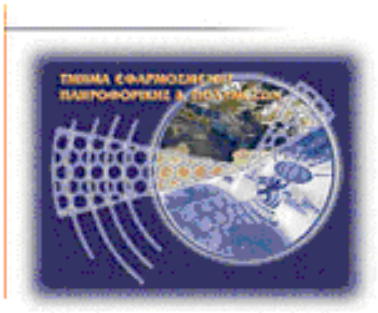
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.



Πτυχιακή Εργασία

Ενιαίο σύστημα διαχείρισης διαφορετικών ιστοσελίδων

Παππάς Εμμανουήλ Α.Μ. 1723
email: mano.pappa@gmail.com

Επιβλέποντες Καθηγητές:

Δρ. Μαλάμος Αθανάσιος
email: amalamos@epp.teicrete.gr

Καπετανάκης Κωνσταντίνος
email: kapekost@epp.teicrete.gr

Ηράκλειο Αύγουστος 2013

Υπεύθυνη Δήλωση: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής του Τ.Ε.Ι. Κρήτης.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Ηράκλειο, …./…./2013

Επιτροπή Αξιολόγησης

1.
2.
3.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1ο : ΕΙΣΑΓΩΓΗ

1.1 Σκοπός πτυχιακής εργασίας

Ο σκοπός και το κίνητρο για τη δημιουργία της παρούσας πτυχιακής είναι η προσπάθεια να γίνει ευκολότερη και συγχρόνως αποδοτικότερη η δουλειά ενός web developers. Η κοινότητα των web developers / designers κι οποιουδήποτε ασχολείται με το διαδίκτυο σε παραγωγικό επίπεδο, εξελίσσεται με ταχύτατους ρυθμούς εδώ και πολλά χρόνια. Μεγάλη είναι επίσης και η χρήση CMS εργαλείων από τους developers με δημοφιλέστερα το Joomla και το Wordpress.

Στα πλαίσια της πτυχιακής αναπτύχθηκε μία εφαρμογή σε Ruby on Rails προσπαθώντας να επωφεληθούμε από τα θετικά χαρακτηριστικά και τις ιδιαιτερότητες της γλώσσας προγραμματισμού Ruby και την άνεση του framework που είναι βασισμένο σε αυτή το Ruby on Rails. Η εφαρμογή θα είναι συμβατή με οποιαδήποτε πλατφόρμα διαχείρισης περιεχομένου CMS.

Μέσω αυτής της εφαρμογής παρέχεται στο χρήστη της η δυνατότητα συνδεθεί σε οποιοδήποτε εγκατεστημένο CMS μέσω ενός ενιαίου περιβάλλοντος διαχείρισης, θέλοντας να διευκολύνουμε έτσι τον administrator της εκάστοτε CMS ιστοσελίδας. Ο διαχειριστής θα χρησιμοποιεί έναν μόνον κωδικό για να συνδεθεί στην εφαρμογή και να αποθηκεύσει εκεί τα συστήματα που θέλει να διαχειρίζεται με κύριο σκοπό να συνδέεται με ένα κωδικό σε όσα projects επιθυμεί.

Ταυτόχρονα θα μπορεί μέσω της εφαρμογής να ορίζει εκκρεμότητες με ημερομηνία έναρξης και λήξης οι οποίες υπάρχει η δυνατότητα να δημιουργούνται και στο Google Calendar αυτόματα, ενώ το σύστημα θα είναι υπεύθυνο για την υπενθύμιση της εκκρεμότητας που πλησιάζει προς τη λήξη της. Επίσης θα μπορεί να κρατάει γενικές σημειώσεις οι οποίες αναρτώνται σε μορφή δημοσιεύσεων όπως σε κάποια κοινωνικά μέσα δικτύωσης τις οποίες υπάρχει η δυνατότητα να τις βλέπουν κι άλλοι χρήστες στη περίπτωση μιας εταιρίας για παράδειγμα.

Είναι λοιπόν σαφές ότι η εφαρμογή μπορεί να διευκολύνει τους διαχειριστές είτε οργανώνοντας τη μέρα τους είτε διαφυλάσσοντας ασφαλείς τους κωδικούς των backend των ιστοσελίδων που διαχειρίζονται.

1.2 Περιγραφή πτυχιακής εργασίας

Δημιουργία εφαρμογής συμβατή με οποιαδήποτε πλατφόρμα διαχείρισης περιεχομένου, η οποία θα παρέχει στον administrator τη δυνατότητα να συνδέεται σε οποιαδήποτε εγκατεστημένη πλατφόρμα διαχείρισης περιεχομένου από ένα ενιαίο περιβάλλον διαχείρισης.

Η πρώτη προσέγγιση θα γίνει απομονώνοντας το περιβάλλον διαχείρισης κάποιων website στημένα με κοινή πλατφόρμα, δημιουργώντας ένα ενιαίο περιβάλλον διαχείρισης. Στη συνέχεια θα δημιουργηθεί ένα υποσύστημα φιλοξενίας οποιουδήποτε περιβάλλοντος εργασίας. Ο διαχειριστής θα μπορεί να αποθηκεύσει τα συστήματα που θέλει να διαχειρίζεται χρησιμοποιώντας μόνο έναν κωδικό και στη συνέχεια με την επιλογή του μπορεί να αποθηκεύει στο σύστημα τους κωδικούς των site που διαχειρίζεται.

Στην κεντρική σελίδα του συστήματος θα υπάρχει ένα υποσύστημα που θα τον ενημερώνει για εκκρεμότητες, και θα μπορεί να ορίζει ημερομηνίες έναρξης και λήξης. Όταν πλησιάζει το τέλος μίας ορισμένης εκκρεμότητας, το σύστημα κατά την εκκίνησή του, θα τον ενημερώνει ιδιαίτερα, και θα του δίνει την επιλογή να στείλει μία σημείωση στο email του, σαν υπενθύμιση. Το περιβάλλον εργασίας του συστήματος θα υλοποιηθεί χρησιμοποιώντας τεχνολογίες όπως HMTL5, CSS3, JavaScript, Apache web server, Mysql server.

Abstract

My diploma thesis is about the creation of an application, compatible with any CMS platform. It will provide the administrator with the ability to connect to any already installed CMS project from a single management interface.

Initially, we will isolate the management environment of some websites, built with a given platform, creating that single management environment. Secondly, a hosting sub-system will be created of any framework environment. Administrators will be able to store the systems they wish to operate using a single password and then they will be able to choose to store the passwords of the sites they are working on, within the system.

There will be a sub-system, in the central page of the system, that will inform administrators on any outstanding issues, and will allow them to determine start and end dates. When the deadline for a certain issue is approaching, the system will alert them upon running, giving them the option to send a reminder-note to their email account. The work-environment of the system will be delivered with the use of technologies, such as HMTL5, CSS3, JavaScript, Apache web server, and Mysql server.

ΚΕΦΑΛΑΙΟ 2ο: ΕΡΓΑΛΕΙΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ

2.1 Η γλώσσα προγραμματισμού Ruby

Η Ruby είναι μία δυναμική γλώσσα προγραμματισμού ανοιχτού κώδικα καθώς και αντικειμενοστραφής. Αναπτύχθηκε και σχεδιάστηκε στα μέσα της δεκαετίας του 90' αρχικά από τον Yukihiro Matsumoto ή αλλιώς "Matz"ο οποίος το προόριζε για προσωπική του χρήση. Έχει σαφείς επιρροές από τις γλώσσες Perl, Smalltalk, Eiffel και Lisp.

2.1.1 Ιστορική αναδρομή της Ruby

Η Ruby είναι μια δυναμική, ανοιχτού κώδικα, αντικειμενοστραφής γλώσσα προγραμματισμού η οποία άρχισε να αναπτύσσεται το 1993 στην Ιαπωνία από τον Yukihiro Matsumoto γνωστού και ως "Matz"ο οποίος το προόριζε για προσωπική του χρήση. Παρόλο που έχει τη ίδια σχεδόν ηλικία με τις γλώσσες Perl και Python θεωρείται από πολλούς ως μία νέα γλώσσα στον κόσμο των γλωσσών προγραμματισμού. Το βασικό κίνητρο του δημιουργού της, κατά τη δημιουργία της γλώσσας ήταν η αποφυγή δυσάρεστων εκπλήξεων καθώς επίσης και μία πιο ευχάριστη και όσο το δυνατόν πιο προσιτή προς τον προγραμματιστή, συγγραφή κώδικα, με σκοπό τη βελτίωση της παραγωγικότητας των προγραμματιστών.

Είπε κάποτε ο Yukihiro Matsumoto :
"Ήθελα να μειώσω όσο το δυνατόν περισσότερο την απογοήτευσή μου όταν προγραμμάτιζα, ήθελα δηλαδή να μειώσω την δυσκολία μιας γλώσσας. Αυτός ήταν ο αρχικός μου στόχος όταν σχεδίαζα την Ruby. Ήθελα ο χρόνος που αφιέρωνα όταν προγραμμάτιζα να είναι στιγμές διασκέδασης".

Το όνομα "Ruby" αποφασίστηκε κατά την διάρκεια μιας on-line συνομιλίας με τον Keitzu Ishitsuka στης 24 Φεβρουαρίου του 1993, πριν ακόμη ξεκινήσει να γράφεται ο οποιοσδήποτε κώδικας για την ανάπτυξη της Ruby. Αρχικά προτάθηκαν δύο ονόματα, το Coral και το Ruby. Το όνομα Coral είχε ήδη μεγάλη ιστορία ως όνομα μιας γλώσσας προγραμματισμού που δημιουργήθηκε το 1964 στο Royal Radar Establishment στο Ηνωμένο Βασίλειο. Έτσι τελικά ο "Matz" επέλεξε το "Ruby". Αργότερα ο "Matz" δήλωσε πως ο κύριος παράγοντας για την επιλογή του ονόματος Ruby ήταν λόγω του ότι ένα ρουμπίνι ήταν το birthstone (το birthstone είναι ένας πολύτιμος λίθος που δίνεται σαν δώρο και συμβολίζει στο Γρηγοριανό ημερολόγιο τον μήνα που γεννήθηκε κάποιος) από κάποιον συνάδελφό του.

Πιστεύοντας λοιπόν ο Matz ότι το ιδανικό μοντέλο που έπρεπε να υιοθετήσει, κι έχοντας μεγάλη εμπειρία σε αυτό, ήταν ο αντικειμενοστραφής προγραμματισμός. Το γεγονός ότι χρησιμοποίησε την αντικειμενοστρέφεια ως θεμέλιο για ολόκληρη τη γλώσσα είναι μία από τις διαφορές της από άλλες

γλώσσες όπως η Perl. Θα έπρεπε λοιπόν τα πάντα να είναι αντικείμενα (objects) και οι μέθοδοι (methods) να αντικαταστήσουν το ρόλο των διαδικασιών που χρησιμοποιούνται σε διαδικαστικές (procedures) γλώσσες (procedural languages) από τους προγραμματιστές. Είπε ο Matsumoto σε μία από τις δηλώσεις του (Συνέντευξη 2001) *"Ήθελα μια γλώσσα που να ήταν πιο δυνατή από την Perl και πιο αντικειμενοστραφής από την Python, γι' αυτό αποφάσισα να φτιάξω την δικιά μου γλώσσα"*.

Τον Δεκέμβριο του 1995, ο Matsumoto δημοσίευσε την πρώτη δημόσια άλφα έκδοση (public alpha version) της Ruby. Μετά από αυτό δεν άργησε να σχηματιστεί μια κοινότητα στην Ιαπωνία. Παρόλο το ότι στην Ιαπωνία κατάφερε να γίνει γρήγορα δημοφιλής χρειάστηκε αγώνα για να καταφέρει το ίδιο και στον υπόλοιπο κόσμο.

Το 1996 πλέον λαμβάνει χώρα η πιο ουσιαστική ανάπτυξη με τη Ruby να αποκτάει σιγά σιγά περισσότερους οπαδούς και ταυτόχρονα γίνεται και η συγκρότηση μίας ομάδας η οποία αποτελείται από προγραμματιστές πυρήνα (core developers) της Ruby καθώς κι από μερικούς θερμούς οπαδούς οι οποίοι αποτέλεσαν μία περισσότερο γενική κοινότητα προγραμματιστών της Ruby. Ένα χρόνο μετά την πρώτη δημόσια άλφα έκδοση (public alpha version) στις 25 Δεκεμβρίου 1996 δημοσιεύτηκε και η έκδοση 1.0 της Ruby με τους core developers να βοηθούν το δημιουργό της στην ανάπτυξη της γλώσσας με το να δημοσιεύουν patches διορθώσεων καθώς και χρήσιμες ιδέες.

Το εγχειρίδιο χρήσης της γλώσσας (documentation) ήταν γραμμένο στα Ιαπωνικά μέχρι και το 1997, απαγορεύοντας έτσι σε όλους τους μη Ιάπωνες προγραμματιστές να ασχοληθούν με τη γλώσσα, ενώ επίσημα άρχισε να προωθείται η γλώσσα στα αγγλικά στα τέλη του 1998 δημιουργώντας τη λίστα ταχυδρομείου (mailing – list) Ruby-Talk, το οποίο παραμένει μέχρι και σήμερα ως ένα από τις καλύτερες τοποθεσίες αφενός για να παρακολουθήσεις να συμμετέχεις σε συζητήσεις πολύ χρήσιμες για τη Ruby κι αφετέρου για να ζητήσεις βοήθεια για από άλλους προγραμματιστές για την επίλυση του προβλήματος σου. Ένα χρόνο μετά δημιουργήθηκε και η πρώτη επίσημη ιστοσελίδα της Ruby με domain http://www.ruby-lang.org/ η οποία την εκπροσωπεί μέχρι και σήμερα.

Παρόλη τη δύναμη που είχε η Ruby ως γλώσσα προγραμματισμού, η προώθησή της στο μεγάλο και ευρύτερο κοινό των προγραμματιστών παρέμενε σε χαμηλά επίπεδα, έως το 2004, που ήταν και η χρονιά που εκδόθηκε το Web Framework Ruby on Rails (RoR), από τον νέο σε ηλικία David Heinemeier Hansson, το οποίο κατάφερε να αλλάξει τον τρόπο που η παγκόσμια κοινότητα προγραμματιστών αντιλαμβανόταν τη γλώσσα.

Σο Μάρτιο του 2012 στο Παγκόσμιο συνέδριο του Ανοικτού Λογισμικού τιμήθηκε με το βραβείο FSM για την συνεισφορά του στο έργο κοινωνικής ωφέλειας από τον οργανισμό GNU Health.

2.1.2 Βασικές επιρροές της Ruby

Η Ruby ως γλώσσα είναι επηρεασμένη από γλώσσες με τις οποίες ο δημιουργός της ήταν ήδη αρκετά εξοικειωμένος. Γλώσσες προγραμματισμού όπως η Perl στη σύνταξη και η Smalltalk από την οποία υιοθετεί χαρακτηριστικά όπως το να δίνει μεθόδους και μεταβλητές σε όλους τους τύπους της. Άλλες βασικές επιρροές δέχεται από τις γλώσσες Eifel και Lisp.

