστη στη βάση.

Λειτουργία:

```
if($register == 1 && !empty($_POST))
```

Έλεγχος αν η μεταβλητή post είναι άδεια και πατήσουμε ενημέρωση. Έπειτα γίνεται αρχικοποίηση όλων των μεταβλητών.

```
$username = strip_tags($username);
```

Αποφεύγουμε sql injection.

```
$username = str_replace(' ', '', $username);
```

Γίνεται αφαίρεση όλων των κενών.

```
$password = md5($password);
```

Κωδικοποίηση md5 password.

```
$sql="SELECT id FROM users WHERE username='$username'";
```

Γίνεται έλεγχος του password.

```
if($num_rows > 0)
```

Ελέγχεται αν υπάρχει ο χρήστης. Αν ναι τότε εμφανίζει ανάλογο μήνυμα.

```
else
```

Αν δεν υπάρχει τότε τοποθετούμε τον καινούριο χρήστη στη βάση.

```
<script type="text/javascript">
```

Γίνεται το validation των εισαγόμενων τιμών και ελέγχονται τα κενά.

```
<?php include ('footer.php'); ?>
```

Εισαγωγή του footer.php.

header_dashboard.php

Περιγραφή: Αυτό το αρχείο φορτώνεται στο <head> μόνο όταν είμαστε στο Dashboard.

Λειτουργία:

```
function startChat()
```

Καλείται όταν μπαίνουμε στο dashboard.php (onload στο <body> μέσω του header.php).

```
function getXmlHttpRequestObject()
```

Δημιουργία XmlHttpRequest Object.

```
function takeChatText()
```

Πάρε τα υπάρχοντα μηνύματα από τον server.

```
function sendChatText()
```

Πρόσθεσε το παρόν μήνυμα στον server.

```
param += '&name=<?php echo $_SESSION['username'];?>';
```

Παίρνουμε το username του αποστολέα του μηνύματος από τη συνεδρία.

```
function handleSendChat()
```

Αφού σταλεί το μήνυμα ανανεώνεται το chatbox.

```
function handleReceiveChat()
```

Παίρνει την XML από το contactChat.php και εμφανίζει το μήνυμα στην οθόνη μαζί με τον αποστολέα και τη χρονική στιγμή που στάλθηκε το μήνυμα. Πραγματοποιείται για όλα τα υπάρχοντα μηνύματα.

```
mTimer = setTimeout('takeChatText();',2000);
```

Ανανέωση 2 δευτερολέπτων, κάθε μήνυμα θα εμφανιστεί τουλάχιστον μέσα σε 2 δευτερόλεπτα σε όλους τους συνδεδεμένους χρήστες.

```
function blockSubmit()
```

Με το πάτημα του Enter στέλνεται το μήνυμα και στον server.

dashboard.php

Περιγραφή: Κεντρικό UI της εφαρμογής

Λειτουργία:

```
session_start();
```

```
if(!isset($_SESSION['username']))
```

Έλεγχος αν ο χρήστης είναι συνδεδεμένος.

```
header("location:index.php");
```

Αν δεν είναι τον στέλνει στην αρχική σελίδα.

```
else {
```

```
    $username = $_SESSION['username'];
```

Αν είναι τότε παίρνει από τη συνεδρία το username.

```
include ('header.php');
```

Εισαγωγή του header.

```
<?php
```

```
$document_get = mysql_query("SELECT * FROM users WHERE username='$username'");
```

```
$match_value = mysql_fetch_array($document_get);
```

```
$fullname = $match_value['fullname'];
```

```
$location = $match_value['location'];
```

```
$gender = $match_value['gender'];
```

```
?>
```

Λειτουργία των παραπάνω: Προσκομίζονται τα στοιχεία του χρήστη από τη βάση.

`<div class="row fileupload-buttonbar">`

Το fileupload-buttonbar περιέχει κουμπιά για την προσθήκη/διαγραφή αρχείων καθώς και για το ξεκίνημα/τελείωμα ανεβάσματος αρχείων.

`<input type="file" name="files[]" multiple>`

Όταν επιλέγουμε αρχεία που θέλουμε να ανεβάσουμε τα βάζουμε όλα σε πίνακα.

`<table role="presentation" class="row table table-striped">`

Ο πίνακας με όλα τα αρχεία που είναι διαθέσιμα για ανέβασμα/κατέβασμα.

`<div id="blueimp-gallery" class="blueimp-gallery blueimp-gallery-controls" data-filter=":even">`

Το Blueimp Image Gallery widget / Gallery Slideshow για τις εικόνες.

`<div id="MakroChat">`

Το interface του Chat.

`<script id="template-upload" type="text/x-tmpl">`

Το συγκεκριμένο script εμφανίζει το preview, το upload κουμπί και το κουμπί cancel μέσα στο table των επιλεγμένων αρχείων για ανέβασμα.