Από τη Perl έχει κρατήσει το βασικό της αξίωμα το οποίο λέει ότι "Υπάρχουν περισσότεροι από έναν τρόποι για να κάνεις κάτι το οποίο είναι εύκολα αντιληπτό και στη Ruby. Αυτή είναι και η βασική διαφορά της Ruby με άλλες γλώσσες όπως η Python, η οποία παρέχει τυποποιημένες μεθόδους, στη Ruby ο προγραμματιστής έχει τη δυνατότητα και την ελευθερία να λύσει ένα πρόβλημα με περισσότερους από έναν τρόπους.

Από την Samlltalk η οποία είναι μία αντικειμενοστραφής γλώσσα που αναπτύχθηκε της δεκαετία του '70, η επιρροή της Ruby αφορά την αντικειμενοστραφή της φύση. Όπως στη Smalltalk έτσι και στη Ruby τα πάντα είναι αντικείμενα. Τα αντικείμενα δημιουργούνται με την κλήση ενός κατασκευαστή (constructor) και ένα ειδικό αντικείμενο (κλάση) που σχετίζεται με μια μέθοδο όπως η new().

Άλλες γλώσσες όπως η Python, η LISP, η ADA καθώς και η C++ αποτελούν βασικές επιρροές της Ruby πράγμα που δείχνει πως αποτελείται από τα καλά στοιχεία πολλών άλλων γλωσσών και στις οποίες οφείλει την δύναμη και την ευελιξία της.

Καταλήγοντας λοιπόν αναφερόμενοι στη Ruby μιλάμε για μία γλώσσα που συνδυάζει χαρακτηριστικά Συναρτησιακού, Αντικειμενοστραφή, Προστατικού και Ανακλαστικού προγραμματισμού. Είναι γλώσσα δυναμικών τύπων κι έχει αυτόματη διαχείριση μνήμης.

2.1.3 Τρόπος Σύνταξης

Η σύνταξη της Ruby είναι σχεδόν ίδια με αυτήν της Perl και της Python. Ο ορισμός των κλάσεων και των μεταβλητών γίνεται με λέξεις κλειδιά. Η διαφορά τους με την Perl είναι ότι στη Ruby οι μεταβλητές δεν είναι απαραίτητο να ξεκινούν με κάποιον ειδικό χαρακτήρα "sigil" που να ορίζει τον τύπο δεδομένων της μεταβλητής. Στην Ruby χρησιμοποιούνται για να δείξουν την εμβέλεια της μεταβλητής. Πολύ σημαντική είναι η διαφορά της Ruby σε σύγκριση με την Perl και την C είναι ότι χρησιμοποιούνται λέξεις κλειδιά για να ορίσουν την αρχή και το τέλος ενός μπλοκ κώδικα, χωρίς να χρειάζεται η χρήση αγκύλων. Καθιστώντας την έτσι, συν τις άλλοις ευανάγνωστη.

2.2 Ruby on Rails Framework

2.2.1 Ιστορική αναδρομή του Ruby on Rails

Η ύπαρξη του Ruby On Rails ξεκίνησε ως μια εφαρμογή που ονομαζόταν Basecamp. Ήταν το framework το οποίο αναπτύχθηκε από τον Δανό προγραμματιστή David Heinemeier Hansson, για την εταιρία 37signals. Λόγω της μεγάλης του επιτυχίας η εταιρία στράφηκε στην ανάπτυξη και παραγωγή εφαρμογών ενώ ο David Heinemeier Hansson έγινε συνεταίρος στην εταιρία.

Το Rails αρχικά δεν δημιουργήθηκε σαν ένα αυτόνομο framework. Ήταν η εξέλιξη μιας ήδη υπάρχουσας εφαρμογής, και θα χρησιμοποιούνταν για να την παραγωγή άλλων εφαρμογών της 37signal. Ο Hansson ξεκίνησε με την προοπτική να κάνει την δουλειά του ευκολότερη προσθέτοντας περισσότερη λειτουργικότητα, όπως ο χειρισμός βάσεων δεδομένων και η παραγωγή templates.

Έτσι γεννήθηκε η πρώτη έκδοση του Ruby on Rails. Ο Hansson αποφάσισε να εκδώσει την εφαρμογή κάτω από μια open source άδεια. Η πρώτη beta έκδοση του Rails κυκλοφόρησε τον Ιούλιο του 2004, με τις εκδόσεις 1.0 και 2.0 να ακολουθούν στις 13 Δεκεμβρίου του 2005 και στις 7 Δεκεμβρίου 2007 αντίστοιχα. Πολλές χιλιάδες προγραμματιστών από όλο τον κόσμο κατέβασε την εφαρμογή από το Internet, και το πλήθος τους συνεχώς αυξανόταν.

Η απόφαση του Hansson να κυκλοφορήσει το framework ως open source, έδωσε την δυνατότητα στο Ruby On Rails να πάρει όλα τα θετικά στοιχεία που μπορεί να προσφέρει η open source φιλοσοφία. Συνεχώς προγραμματιστές που δουλεύουν με το RoR εκδίδουν εργαλεία και διορθώνουν λάθη που βρίσκουν στον πηγαίο κώδικα. Το repository (αποθήκη) ελέγχεται από τον πυρήνα του Rails που αποτελείται από μια ομάδα έξι προγραμματιστών της οποίας ηγείται ο Hansson.

Το Ruby on Rails είναι ένα framework το οποίο επιτρέπει την γρήγορη ανάπτυξη εφαρμογών και αποτελείτε από κομμάτια κώδικα τα οποία αλληλεπιδρούν όλα μεταξύ τους με συγκεκριμένο τρόπο τα οποία αποτελούν και τα βασικά στοιχεία της.

Αναλυτική αναφορά στα βασικά στοιχεία της Ruby on Rails:

Active Record:
Είναι η βάση για κάθε μοντέλο (model) που δημιουργούμε στην εφαρμογή μιας και μας προσφέρει τις βασικές λειτουργίες όπως δημιουργία, διαγραφή, ανάγνωση και τροποποίηση καθώς και διασύνδεση-πρόσβαση στη Βάση Δεδομένων.

Active Resource:
Προσφέρει web services και φιλτράρει την επικοινωνία μεταξύ των αντικειμένων την διευκόλυνση των υπηρεσιών αυτών.

Action Pack:
Είναι το πακέτο που δημιουργεί τη σχέση ανάμεσα σε Controller και View όπως και τις απαραίτητες λειτουργίες για την διαχείριση των εκάστοτε respond-request.

Active support :
Περιέχει όλες τις βοηθητικές κλάσεις και λειτουργίες πολλές από τις οποίες είναι επεκτάσεις βασικών κλάσεων της Ruby.

Action Mailer:
Παρέχει τη δυνατότητα αξιοποίησης e-mail υπηρεσιών. Μπορούμε να στέλνουμε και να λαμβάνουμε e-mail προς στην εφαρμογή.

Active Model:
Ορίζει τη διεπαφή μεταξύ του Action Pack και του Object Relationship Mapping (ORM).

Railties :
Είναι ο πηγαίος κώδικας της rails από τον οποίο δημιουργήθηκε το πρώτο στάδιο της εφαρμογής και συνδέει τα framework και plugins μέσα σε αυτήν.

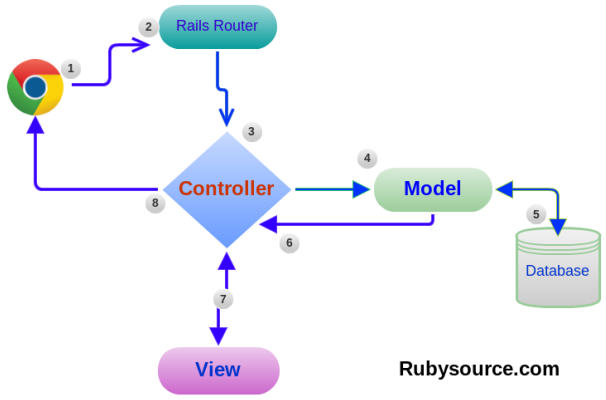
Πρέπει να αναφερθεί ότι ο κάθε χρήστης στη RoR μπορεί να δημιουργήσει τα δικά του plugins για να επεκτείνει την λειτουργία των βασικών κλάσεων αφού το Rails είναι open source Framework .

Το Rails είναι γνωστό για την εκτενή χρήση των βιβλιοθηκών Javascript Prototype η οποία αντικαθιστά την JQuery και Script.aculo.us για Ajax.

2.2.2 Αρχιτεκτονική Models – Views – Controllers (MVC)

Είναι μία αρχιτεκτονική που χρησιμοποιείται για την ανάπτυξη διαδραστικών εφαρμογών στην οποία βασίζεται το Ruby on Rails για την κατασκευή της. Στην MVC αρχιτεκτονική η εφαρμογή χωρίζεται σε τρία μέρη (επίπεδα):

a) Models
b) Views
c) Controllers



Εικόνα 1 Αρχιτεκτονική MVC

a) **Models**: Το **model** είναι το κομμάτι της εφαρμογής το οποίο θέτει τους κανόνες σε αυτήν. Επιβάλει δηλαδή τις ενέργειες και τους περιορισμούς που αφορούν τα δεδομένα της εφαρμογής όντας υπεύθυνο για την αλληλεπίδραση με τη Βάση Δεδομένων. Διαχειρίζεται και προστατεύει τα δεδομένα.
b) **Views**: Είναι υπεύθυνο για τη δημιουργία διεπαφής χρήστη-υπολογιστή (user interface), το πώς δηλαδή θα γίνει η αλληλεπίδραση της εφαρμογής με το χρήστη. Το **view** δημιουργείται με βάση τα δεδομένα του **model.** Για ένα μοντέλο μπορεί να δημιουργήσουμε παραπάνω από ένα **views**, όπως για παράδειγμα ένα **view** για μία φόρμα εισαγωγής κάποιων δεδομένων και ένα διαφορετικό για την παρουσίαση τους συγκεντρωμένα. Για το ίδιο **model** δηλαδή μπορούν να υπάρξουν πολλά **views** κάθε μία για διαφορετικό σκοπό.
c) **Controllers**: Είναι υπεύθυνοι να οργανώσουν την εφαρμογή κάνοντας τον συντονιστή ανάμεσα στο **model** και το **view**. Δέχονται δεδομένα από τον χρήστη ή κάποια άλλη εφαρμογή, αλληλεπιδρούν με τα **models** και παρουσιάζουν τα κατάλληλα δεδομένα μέσω του **view** στο χρήστη.

Σε μια Rails εφαρμογή, ένα εισερχόμενο αίτημα στέλνεται αρχικά σε κάποιον δρομολογητή, ο οποίος αποφασίζει που πρέπει να σταλεί και πως πρέπει να επεξεργαστεί το αίτημα. Σε αυτό το σημείο καλείται η κατάλληλη μέθοδος στον **controller**, ο **controller** θα επεξεργαστεί το αίτημα, αν χρειαστεί θα αλληλεπιδράσει με το **model** για να εξάγει κάποια δεδομένα από την βάση δεδομένων και τελικά θα εμφανίσει τα αποτελέσματα στον χρήστη μέσω του **view.**

2.2.3 Αρχές & χαρακτηριστικά του Ruby on Rails

2.2.3.1 Η αρχή του Μην Επαναλαμβάνεστε DRY (Don’t Repeat Yourself)

Η αρχή DRY ορίζει έναν συγκεκριμένο τρόπο προγραμματισμού. Η λογική της είναι η αποφυγή της επανάληψης κα τη προσθήκη ή την αλλαγή κώδικα στην εφαρμογή μας. Αυτό επιτυγχάνεται με το να αποφεύγουμε να γράφουμε συνεχώς τον ίδιο κώδικα σε διάφορα σημεία της εφαρμογής, αλλά να γράφουμε μια φορά τον κώδικα σε κάποιο “κεντρικό” σημείο και όταν θέλουμε να χρησιμοποιήσουμε αυτό το τμήμα του κώδικα κάπου στην εφαρμογή, απλά κάνουμε μια αναφορά στο μέρος όπου βρίσκεται γραμμένος ο κώδικας. Έτσι αν θέλουμε να αλλάξουμε την συμπεριφορά της εφαρμογής δεν χρειάζεται να αλλάξουμε τα σημεία όπου καλείται ο κώδικας, αλλά μόνο το σημείο που είναι γραμμένος ο κώδικας.

2.2.3.2 Συμβάσεις αντί για Ρυθμίσεις CoC (Convention over Configuration)

Η έννοια του Convention over Configuration αναφέρεται στο γεγονός ότι το Ruby On Rails έχει ένα μεγάλο αριθμό προεπιλεγμένων ρυθμίσεων για την κατασκευή μιας τυπικής web εφαρμογής.

Ο Hansson σκοπίμως δημιούργησε το Rails έτσι ώστε να μην χρειάζεται ιδιαίτερες ρυθμίσεις, παρά μόνο να υπάρχουν κάποιες συμβάσεις. Το αποτέλεσμα ήταν να μην χρειάζεται μεγάλος αριθμός αρχείων που περιέχουν ρυθμίσεις. Στην πραγματικότητα, εάν δεν χρειάζεται να αλλάξουμε αυτές

τις προεπιλεγμένες ρυθμίσεις, το μόνο που θέλει το Rails από μας, είναι να αλλάξουμε ένα μόνο μικρό αρχείο. Αυτό το αρχείο αφορά τις ρυθμίσεις για την σύνδεση με την βάση δεδομένων.

2.2.3.3 Ευέλικτος Προγραμματισμός (Agile Development)

Οι παραδοσιακές προσεγγίσεις για την ανάπτυξη εφαρμογών, συνήθως οργανώνουν ένα μακροχρόνιο και στατικό σχέδιο ώστε να καταφέρουν τους στόχους μιας εφαρμογής, χρησιμοποιώντας διάφορες προφητικές μεθόδους. Αυτός ο τύπος προγραμματισμού ονομάζεται και "από κάτω προς τα πάνω", που σημαίνει πως ο προγραμματιστής ξεκινάει να δουλεύει με τα τα δεδομένα και έπειτα ασχολείται με το σχεδιαστικό μέρος.

Σε αντίθεση οι μέθοδοι agile development, χρησιμοποιούν μια πιο προσαρμοσμένη προσέγγιση. Μικρές ομάδες προγραμματιστών αναλαμβάνουν να ολοκληρώσουν με επαναλαμβανόμενο τρόπο μικρά μέρη της εφαρμογής. Πριν ξεκινήσει μια επανάληψη, η ομάδα επανεκτιμά τους στόχους και τις προτεραιότητες του τμήματος που έχει αναλάβει να αναπτύξει, με βάση την μέχρι τότε πορεία όλου του project. Οι στόχοι και οι προτεραιότητες ενδέχεται να έχουν αλλάξει σε σχέση με την προηγούμενη επανάληψη, όποτε χρειάζεται να επαναπροσδιοριστούν. Οι προγραμματιστές που χρησιμοποιούν μεθόδους Agile development, αρχίζουν το "χτίσιμο" της εφαρμογής "από πάνω προς τα κάτω", ξεκινώντας με το σχεδιαστικό μέρος που συνήθως είναι ένα απλό σκίτσο της διεπαφής σχεδιασμένο σε χαρτί. Όταν μια εφαρμογή αναπτύσσεται με μεθόδους agile development, μοιάζει να είναι εκτός ελέγχου κατά την διάρκεια του προγραμματισμού, εξαιτίας των συνεχών επαναπροσδιορισμών των στόχων και των προτεραιοτήτων. Ξοδεύοντας όμως λίγο χρόνο στη σχεδιασμό λειτουργικών προδιαγραφών και πιο μακροπρόθεσμων σχεδίων, χωρίς να βέβαια να ξεφεύγουμε από την λογική του agile development, η ανάπτυξη της εφαρμογής αποκτά συνοχή και ξεπερνιέται το πρόβλημα που αναφέρθηκε.