`<script id="template-download" type="text/x-tmpl">`

Το script αυτό εμφανίζει το preview, το download link, το κουμπί διαγραφής και το checkbox για μαζική διαγραφή μέσα στο table των διαθέσιμων αρχείων για κατέβασμα.

`<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script >`

`<script src="assets/js/vendor/jquery.ui.widget.js"></script>`

....

`<script src="assets/js/main.js"></script>`

Τα παραπάνω scripts είναι απαραίτητα για το Blueimp.

`<script type="text/javascript">`

Αυτόματα ανανεώνεται το interface των αρχείων κάθε 12 λεπτά για να βλέπουν όλοι οι χρήστες τα ανεβασμένα αρχεία χωρίς να κάνουν refresh.

`<?php include ('footer.php'); ?>`

Εισαγωγή του footer.

settings.php

Περιγραφή: Χρησιμοποιείται για την αλλαγή των στοιχείων του χρήστη στον server.

Λειτουργία:

`session_start();`

`if(!isset($_SESSION['username']))`

Κώδικας ελέγχου session χρήστη.

```
if($update == 1 && !empty($_POST))
```

Έλεγχος αν η μεταβλητή post είναι άδεια και έχουμε πατήσει ενημέρωση, εμφανίζοντας τα ανάλογα μηνύματα.

contactChat.php

Περιγραφή: Ρωτάει τη βάση για υπάρχοντα μηνύματα chat και τα μετατρέπει σε μορφή xml για να καταχωρηθούν στο παράθυρο της εφαρμογής JSON.

Λειτουργία:

```
header("Expires: Mon, 26 Jul 2014 03:00:00 GMT" );
header("Last-Modified: " . gmdate( "D, d M Y H:i:s" ) . "GMT" );
header("Cache-Control: no-cache, must-revalidate" );
header("Pragma: no-cache" );
header("Content-Type: text/xml; charset=utf-8");
```

Λειτουργία των παραπάνω: Αποφυγή caching της σελίδας και εξαναγκασμός αναζήτησης μηνυμάτων από τον server κάθε φορά που ανοίγει η σελίδα.

```
if(isset($_POST['message']) && $_POST['message'] != "")
```

Ελέγχει αν έχει σταλεί κάποιο μήνυμα από το chat για να μπει στον server.

```
$xml = '<?xml version="1.0" ?><root>';
XML output.
```

```
if(!isset($_GET['chat']))
```

Έλεγχος συνεδρίας

```
while($message_array = db_fetch_array($message_query))
```

Βάζουμε το αποτέλεσμα σε XML (αυτό γίνεται για κάθε μήνυμα)

```
$xml .= '<message id="' . $message_array['message_id'] . '">';
Το ID του μηνύματος.
$xml .= '<user>' . htmlspecialchars($message_array['user_name']) . '</user>';
Ο αποστολέας του μηνύματος.
$xml .= '<text>' . htmlspecialchars($message_array['message']) . '</text>';
Το περιεχόμενο του μηνύματος.
$xml .= '<text>' .
        htmlspecialchars($message_array['message']) . '</text>';
Η χρονική στιγμή αποστολής του μηνύματος.
```

logout.php

Περιγραφή: Εκκαθάριση συνεδρίας και επιστροφή στην αρχική και το ανάλογο μήνυμα.

makros-diploma.sql

Περιγραφή: Στιγμιότυπο της βάσης δεδομένων.

Λειτουργία:

```
CREATE DATABASE IF NOT EXISTS `makros-diploma` DEFAULT CHARACTER SET utf8  
COLLATE utf8_unicode_ci;  
USE `makros-diploma`;
```

Δημιουργία της βάσης δεδομένων.

```
CREATE TABLE IF NOT EXISTS `chat`
```

Δημιουργία του πίνακα chat.

```
CREATE TABLE IF NOT EXISTS `message`
```

Δημιουργία του πίνακα message.

```
INSERT INTO `message` (`message_id`, `chat_id`, `user_id`, `user_name`,  
`message`, `post_time`) VALUES
```

Εισαγωγή στοιχείων στον πίνακα message.

```
CREATE TABLE IF NOT EXISTS `users`
```

Δημιουργία του πίνακα users.

```
INSERT INTO `users` (`id`, `username`, `password`, `fullname`, `location`,  
`gender`) VALUES
```

Εισαγωγή στοιχείων στον πίνακα users.

References

[1]. XAMPP. <http://el.wikipedia.org/wiki/XAMPP>

[2]. Bootstrap. <http://el.wikipedia.org/wiki/Bootstrap>

[3]. Bootstrap - Global CSS settings, fundamental HTML elements styled and enhanced with extensive classes and an advanced grid system. <http://getbootstrap.com/css/>

[4]. JQuery - Setup instructions. <https://github.com/blueimp/jQuery-File-Upload/wiki/Setup>

[5]. JQuery File Upload - Basic Plus UI version. <http://blueimp.github.io/jQuery-File-Upload/>