Όταν μια εφαρμογή αναπτύσσεται με μεθόδους agile development, μοιάζει να είναι εκτός ελέγχου κατά την διάρκεια του προγραμματισμού, εξαιτίας των συνεχών επαναπροσδιορισμών των στόχων και των προτεραιοτήτων. Ξοδεύοντας όμως λίγο χρόνο στη σχεδιασμό λειτουργικών προδιαγραφών και πιο μακροπρόθεσμων σχεδίων, χωρίς να βέβαια να ξεφεύγουμε από την λογική του agile development, η ανάπτυξη της εφαρμογής αποκτά συνοχή και ξεπερνιέται το πρόβλημα που αναφέρθηκε.

Μερικά παραδείγματα που φανερώνουν πως το Rails χρησιμοποιεί μεθόδους του agile development είναι:

a) Έχουμε την δυνατότητα να ξεκινήσουμε με τον σχεδιασμό (layout), πριν ξεκινήσουμε να ασχολούμαστε με το κομμάτι των δεδομένων. Δεν χρειάζεται να ασχοληθούμε ξανά με το σχεδιαστικό μέρος όταν ξεκινήσουμε να δίνουμε λειτουργικότητα στην εφαρμογή μας, εκτός αν επιθυμούμε να αλλάξουμε την εμφάνισή της.

b) Αντίθετα με γλώσσες όπως η C και η Java, μια Rails εφαρμογή δεν χρειάζεται να κάνει μεταγλώττιση για να γίνει εκτελέσιμη. Ο κώδικας σε Ruby ερμηνεύεται αμέσως έτσι δεν χρειάζεται την οποιαδήποτε μεταγλώττιση για να γίνει εκτελέσιμος. Η αλλαγή κάποιου τμήματος κώδικα δίνει στον προγραμματιστή αμέσως αποτελέσματα και έτσι αυξάνεται η ταχύτητα της ανάπτυξης μιας εφαρμογής.

c) Το Rails παρέχει ένα χρήσιμο framework αυτόματης δοκιμής (testing) κάποιου τμήματος του κώδικα της εφαρμογής. Οι προγραμματιστές που κάνουν χρήση αυτού του εργαλείου είναι σίγουροι πως δεν υπάρχει περίπτωση κάνοντας δοκιμές να χαλάσουν ότι έχουν φτιάξει μέχρι εκείνη την στιγμή.

Όταν χρειαστεί να αλλαχτεί κάποιο τμήμα του κώδικα για να βελτιστοποιηθεί η εφαρμογή αλλάζοντας κάποιες προτεραιότητες, ή αν χρειαστεί να προστεθούν περισσότερα χαρακτηριστικά στην εφαρμογή, γίνεται πολύ απλά αν ο προγραμματιστής χρησιμοποιήσει τις αρχές του DRY. Αυτό ισχύει γιατί χρειάζονται πολύ λιγότερες αλλαγές αν χρησιμοποιήσουμε ήδη υπάρχοντα κώδικα που έχει γραφτεί μια φορά και επαναχρησιμοποιείται σε διάφορα σημεία της εφαρμογής.

2.2.4 Βάσεις Δεδομένων στο Rails

Το μεγαλύτερα μέρος των εφαρμογών, αν όχι όλες, χρειάζονται μια βάση δεδομένων ώστε να αποθηκεύουν τα δεδομένα και να τα χρησιμοποιούν όποτε τα χρειάζονται. Το Framework Ruby On Rails υποστηρίζει μια αρκετά μεγάλη γκάμα από συστήματα διαχείρισης βάσεων δεδομένων.

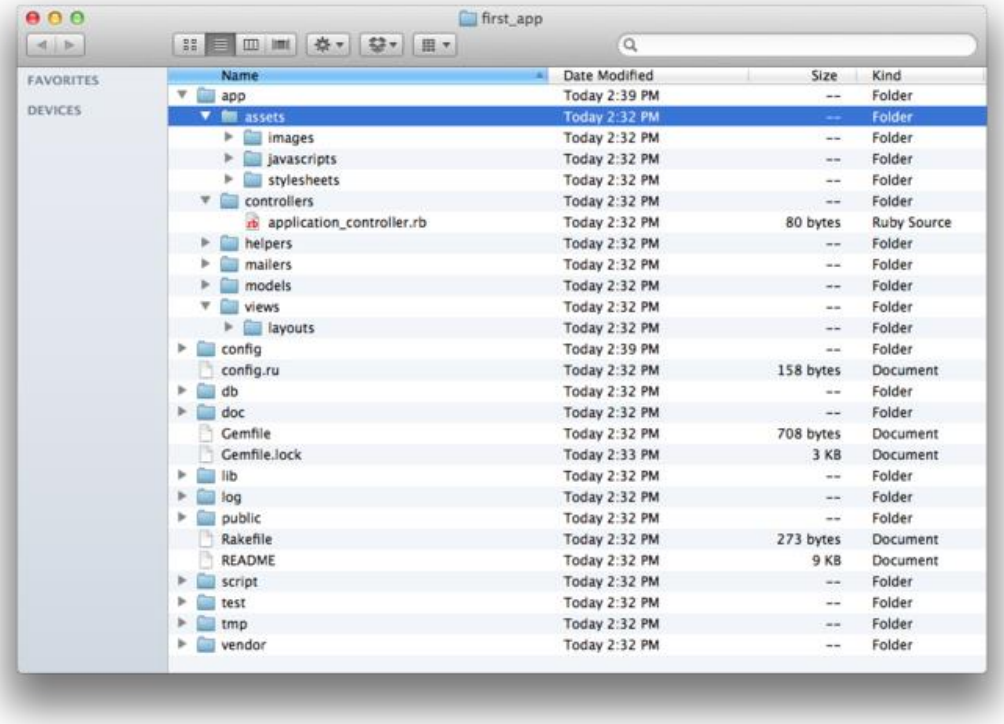
Μια ενδεικτική λίστα με τις διευθύνσεις που μπορούμε να βρούμε τους προσαρμογείς (adapters) είναι:

a. MySQL http://www.tmtm.org/en/mysql/ruby/
b. Oracle http://rubyforge.org/projects/ruby-oci8/
c. SQL Server http://rubyforge.org/projects/ruby-dbi/
d. SQLite http://rubyforge.org/projects/sqlite-ruby/
e. Postgres http://ruby.scripting.ca/postgres/
f. Firebird http://rubyforge.org/projects/fireruby/
g. DB2 http://rubyforge.org/projects/ruby-ibm/

Το προεπιλεγμένο σύστημα διαχείρισης βάσεων δεδομένων που έχει το Rails, όταν ξεκινάει για μια νέα εφαρμογή είναι το SQLite, το οποίο μπορεί πολύ εύκολα να αλλάξει, εγκαθιστώντας τον προσαρμογέα για την βάση δεδομένων, και παραμετροποιώντας ένα μικρό αρχείο που έχει πληροφορίες σχετικά με την σύνδεση με την βάση δεδομένων.

2.2.5 Βασική δομή των καταλόγων (directory) μιας Rails εφαρμογής

Χρησιμοποιώντας την Ruby on Rails για τη δημιουργία μίας εφαρμογής, η δομή που δημιουργείται αποτελείται από τους παρακάτω καταλόγους:



Εικόνα 2 Δομή καταλόγων RoR

Ο σκοπός αυτών των καταλόγων είναι:

a) **app:**
Περιέχει τους καταλόγους με τα αρχεία των models, των views και των controllers.

b) **app/controller:**
Περιέχει τα αρχεία των controllers.

c) **app/views:**
Περιέχει τα αρχεία των views.

d) **app/models:**
Περιέχει τα αρχεία των models.

e) **app/helpers:**

Περιέχει βοηθητικά αρχεία που σκοπό έχουν να συμβάλλουν ώστε να μείνει ο κώδικας στους controllers και στα models σχετικά μικρός και ευανάγνωστος.

f) **app/views/layouts:**
Περιλαμβάνει templates τα οποία είναι χρήσιμα και συνδυάζονται με τα views.

g) **components:**
Ο συγκεκριμένος κατάλογος περιέχει μικρές αυτόνομες εφαρμογές χρήσιμες για models, controllers και views.

h) **config:**
Περιέχει αρχεία ρυθμίσεων της εφαρμογής όπως για παράδειγμα το database.yml το οποίο είναι για τη σύνδεση με τη Βάση Δεδομένων.

i) **db:**
Περιέχει scripts διαχείρισης της Βάσης Δεδομένων της εφαρμογής.

j) **doc:**
Περιέχει αρχεία με οδηγίες χρήσης (documentation) τα οποία παράγει το rdoc της Rails τα οποία βασίζονται στα σχόλια στα οποία υπάρχουν στον κώδικα.

k) **lib:**
Περιέχει τις βιβλιοθήκες της εφαρμογής.

l) **log:**
Περιέχει αρχεία με καταγεγραμμένα λάθη της εφαρμογής όπως το server.log για τα λάθη που γίνονται στον server και development.log για αυτά της εφαρμογής.

m) **public:**
Περιέχει αρχεία Jacascript, Images, Stylesheets κ.α.

n) **script:**
Περιέχονται scripts με σκοπό τη διαχείριση κάποιων εργαλειών.

o) **test:**
Αρχεία που χρειάζονται σε λειτουργία περιβάλλοντος test.

p) **tmp:**
Περιέχει προσωρινά αρχεία.

q) **vendor:**
Περιέχει βιβλιοθήκες που χρειάζονται για τρίτες εφαρμογές.

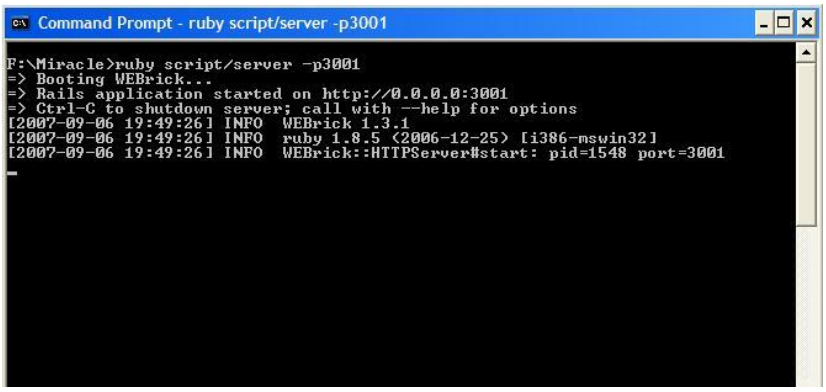
r) **Readme:**
Αρχείο το οποίο περιέχει πληροφορίες για την εφαρμογή μας.

s) **Rakefile:**
Είναι αρχείο παρόμοιο με το makefile αρχείο του Linux. Βοηθάει στην ανάπτυξη (building), την πακετοποίηση (packaging) και στις δοκιμές (testing) του κώδικα της Ruby on Rails.

2.2.6 Δευτερεύοντα Frameworks και βοηθητικά εργαλεία (Webrick & Rake & Devise)

Είναι επιμέρους εργαλεία με τα οποία είναι συνδεδεμένο το Rails και είναι απαραίτητα στους προγραμματιστές που χρησιμοποιούν το Framework αυτό.

Webrick:
Είναι ένας ελαφρύς web server για καθαρά δοκιμαστική χρήση στη φάση της συγγραφής ο οποίος περιέχεται στις βασικές βιβλιοθήκες της Ruby on Rails.



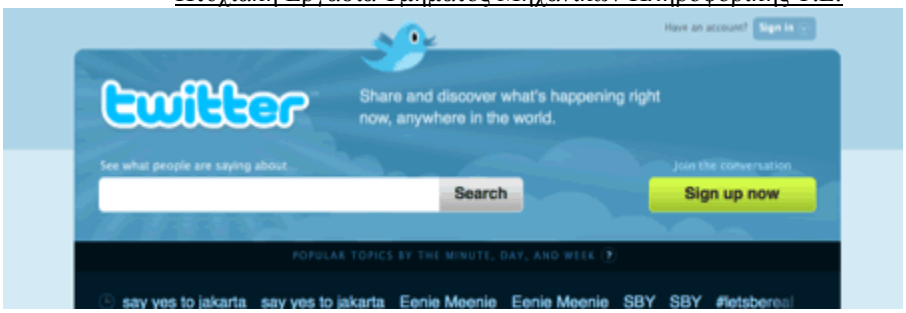
Εικόνα 3 Webrick web server

Rake:
Είναι ένα εργαλείο διαχείρισης προγραμμάτων σε command prompt περιβάλλον.

Devise:
Είναι ένα μικρό αλλά αποτελεσματικό και πρακτικό framework. Περιέχει και υλοποιεί όλες τις απαραίτητες λειτουργίες για τη καταχώρηση χρηστών σε μία βάση, για την πιστοποιημένη είσοδό τους σε μία εφαρμογή και τη καταχώρηση όλων των σημαντικών πληροφοριών όπως ώρα, Διευθ. IP κλπ. και του ιστορικού εισόδων ενός χρήστη σε μια εφαρμογή. Είναι εύκολα τροποποιήσιμο framework το οποίο έχει πολύ καλή συνεργασία με το Ruby on Rails.

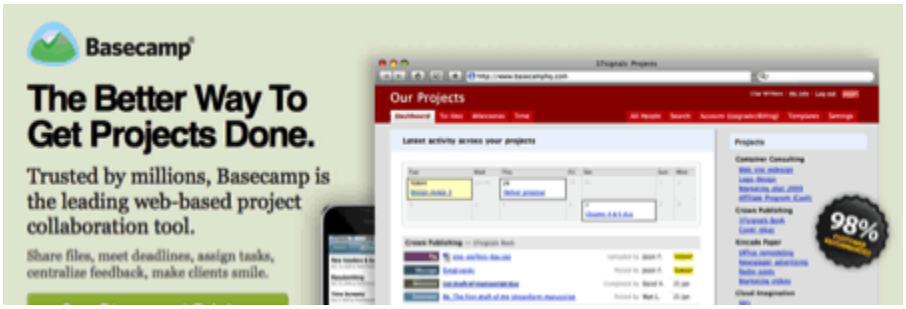
2.2.7 Παραδείγματα υλοποιήσεων σε Ruby on Rails

a) Το γνωστό σε όλους μας Twitter



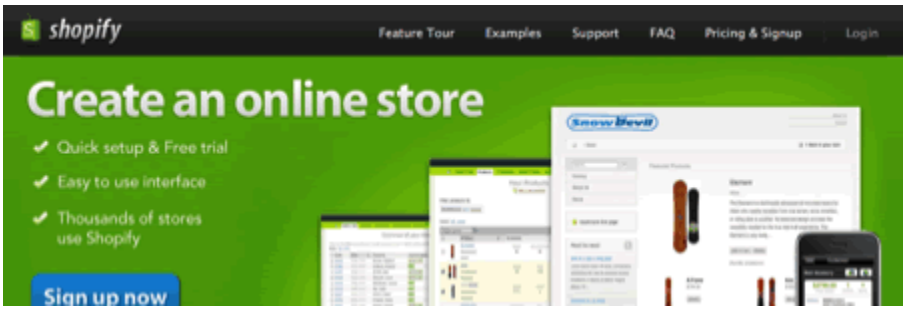
https://twitter.com/

b) Basecamp. Ένα διάσημο browser – based εργαλείο ανάθεσης projects



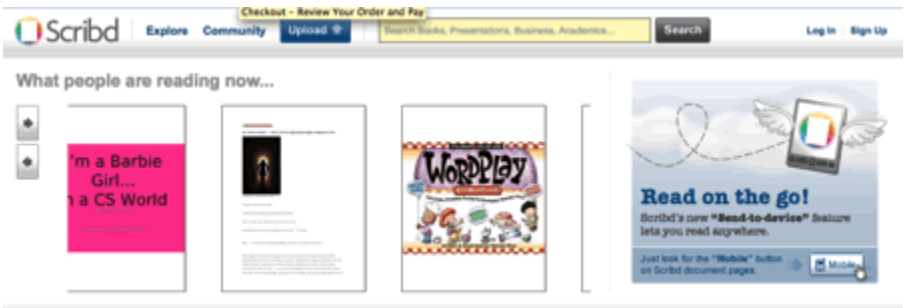
https://basecamp.com/

c) Shopify. Μία διάσημη πλατφόρμα δημιουργίας ηλ.καταστήματος



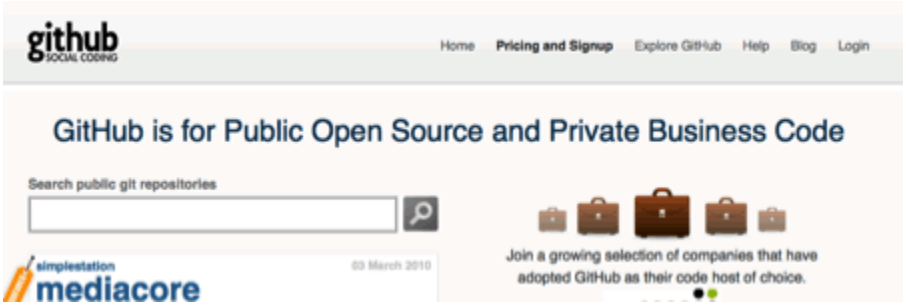
http://www.shopify.com/

d) Scribd. Μία ωραία και διάσημη ιδέα κοινωνικής δημοσίευσης άρθρων κ.α.



http://www.scribd.com/

e) Το Github παρέχει μία web-based φιλοξενία του κώδικα ολοκληρωμένων εφαρμογών με έντονο "κοινωνικής δικτύωσης χαρακτήρα".



https://github.com/

f) Το skroutz που είναι αρκετά διάσημο στην Ελλάδα βασίζεται σε μεγάλο βαθμό στη Ruby.

http://www.skroutz.gr/

2.3 HTML5 & CSS3

2.3.1 HTML5 (Συνοπτικά)

Η HTML5 είναι μια υπό ανάπτυξη γλώσσα Υπερκειμένου (HyperText Markup Language) που προορίζεται για την επόμενη έκδοση της HTML. Η W3C (World Wide Web Consortium) και η WHATWG (Hypertext

Application Technology Working Group) συνεργάστικαν για την ανάπτυξη της. Η πρώτη ιδέα για τη συνεργασία τους ώστε να δημιουργήσουν την HTML5 έγινε το 2006.

Το 2004 ξεκίνησε η WHATWG τη δουλειά για την έκδοση αυτή.με το Web Appliacation 1.0 μέχρι το 2010 πού δεν ήταν ακόμα έτοιμο.

Η έκδοση HTML5 δημιουργήθηκε με σκοπό την αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM HTML.

Η WHATWG πρόσθεσε στην έκδοση το πρότυπο Web Forms 2.0.

2.3.2 CSS3 (Συνοπτικά)

Τα CSS (Cascading Style Sheets) είναι αρχεία τα οποία περιέχουν κανόνες μορφοποίησης περιεχομένου και χρησιμοποιούνται στην κατασκευή ιστοσελίδων.

Κανόνες που μπορεί να περιέχει ένα CSS είναι:

a) Καθορισμός χρωμάτων και μέγεθος στοιχείων των περιεχομένων της ιστοσελίδας.
b) Περιλαμβάνει κανόνες συμπεριφοράς.

2.4 Javascript

Η Javascript είναι μια γλώσσα σεναρίων με τη βοήθεια της οποίας οι ιστοσελίδες βελτιώνουν τη λειτουργικότητά τους. Μέσω της Javascript δημιουργούμε μεθόδους οι οποίες μέσω βιβλιοθηκών ενσωματώνονται ή εισάγονται σε HTML σελίδες. Μία από τις χρήσεις είναι για να κάνουμε διαδραστικό το περιεχόμενο μιας ιστοσελίδας. Άλλα παραδείγματα χρήσης της Javascript εκτός ιστοσελίδων, είναι για παράδειγμα σε έγγραφα PDF και εφαρμογές στη επιφάνεια εργασίας.

2.5 Apache Web Server

Ο apache HTTP αναπτύσσεται από την Κοινότητα Ανοιχτού Λογισμικού με την υποστήριξη της Apache Software Foundation. Πρόκειται για ένα ανοιχτού κώδικα λογισμικό οπότε και διατήθεται δωρεάν με ελευθερία στους χρήστες να πραγματοποιήσουν αλλαγές στο κώδικά τους.

Ο Apache μπορεί να λειτουργήσει σε πολλά λειτουργικά συστήματα π.χ. Linux, Unix, Microsoft Windows, GNU, FreeBSD, Solaris, Novell, NetWare, Mac OS X, OS/2, TPF. Ο Apache διαχειρίζεται αιτήσεις, τις δέχεται από τους χρήστες (clients) για παράδειγμα έναν browser και έπειτα επιστρέφει τη σελίδα που ζητήθηκε.

Χρησιμοποιεί HTTP πρωτόκολλο για τι διαχείτιση αυτή.

2.7 Πλατφόρμα Διαχείρισης Περιεχομένου (CMS)

Το CMS (Content Management System) είναι μία μορφή λογισμικού για ηλεκτρονικούς υπολογιστές, που αυτοματοποιεί τις διαδικασίες δημιουργίας, οργάνωσης, ελέγχου και δημοσίευσης περιεχομένου σε μία πληθώρα μορφών.

2.7.1 Joomla

Το Joomla είναι ένα πλήρες Σύστημα Διαχείρισης Περιεχομένου (CMS) του οποίου οι δυνατότητες το καθιστούν ένα πλήρως ευέλικτο και φιλικό στη χρήση περιβάλλον εφαρμογής. Χρησιμοποιώντας Joomla μπορούμε να δημοσιεύσουμε στο διαδίκτυο από μια προσωπική ιστοσελίδα (portfolio) μέρχι κι ένα μεγάλο ηλεκτρονικό κατάστημα.

Η εφαρμογή είναι Open source δηλαδή ανοιχτού κώδικα και η χρήση του είναι εντελώς δωρεάν. Μας αφήνει λοιπόν ελεύθερους να το χρησιμοποιήσουμε, να το τροποποιήσουμε και να εξαντλήσουμε τις δυνατότητές του χωρίς να χρειαστεί να πληρώσουμε κάποια άδεια χρήσης. Οι δυνατότητες επέκτασής της είναι απεριόριστες.

ΚΕΦΑΛΑΙΟ 3ο: ΔΙΑΔΙΚΑΣΙΑ ΥΛΟΠΟΙΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

3.1 Περιγραφή της εφαρμογής

Ο σκοπός και το κίνητρο για τη δημιουργία της παρούσας πτυχιακής είναι η προσπάθεια να γίνει ευκολότερη και συγχρόνως αποδοτικότερη η δουλειά ενός web developers. Η κοινότητα των web developers / designers κι οποιουδήποτε ασχολείται με το διαδίκτυο σε παραγωγικό επίπεδο, εξελίσσεται με ταχύτατους ρυθμούς εδώ και πολλά χρόνια. Μεγάλη είναι επίσης και η χρήση CMS εργαλείων από τους developers με δημοφιλέστερα το Joomla και το Wordpress.

Στα πλαίσια της πτυχιακής αναπτύχθηκε μία εφαρμογή σε Ruby on Rails προσπαθώντας να επωφεληθούμε από τα θετικά χαρακτηριστικά και τις ιδιαιτερότητες της γλώσσας προγραμματισμού Ruby και την άνεση του framework που είναι βασισμένο σε αυτή το Ruby on Rails. Η εφαρμογή θα είναι συμβατή με οποιαδήποτε πλατφόρμα διαχείρισης περιεχομένου CMS.

Μέσω αυτής της εφαρμογής παρέχεται στο χρήστη της η δυνατότητα συνδεθεί σε οποιοδήποτε εγκατεστημένο CMS μέσω ενός ενιαίου περιβάλλοντος διαχείρισης, θέλοντας να διευκολύνουμε έτσι τον administrator της εκάστοτε CMS ιστοσελίδας. Ο διαχειριστής θα χρησιμοποιεί έναν μόνον κωδικό για να συνδεθεί στην εφαρμογή και να αποθηκεύσει εκεί τα συστήματα που θέλει να διαχειρίζεται με κύριο σκοπό να συνδέεται με ένα κωδικό σε όσα projects επιθυμεί.

Ταυτόχρονα θα μπορεί μέσω της εφαρμογής να ορίζει εκκρεμότητες με ημερομηνία έναρξης και λήξης οι οποίες υπάρχει η δυνατότητα να δημιουργούνται και στο Google Calendar αυτόματα, ενώ το σύστημα θα είναι υπεύθυνο για την υπενθύμιση της εκκρεμότητας που πλησιάζει προς τη λήξη της. Επίσης θα μπορεί να κρατάει γενικές σημειώσεις οι οποίες αναρτώνται σε μορφή δημοσιεύσεων όπως σε κάποια κοινωνικά μέσα δικτύωσης τις οποίες υπάρχει η δυνατότητα να τις βλέπουν κι άλλοι χρήστες στη περίπτωση μιας εταιρίας για παράδειγμα.

Είναι λοιπόν σαφές ότι η εφαρμογή μπορεί να διευκολύνει τους διαχειριστές είτε οργανώνοντας τη μέρα τους είτε διαφυλάσσοντας ασφαλείς τους κωδικούς των backend των ιστοσελίδων που διαχειρίζονται.

3.2 Άλλες εφαρμογές της αγοράς

Υπάρχουν αρκετές εφαρμογές στο διαδίκτυο που σκοπό έχουν τη διευκόλυνση ενός web developer ή ενός διαχειριστή ιστοσελίδας.

Σας παραθέτω μερικές:

a) LaunchList

Αυτό το εργαλείο βοηθά το χρήστη να αναθεωρήσει σημαντικά στοιχεία πριν από τη μεγάλη έναρξη. Από προεπιλογή, το εργαλείο παρέχει 28 σημεία που πρέπει να ελεγχθούν, αλλά σας επιτρέπει επίσης να προσθέσετε προσαρμοσμένα στοιχεία στη λίστα. Κάθε στοιχείο μπορεί να σχολιαστεί ή να διαγραφεί. Μόλις τελειώσετε, μπορείτε να στείλετε την έκθεση μαζί με τις λεπτομέρειες του έργου του σε πολλαπλούς παραλήπτες μέσω του ηλεκτρονικού ταχυδρομείου. Εναλλακτικές λύσεις είναι Ultimate Website Launch Checklist and Paul Boag's The Ultimate Website Prelaunch Checklist.

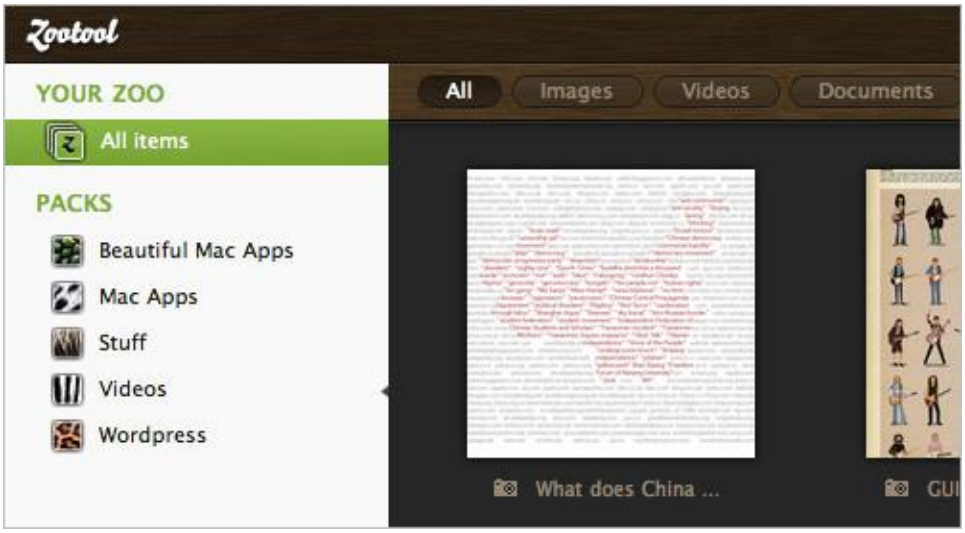
http://launchlist.net/

b) Pencil Project: Σκίτσο και Πρωτότυπα με τον Firefox. Το Pencil είναι ένα open source εργαλείο GUI πρωτοτύπων. Περιέχει ενσωματωμένα stencils για διαγράμματα και προτυποποίηση, στην οθόνη επεξεργασίας κειμένου με υποστήριξη εμπλουτισμένου κειμένου, καθώς και τυπικές λειτουργίες σχεδίασης. Συμβατό με Firefox 3.5 +.

http://pencil.evolus.vn/Default.html

c) Zootool

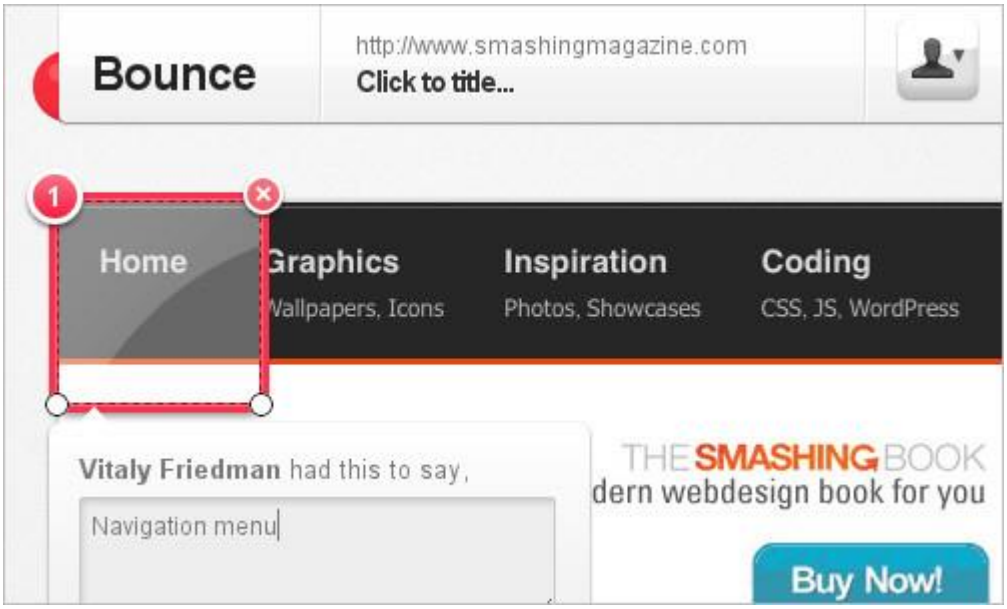
Το Zootool είναι μια ιστοσελίδα bookmarking και εργαλείο για τη συλλογή εικόνων, εγγράφων, συνδέσμων και βίντεο από οπουδήποτε στο Διαδίκτυο. Ένα bookmarklet σας επιτρέπει να συλλέγετε στοιχεία γρήγορα και εύκολα. Στη συνέχεια μπορείτε να επισημάνετε και να οργανώσετε τα αποθηκευμένα στοιχεία σας στην back end του Zootool. Μπορείτε επίσης να διασυνδεθεί το Zootool με Tumblr, Twitter, FriendFeed Delicious και να μοιράζεστε ό, τι μπορείτε να βρείτε. Screenshot μέσω MacStories.



http://zootool.com/

d) Bounce

Ένα διασκεδαστικό και εύκολο τρόπο για να μοιραστούν τις ιδέες σε ένα δικτυακό τόπο. Το εργαλείο σας επιτρέπει να κρατάτε σημειώσεις, να γράψετε σχόλια σε μια επικάλυψη του κάθε τόπου και στη συνέχεια να μοιραστείτε τις σημειώσεις σας με τους φίλους σας.



http://www.bounceapp.com/

3.3 Υλοποίηση εφαρμογής

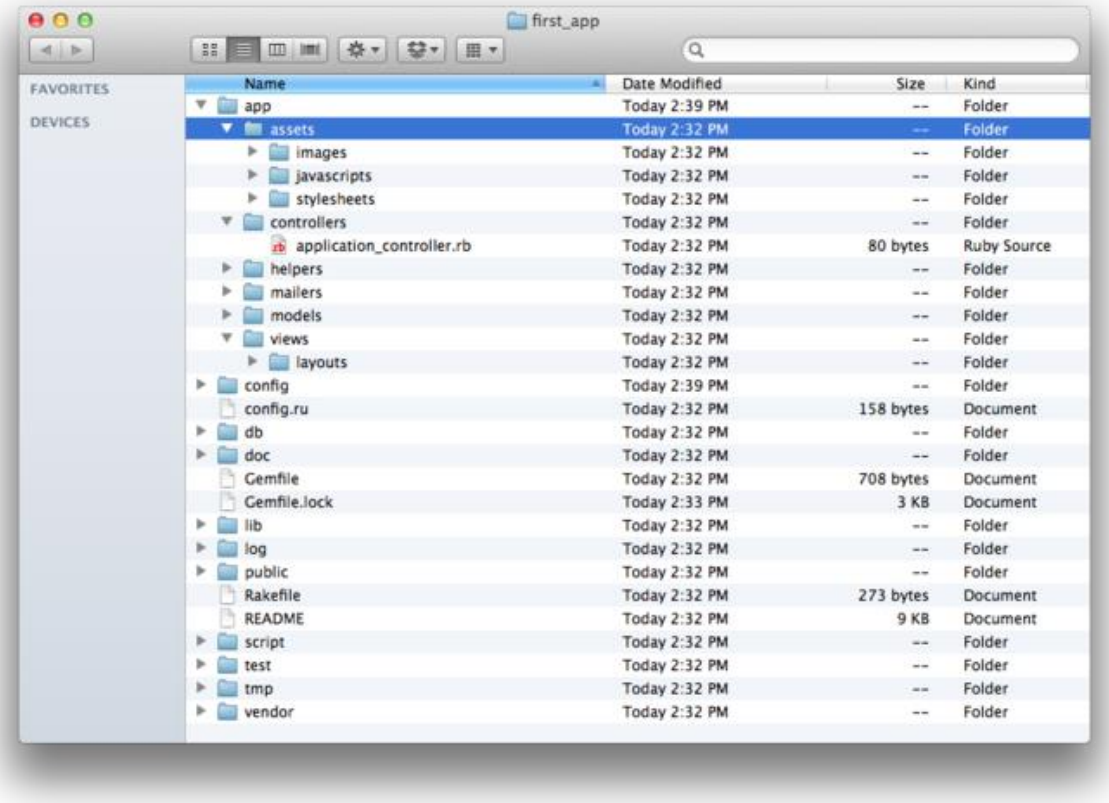
Πλήρης διαδικασία σύμφωνα με την οποία υλοποιήθηκε η εφαρμογή.

3.3.1 Υλοποίηση της εφαρμογής σε Ruby on Rails

Η υλοποίηση της εφαρμογής στο Ruby on Rails ξεκινά χρησιμοποιώντας το τερματικό στο οποίο αφού πρώτα βρούμε το φάκελο στον οποίο θέλουμε να εγκατασταθεί η εφαρμογή δίνουμε την αντίστοιχη εντολή.

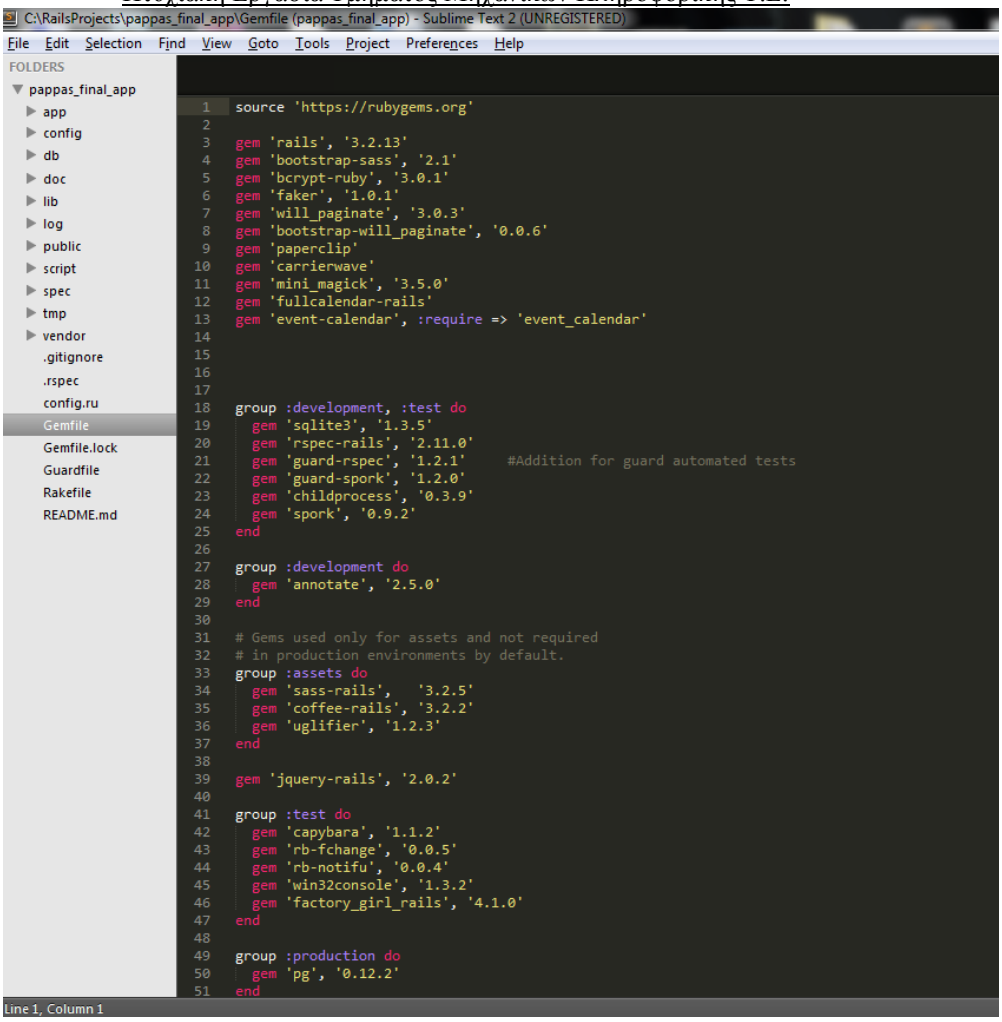
```
$ cd ~/rails_projects
$ rails new sample_app --skip-test-unit
$ cd sample_app
```

Με τον τρόπο αυτό δημιουργούνται αυτόματα όλοι οι φάκελοι και τα αρχεία που αποτελούν τη βασική δομή της εφαρμογής όπως την αναλύσαμε στην ενότητα 2.3.5.



Εικόνα 4 Δομή καταλόγων RoR

Επόμενη κίνηση είναι να συμπληρώσουμε στο Gemfile της εφαρμογής, το οποίο ένα από τα σημαντικότερα της αρχεία. Εδώ γίνεται η εγκατάσταση των βιβλιοθηκών και των gems (πακέτα εφαρμογών) της Ruby. Το Gemfile το γεμίζουμε σταδιακά με ότι χρειαζόμαστε σε κάθε στάδιο. Παρακάτω σας παραθέτω μια πλήρη και τελική εικόνα του Gemfile της εφαρμογής.



Εικόνα 5 Gemfile

Αξίζει να απισημάνουμε κάποια από τα Gems που χρησιμοποιήθηκαν στην εφαρμογή μας.

- gem 'bootstrap-sass', '2.1' :
  Προσφέρει έτοιμες λύσεις σχεδίασης css.scss και εντολών javascript.

- gem 'bcrypt-ruby', '3.0.1' :
  Προσφέρει ασφάλεια στην εφαρμογή (Password Encryption)

- gem 'paperclip' :
  Πρόκειται για μία βιβλιοθήκη για τον Active Record επισύναψης αρχείων.

- gem 'carrierwave' :
  Uploading αρχείων στην εφαρμογή.

- gem 'rspec-rails', '2.11.0' :
  Πολύ σημαντικό βοήθημα στην ανάπτυξη μιας Rails εφαρμογής. Πρόκειται για ένα framework το οποίο σου επιτρέπει να ελέγχεις την ορθότητα του κώδικα.

- gem 'coffee-rails', '3.2.2' :
  Αποτελεί αντάπτορα για το asset pipeline της Rails καθώς επίσης υποστηρίζει Javascript Requests.

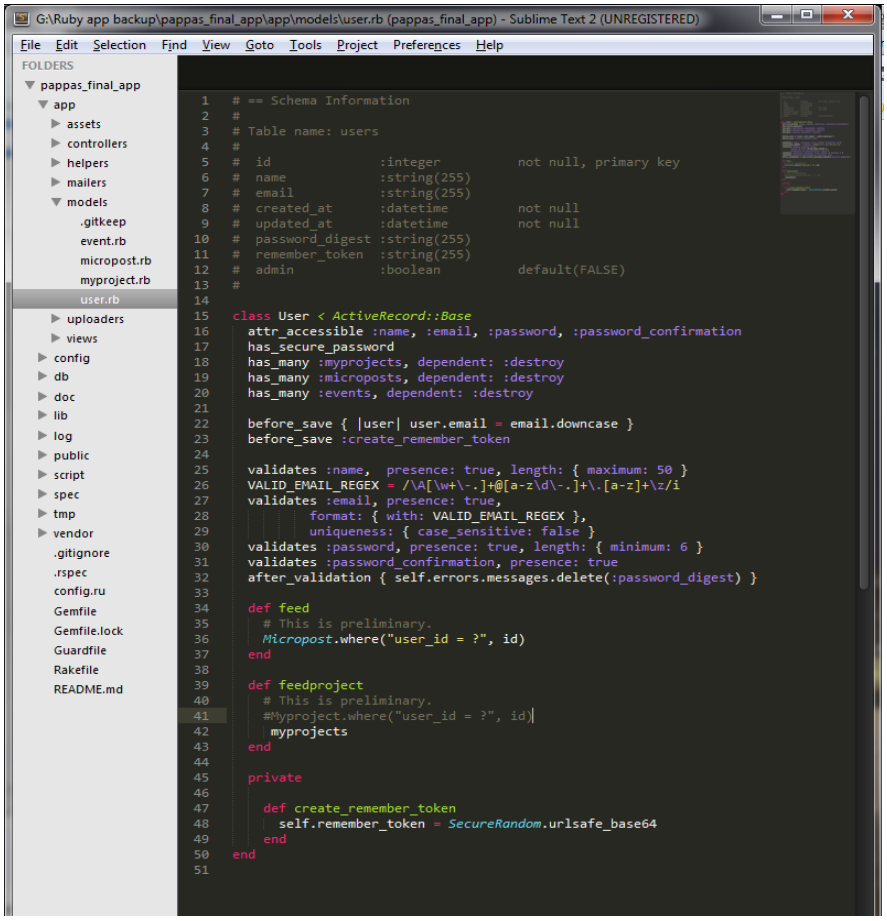
- gem 'capybara', '1.1.2' :
  Σου δίνει τη δυνατότητα της δοκιμαστικής εξομοίωσης του πώς αλληλεπιδρά ο χρήστης με την εφαρμογή.

3.3.2 Μοντέλα και όψεις και ελεγκτές της εφαρμογής

Όπως κάθε εφαρμογή ανεπτυγμένη σε Ruby on Rails αποτελείται από Μοντέλα (Models), Όψεις (Views) και Ελεγκτές (Controllers) όπως προκύπτει από την αρχιτεκτονική MVC έτσι και η δική μας, τα οποία θα συζητήσουμε σε αυτή την ενότητα.

Ένα από τα σημαντικά Models είναι ο χρήστης (user) του οποίου στο μοντέλο ξεκαθαρίζεται η σχέση που έχει με τα υπόλοιπα μοντέλα. Όπως θα δούμε και παρακάτω άλλα μοντέλα είναι το myproject και το micropost. Η σχέση που έχουν ο χρήστης με τα δύο αυτά μοντέλα είναι ότι ο χρήστης έχει πολλά myproject και πολλά microposts, ενώ και τα δύο αυτά μοντέλα έχουν ένα μόνον χρήστη.. Ορίζονται επίσης και τα πεδία που έχει αυτό το μοντέλο στη βάση δεδομένων. Το ότι encrypted κωδικό, καθώς και τα validations που επιβάλει η εφαρμογή στον ενδυνάμει χρήστη όσον αφορά τα στοιχεία που εισάγει.

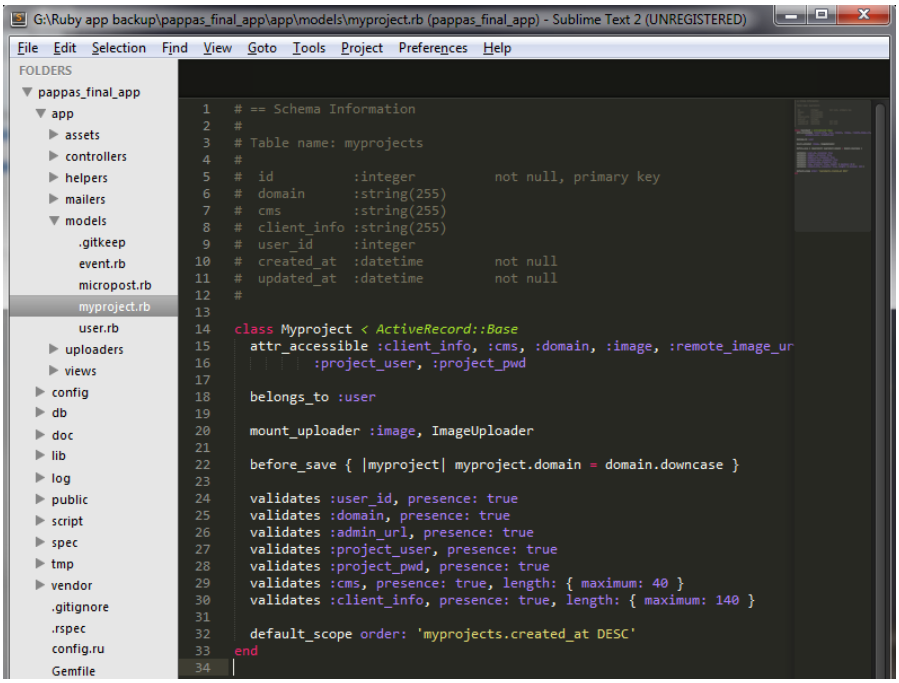
Ακολουθεί ο κώδικας του μοντέλου του χρήστη (user.rb)



Εικόνα 6 Model user.rb

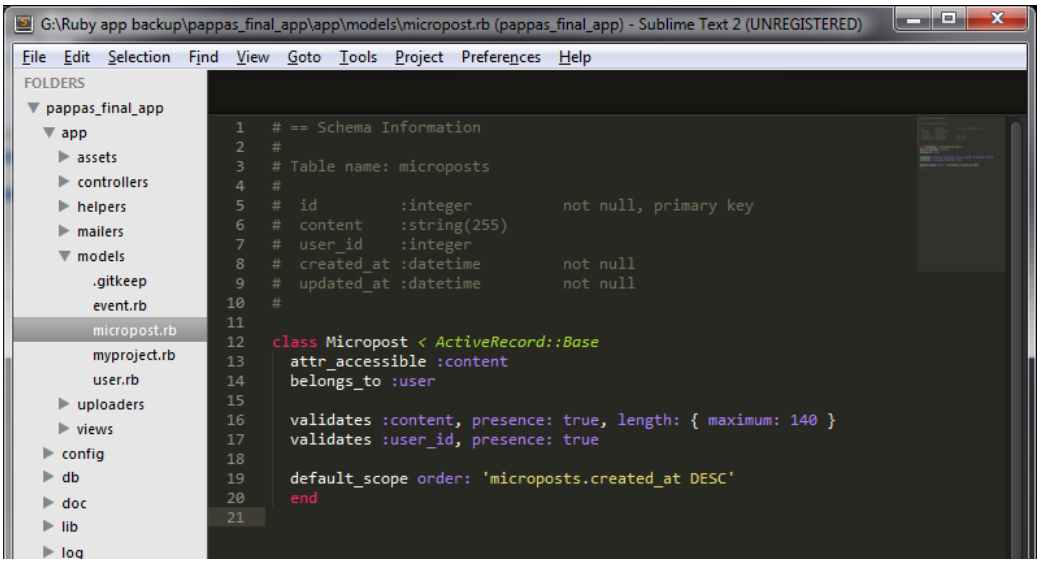
Το μοντέλο myproject περιέχει και αυτό τα πεδία από τα οποία αποτελείται στη βάση δεδομένων. Περιέχει επίσης validations για την είσοδο των δεδομένων καθώς και τη σχέση του με το χρήστη. Τέλος περιέχει έναν image uploader για τις φωτογραφίες η οποίες ανεβαίνουν στα project που εγκαθιστά ο χρήστης.

Ακολουθεί ο κώδικας του μοντέλου του myproject (myproject.rb)



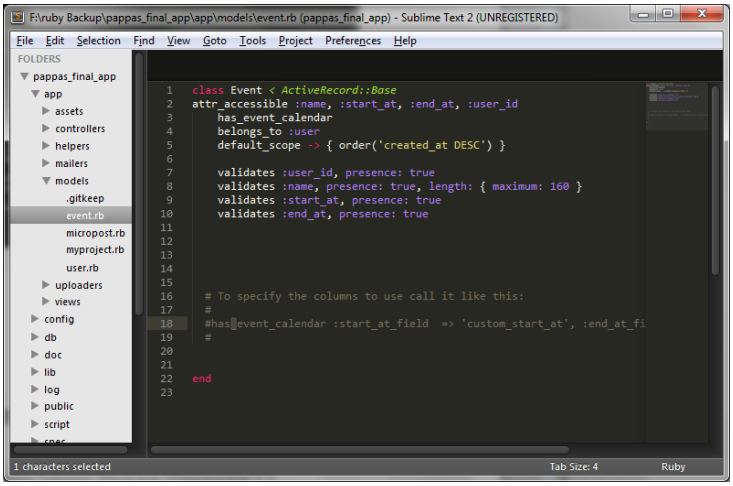
Εικόνα 7 Model Myproject.rb

Με τη σειρά του το μοντέλο microposts το οποίο αναφέρεται σε σημειώσεις που θα κρατάει ο χρήστης περιέχει κάποια λίγα validations, το ένα πεδίο που έχει στη βάση και τη σχέση του με το χρήστη.



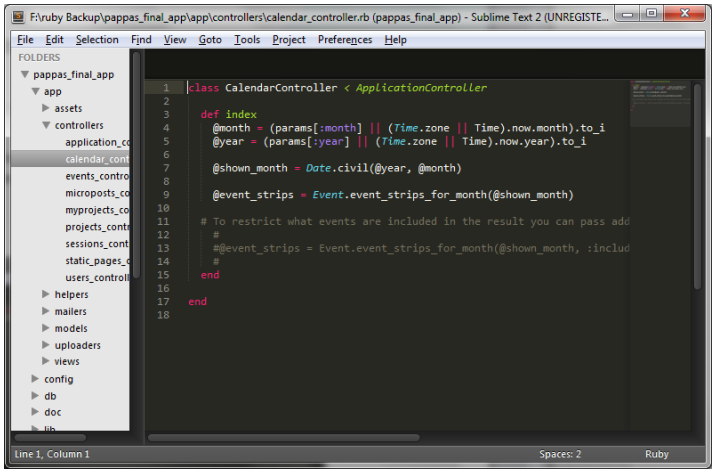
Εικόνα 8 Model Micropost.rb

Υπάρχει και το event μοντέλο το οποίο το χρησιμοποιεί το calendar για την λειτουργία του ημερολογίου. Μας δείχνει τη συσχέτιση που μπορεί να έχει με το Calendar καθώς και με το χρήστη user στον οποίο ανήκει.



Εικόνα 9 Model Event.rb

Τα μοντέλα Calendar και Event έχουν από έναν Controller οι οποίοι περιέχουν συγκεκριμένες οδηγίες για τη λειτουργία τους.



Εικόνα 10 Calendar Controller

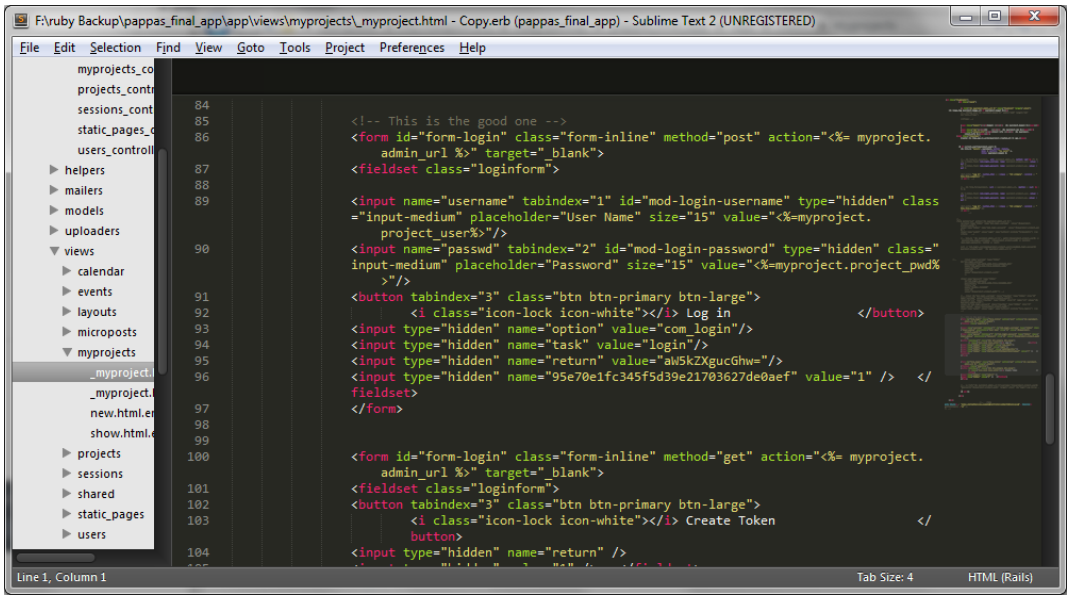
Γενικότερα όπως έχει αναφερθεί και παραπάνω στη Ruby on Rails που βασίζεται στη Ruby κάθε οντότητα περιγράφεται με ένα Model το οποίο ακούει τον Controller του και μπορούμε να το δούμε από το View του.

3.3.3 Υλοποίηση Διασύνδεσης με Joomla

Όπως αναφέραμε και παραπάνω το Joomla είναι ένα πλήρες Σύστημα Διαχείρισης Περιεχομένου (CMS) του οποίου οι δυνατότητες το καθιστούν ένα πλήρως ευέλικτο και φιλικό στη χρήση περιβάλλον εφαρμογής. Ένα τέτοιο CMS δε θα μπορούσε να αφήσει την ασφάλεια του να είναι μια απλή υπόθεση.

Η διαδικασία κατά την απόπειρά μας να συνδεθούμε με το Backend του Joomla έχει ως εξής. Με την επίσκεψή μας στη τοποθεσία URL του site που διαχειριζόμαστε το οποίο θα μπορούσε να είναι http://www.myjoomla.com/administrator/ δημιουργείται ακολουθία στοιχείων τυχαία σύμφωνα με έναν αλγόριθμο οποίος ονομάζεται "Security Token" και αυτόματα αποθηκεύεται στη Βάση Δεδομένων της Joomla. Αυτός είναι και ο αριθμός που στιγματίζει την συγκεκριμένη συνεδρία "Session" που πρόκειται να δημιουργήσουμε για να εργαστούμε. Κατά το Login μας πλέον είμαστε ο χρήστης με τα συγκεκριμένα στοιχεία "Credentials" και το συγκεκριμένο "Token". Κάθε φορά που επισκεπτόμαστε τη συγκεκριμένη σελίδα παράγεται ένας νέος "Security Token". Κάθε φορά λοιπόν γίνεται η διαδικασία ελέγχου της εγκυρότητας του "Token" για την αποφυγή όσο είναι αυτό δυνατό κακόβουλων επιθέσεων.

Μία πρώτη ιδέα παράκαμψης αυτής της διαδικασίας ώστε γίνει δυνατή η αυτόματη σύνδεση ήταν η χρήση της ίδιας φόρμας που η Joomla χρησιμοποιεί ώστε να περαστούν μέσω τις εφαρμογής όλα τα απαραίτητα στοιχεία, να γίνει το authentication και να συνδεθούμε.



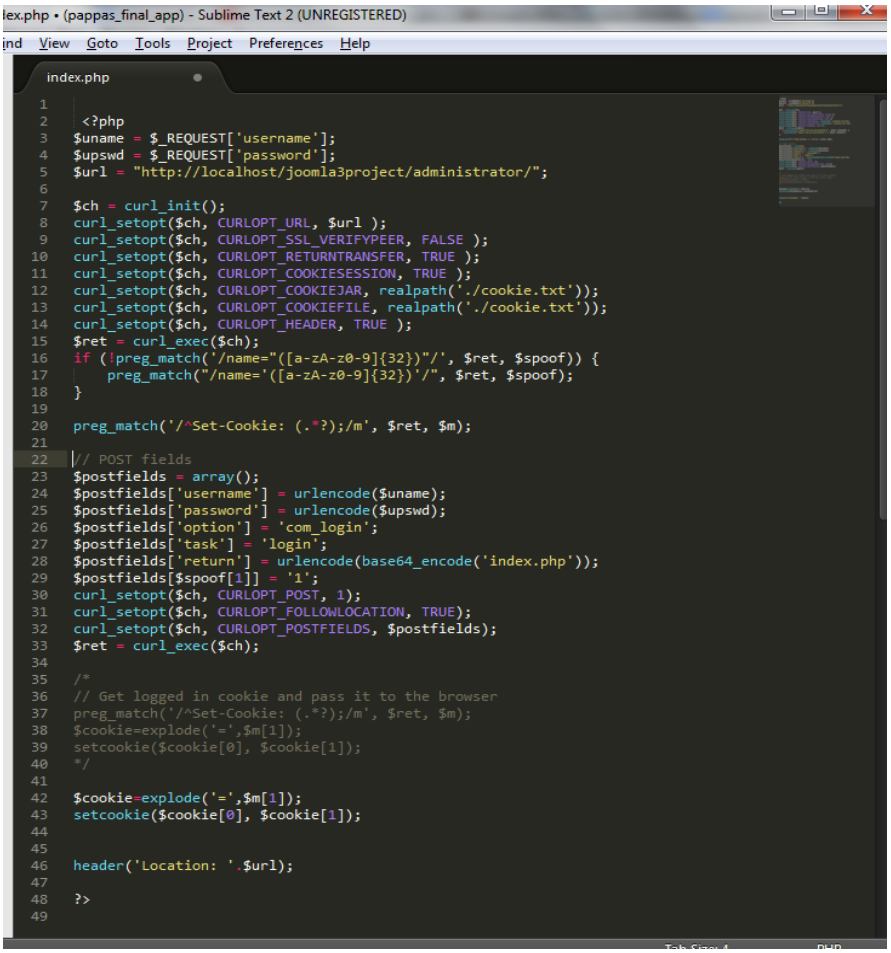
Εικόνα 11 Myproject View

Δηλαδή να μέσω της φόρμας να επιχειρήσουμε ένα http request με μέθοδο post ή get και να στείλουμε στον browser όσα χρειαζόταν. Η αδυναμία που αμέσως προέκυψε ήταν το ότι δεν υπήρχε τρόπος να τραβήξουμε από τη Joomla το Security Token που παρήγαγε κάθε φορά ώστε να της στο στείλουμε πίσω.

Μία άλλη ιδέα ήταν η απλή αποστολή μέσω http των Credentials του διαχειριστή, δηλαδή του username και του password τα οποία είναι αποθηκευμένα στην εφαρμογή μας και απλά να ζητούσαμε ένα Auto Submit ώστε να μην χρειαστεί ο χρήστης να κάνει δεύτερο click. Δεν γίνεται και αυτό μέσω φόρμας γιατί η Joomla δεν το επιτρέπει για λόγους ασφαλείας.

Τρίτη ιδέα για τη λύση του προβλήματος ήταν η υλοποίηση ενός Component της Joomla το οποίο ο χρήστης της εφαρμογής θα ήταν αναγκασμένος να το εγκαταστήσει σε κάθε ένα από τα site του πράγμα που καθιστά τη λύση μη λειτουργική. Επιπλέον αρνητικό είναι ότι σε κάθε Update της Joomla πράγμα που γίνεται συχνά θα πρέπει να γίνεται έλεγχος συμβατότητας και επανεγκατάσταση σε περίπτωση που αυτή δε υπάρχει.

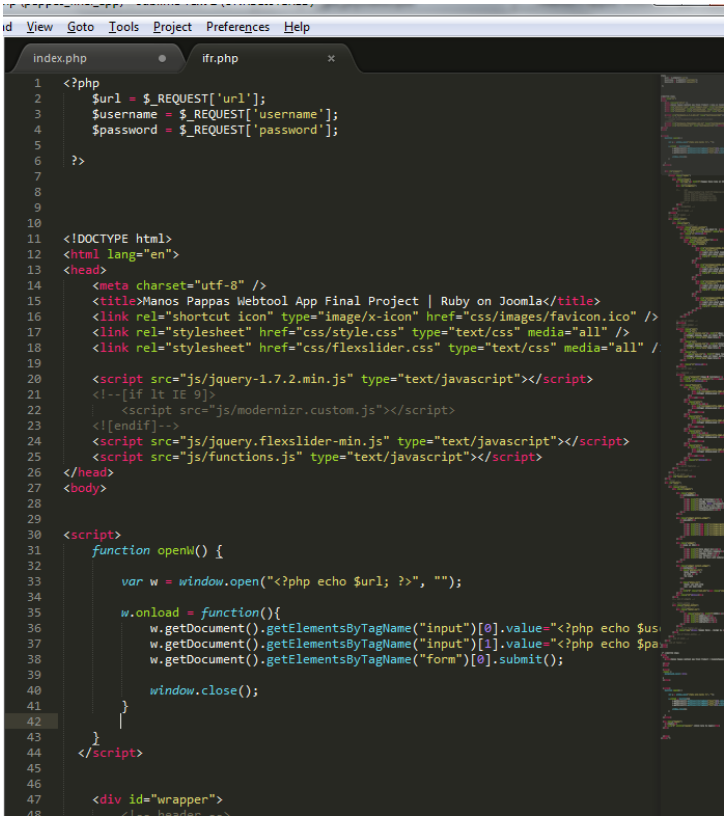
Ο παρακάτω κώδικας που χρησιμοποιήθηκε για την υλοποίηση του component ουσιαστικά μέσω των request του στέλνω τα credentials, χρησιμοποιεί την εντολή curl τραβήξει ότι χρειάζεται από την Joomla, όπως το token και στη συνέχεια χρησιμοποιεί την μέθοδο post για να τα στείλει πίσω στην Joomla. Δεν είχε σαν λύση επιτυχία για το λόγω του ότι τα παραγόμενα tokens, τα οποία όπως είπαμε είναι διαφορετικά σε κάθε επίσκεψη, αποθηκεύονται στη Βάση οπότε πάντα θα υπάρχει η μη εγκυρότητα αυτού,



Εικόνα 12 PHP Script

Η τελευταία και επιτυχημένη ιδέα για να παρακάμψουμε το Login της Joomla και να συνδεθούμε μέσω της εφαρμογής μας προκύπτει από τα εξής. Μέσω της εφαρμογής, χρησιμοποιώντας ένα link με get method τα credentials του χρήστη, σε ένα php script το οποίο φιλοξενείται από τον ίδιο server με την Joomla. Το script αυτό στη συνέχεια χρησιμοποιεί την κατάλληλη javascript την οποία περιέχει λειτουργεί σαν ένα popup window το οποίο ενεργοποιείται όταν ο χρήστης πατήσει το login της εφαρμογής μας και ουσιαστικά κάνει auto complete στη φόρμα της Joomla με τα στοιχεία μας. Το autocomplete που περιέχει ολοκληρώνει τη δουλειά που θέλουμε να κάνουμε. Στο νέο παράθυρο που ανοίγει έχει στηθεί μία ιστοσελίδα.

Ακολουθεί μέρος του κώδικα του script.



Εικόνα 13 PHP Script Εφαρμοσμένο

3.4 Λειτουργικότητα

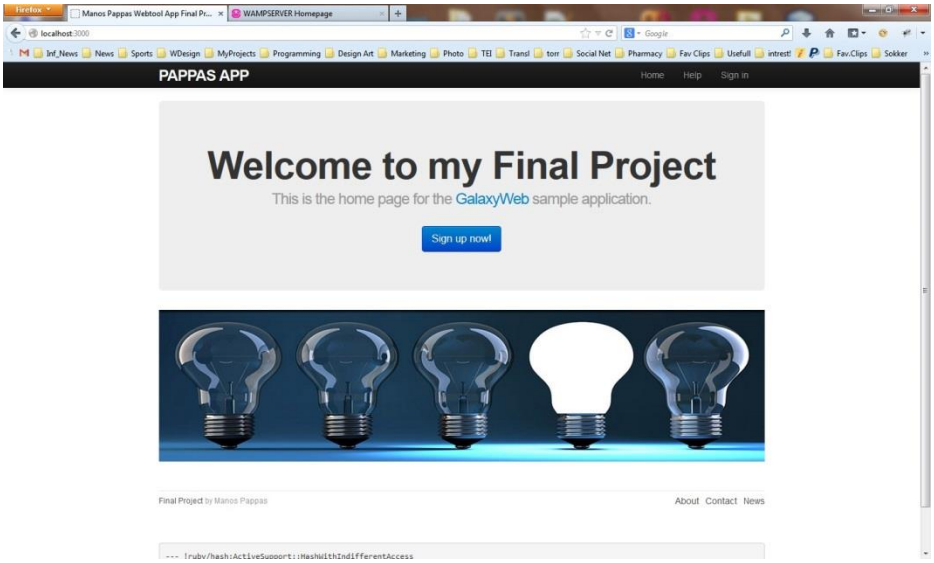
Ο απλός σχεδιασμός της εφαρμογής, δηλώνει και την απλότητα στη χρήση της. Αυτό σημαίνει ότι δεν απευθύνεται μόνο σε προγραμματιστές ή σχεδιαστές ιστοσελίδων που σίγουρα είναι αρκετά εξοικειωμένοι με εφαρμογές αλλά και στον απλό διαχειριστή ακόμα με πολύ βασικές γνώσεις. Ο χρήστης έχει τη

δυνατότητα δημιουργίας λογαριασμού, καθώς και τη δυνατότητα να τον επεξεργαστεί όταν το επιθυμήσει. Επίσης μπορεί να κρατήσεις σημειώσεις σε πολλές από τις σελίδες της εφαρμογής οι οποίες αναρτώνται στην "Home" σελίδα της εφαρμογής με πρώτη στη σειρά την σημείωση που έγραψε τελευταία. Τις σημειώσεις αυτές μπορεί να τις διαχειριστεί σβήνοντας τις καθώς και μία ιδέα είναι να φαίνονται και σε άλλους χρήστες σε μορφή δημοσίευσης. Επιπλέον έχει τη δυνατότητα ανά πάσα στιγμή να δημιουργήσει ένα "Event" στο ημερολόγιο, με όλα τα συμβάντα συγκεντρωτικά να φαίνονται στη σελίδα "Calendar" της εφαρμογής. Μπορεί τέλος να δημιουργήσει και να διαχειριστεί όσα Project επιθυμεί επιλέγοντας την επιλογή "Create Project" από το dropdown menu "Account". Στην σελίδα "My Projects" εμφανίζονται συγκεντρωτικά όλες οι δουλειές που ο χρήστης έχει αποθηκεύσει στη Βάση δεδομένων της εφαρμογής.

Πρώτο μέλημα του εγχειρήματος ήταν ο εκάστοτε διαχειριστής δημιουργώντας το λογαριασμό του στην εφαρμογή και επιλέγοντας το κωδικό εισόδου σε αυτή, να είναι ο μόνο κωδικός που θα χρειάζεται να θυμάται στο μέλλον. Δημιουργείται η δυνατότητα λοιπόν να συνδεθεί σε οποιοδήποτε από τα αποθηκευμένα στην εφαρμογή περιβάλλοντα που διαχειρίζεται χωρίς να χρειάζεται να θυμάται όλους αυτού τους κωδικούς. Κατά τη δημιουργία του κάθε "Project" αποθηκεύεται στη Βάση Δεδομένων ο κωδικός όπως και πολλές άλλες πληροφορίες σχετικά με το συγκεκριμένο έργο.

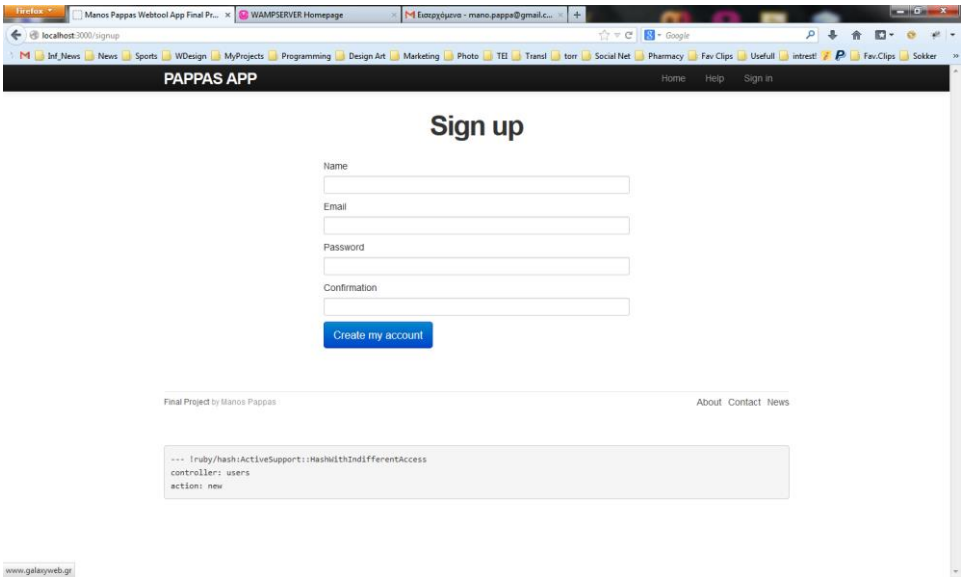
Μέσο του ημερολογίου αυτό που επιχειρείται να γίνει είναι να λειτουργεί η εφαρμογή ως γραμματειακή του εκάστοτε χρήστη θυμίζοντάς του όλες τις εκκρεμότητες που φτάνουν στη λήξη τους φροντίζοντας να μην ξεχαστεί καμία από αυτές.

Κατά την επίσκεψή μας η εφαρμογή μας ζητά είτε να κάνουμε login εάν είμαστε ήδη χρήστες είτε να εγγραφούμε εάν τον επισκεπτόμαστε πρώτη φορά.



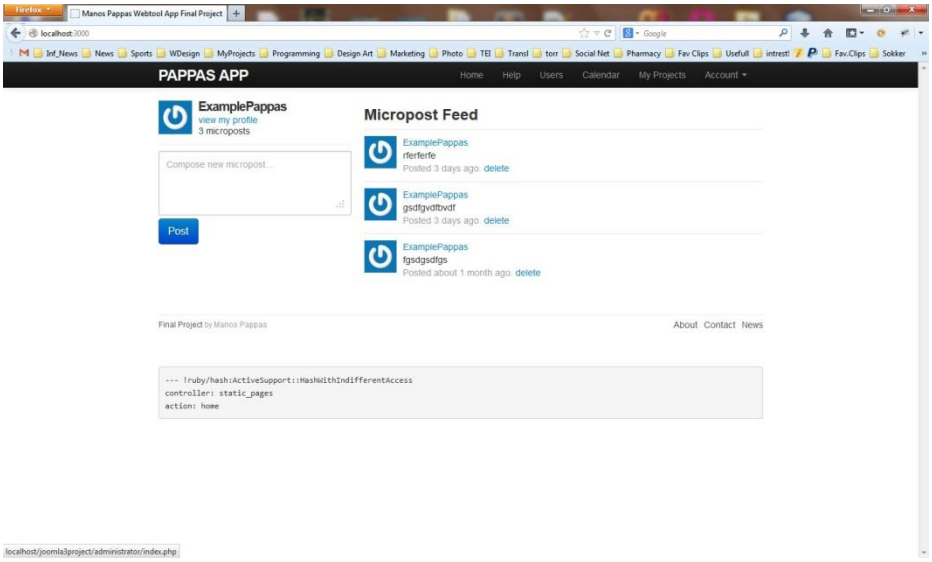
Εικόνα 14 Home Page

Μία φόρμα εγγραφής μας εμφανίζεται εάν πατήσουμε το Sign up now!



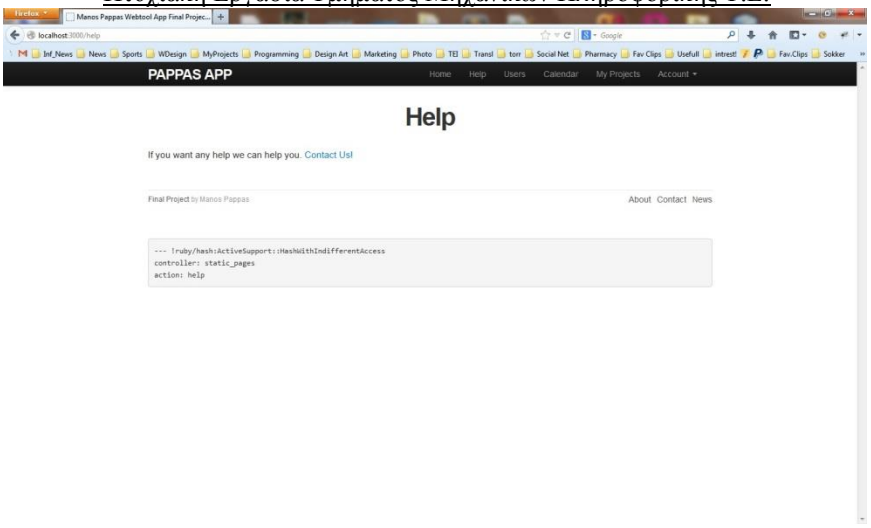
Εικόνα 15 Sugn Up Page

Το home page της εφαρμογής μας δίνει επιλογές όπως το να κρατήσουμε μία σημείωση καθώς και μας παρουσιάζει και το πλήθος των σημειώσεων που έχουμε ήδη κρατήσει.



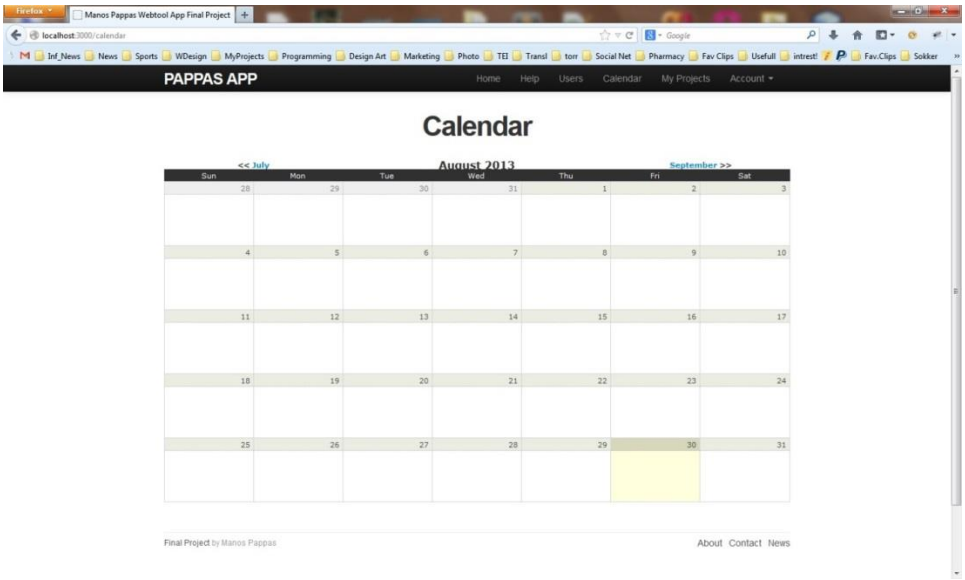
Εικόνα 16 Home Page Signed In

Υπάρχει μία σελίδα η οποία θα περιέχει ένα σύνολο από άρθρα με σκοπό τη βοήθεια των χρηστών.



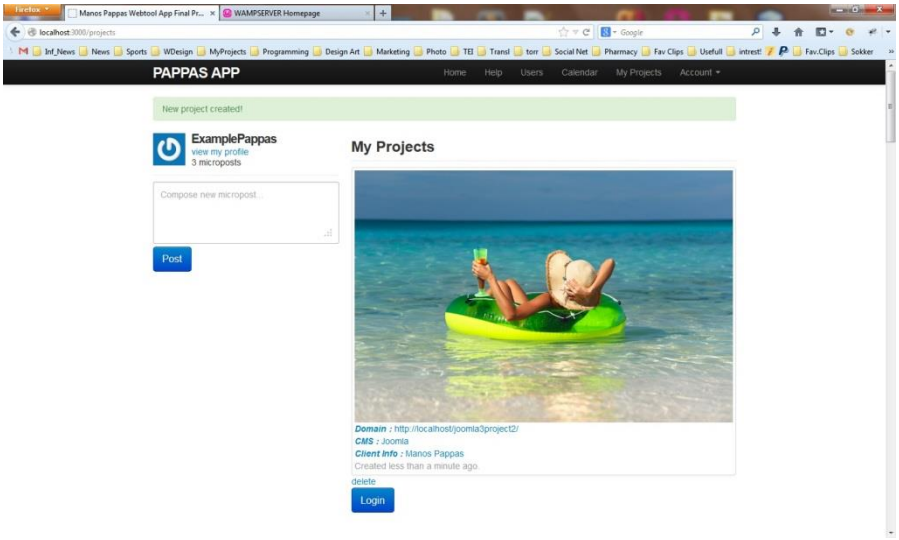
Εικόνα 17 Help Page

Το Calendar θα είναι μία σελίδα όπου ο κάθε χρήστης θα μπορεί να δημιουργεί events. Αυτά τα events θα παρουσιάζονται με όμορφο τρόπο στο ημερολόγιο που έχει σχεδιαστεί εκεί. Κατά τη δημιουργία του event στην εφαρμογή θα δημιουργείται ταυτόχρονα το ίδιο event και στο Google Calendar του χρήστη για περεταίρω διευκόλυνση καθώς επίσης η εφαρμογή θα είναι υπεύθυνη να ενημερώνει τον χρήστη μέσω ενός email όταν το event φτάνει στη λήξη του.

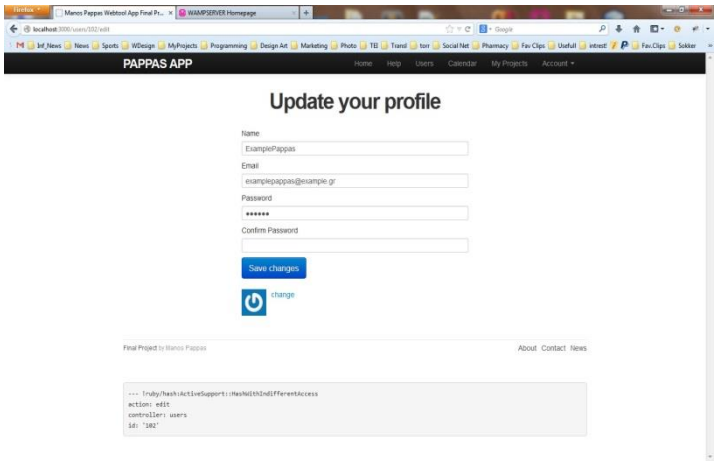


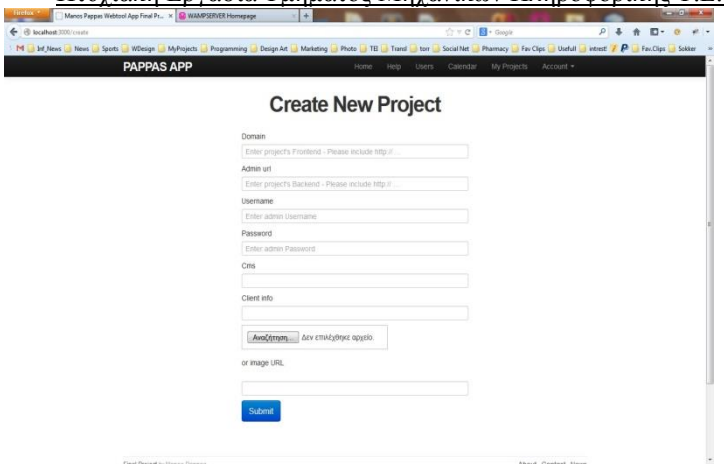
Εικόνα 18 Calendar Page

Στη σελίδα Myprojects συγκεντρώνονται τα projects που ο χρήστης έχει αποθηκεύσει στη Βάση της εφαρμογής σελίδα από την οποία θα μπορεί να συνδεθεί στις αντίστοιχες σελίδες διαχείρισής τους μέσω ενός button. Επίσης θα μπορεί αν κρατάει σημειώσεις.



Κάτω από το μενού Account βρίσκονται τα μενού των σελίδων Profile το οποίο συγκεντρώνει τα projects και τις σημειώσεις του χρήστη, η Settings στην οποία ο χρήστης μπορεί να αλλάξει τα στοιχεία του, το Create Project από όπου δημιουργεί και αποθηκεύει τα νέα του project καθώς και το sign out ώστε αν αποσυνδεθεί.





3.5 Ασφάλεια

Sessions

Η Rails χρησιμοποιεί τον όρο συνεδρία χρήστη (session) για να μπορέσει να προσδιορίσει τις ενέργειές – αιτήσεις που ανήκουν σε κάθε χρήστη. Χωρίς τις συνεδρίες θα έπρεπε να προσδιορίζουμε την ταυτότητα του χρήστη για κάθε αίτηση. Η Rails δημιουργεί αυτόματα μια συνεδρία αν κάποιος μπει στο σύστημα ή φορτώνει την υπάρχουσα συνεδρία αν ο χρήστης χρησιμοποιεί ήδη την εφαρμογή.

Μια συνεδρία αποτελείτε από ένα hash και ένα αριθμό id με 32 χαρακτήρες. Κάθε μπισκότο (cookie) που στέλνετε στο brower του πελάτη περιέχει το id της συνεδρίας , ο brower το στέλνει πίσω σε κάθε αίτημα προς τον server έτσι εξασφαλίζετε η μοναδικότητα της. Σο id της συνεδρίας αποτελείτε από τυχαίες τιμές που λαμβάνονται από την τρέχουσα ώρα με τυχαίους αριθμούς 0 ή 1 και μέσα στον επεξεργαστή της ruby μεταφράζονται σε μια συμβολοσειρά . Μέχρι σήμερα ο MD5 είναι ασυμβίβαστες οι συμβολοσειρές που παράγει, θεωρητικά είναι δυνατόν να παραχθούν όμοια hash με απειροελάχιστη πιθανότητα και δεν έχει καταγραφεί καμία επίθεση από αυτό το γεγονός έως τώρα.

Η εφαρμογή μας πέρα από το μηχανισμό authentication τον οποίο διαθέτει, κάθε κωδικός ο οποίος αποθηκεύεται στη βάση κωδικοποιείται μέσω του bcrypt-ruby που χρησιμοποιεί SHA1μηχανισμό.

3.6 Πλεονεκτήματα εφαρμογής

Ο λιτός σχεδιασμός της εφαρμογής μαρτυρά και το βαθμό της ευχρηστίας της. Εξαρχής ο στόνος ήταν το να μπορούν το χειριστούν όσο το δυνατόν περισσότεροι άνθρωποι είτε αυτοί είχαν περισσότερη εξοικείωση με τους υπολογιστές είτε λιγότερη.

Από την άλλη η Ruby on Rails είναι μια πλήρης γλώσσα με αρκετά χρόνια στο χώρο που τη καθιστά ώριμη γλώσσα. Έχει άμεση επαφή με HTML, Javascript και CSS. Παρέχει σημαντική ασφάλεια είτε μέσω των μηχανισμών αλλά ακόμα γιατί η εφαρμογή μπορεί να δουλέψει μία χαρά και τοπικά (localhost). Το μεγαλύτερο όμως μέρος της δύναμης της Ruby (και κατά συνέπεια και του RoR) προέρχεται από τον ίδιο της τον εαυτό, μιας και είναι μια πολύ ευέλικτη high level γλώσσα, που δίνει την δυνατότητα στον προγραμματιστή να γράψει κώδικα όχι με βάση την γλώσσα αλλά με βάση τον τρόπο που ο ίδιος θα επιλέξει. Το framework ruby on rails αν και στην αρχή κάποιον ίσως να μην τον ενθουσιάσει ιδιαίτερα, όταν όμως εξοικειωθεί μαζί του θα διαπιστώσει ότι μπορεί να αναπτύξει πολύ "δυνατές" εφαρμογές με πολύ λίγο κόπο.

Μπορεί να καλύψει τις ανάγκες για οργάνωση ενός γραφείου αφού μέσω της εφαρμογής χρησιμοποιώντας το Calendar καθώς και τις σημειώσεις μπορούν να γίνουν αναθέσεις μέρους των επιμέρους έργων. Εννοώ ότι ο υπεύθυνος της επιχείρησης μπορεί να τη χρησιμοποιήσει ως εργαλείο υπενθύμισης αρμοδιοτήτων μέσα στο γραφείο.

ΚΕΦΑΛΑΙΟ 4ο : ΣΥΜΠΕΡΑΣΜΑ

4.1 Τελικό συμπέρασμα

Συνεχώς αυξανόμενος είναι ο αριθμός αυτών που χρειάζονται μία ή και περισσότερες ιστοσελίδες, είτε για να δημιουργήσουν το προσωπικό τους blog, είτε για να παρουσιάσουν τη δουλειά τους (portfolio) είτε για να παρουσιάσουν την επιχείρησή τους. Επίσης νέα ηλεκτρονικά καταστήματα ολοένα και εμφανίζονται στη αγορά.

Από προσωπική εμπειρία στην κατασκευή ιστοσελίδων, μεγάλο μέρος της ενέργειας καθημερινά καταναλώνεται στο να οργανώσεις τη μέρα σου. Να σημειώσεις σε πολλά σημεία τα σημαντικά ζητήματα της ημέρας. Αναλαμβάνοντας λοιπόν πολλά project μειώνεται η παραγωγικότητά σου όταν προσπαθείς να τα συγκρατείς όλα στο μυαλό σου ώστε να μην χάσεις κάποια ημερομηνία.

Αυτό ήταν και το βασικό κίνητρο που με έκανε να ασχοληθώ με το συγκεκριμένο project.

Τελειώνοντας και το τελευταίο κομμάτι τις πτυχιακής, είμαι σε θέση να πω πως έχω αποκτήσει αρκετά μεγάλη πείρα στην ανάπτυξη μιας WEB εφαρμογής. Αυτό ήταν αποτέλεσμα των σίγουρων, σταθερών και όχι επιπόλαιων βημάτων από την στιγμή που άρχισα να δουλεύω πάνω στην πτυχιακή. Τουλάχιστον προς το τέλος αυτής.

Μην έχοντας άλλη επαφή με την γλώσσα, παρά μόνο από την στιγμή που ξεκίνησε η ενασχόλησή μου με την πτυχιακή, έπρεπε να αφιερώσω αρκετό χρόνο στην σωστή εκμάθηση της γλώσσας Ruby, ώστε να αποφευχθούν συντακτικά λάθη και λάθη που αφορούσαν την δομή της γλώσσας. Έπειτα από τρεις μήνες διαβάσματος βιβλίων που αφορούσαν την Ruby και νιώθοντας ότι έχω αποκτήσει δυνατές βάσεις, ξεκίνησε η ανάπτυξη της εφαρμογής.

Δεν μπορώ να μην αναφερθώ στην πολύτιμη βοήθεια του επιβλέποντα καθηγητή, που με τις παρατηρήσεις του, μου έδειξε πως θα μπορεί να δομηθεί μια WEB εφαρμογή.

Μέτα την περάτωση της πτυχιακής και έχοντας αποκτήσει ικανοποιητική πείρα μπορώ να πω, πως η Ruby και το Web framework Ruby On Rails είναι δύο πολύ δυνατά εργαλεία για τον οποιονδήποτε θέλει να ασχοληθεί με το Web programming. Μεγάλο μέρος της δύναμης αυτής αντλείται από την μεγάλη κοινότητα των προγραμματιστών Ruby, που έχουν αναπτύξει έναν τεράστιο αριθμό από gems (βιβλιοθήκες) και της open source φιλοσοφίας που έχει ενστερνιστεί η γλώσσα και κατά συνέπεια και η κοινότητα των προγραμματιστών που την χρησιμοποιεί.

Το μεγαλύτερο όμως μέρος της δύναμης της Ruby (και κατά συνέπεια και του RoR) προέρχεται από τον ίδιο της τον εαυτό, μιας και είναι μια πολύ ευέλικτη high level γλώσσα, που δίνει την δυνατότητα στον προγραμματιστή να γράψει κώδικα όχι με βάση την γλώσσα αλλά με βάση τον τρόπο που ο ίδιος θα επιλέξει. Το framework ruby on rails αν και στην αρχή κάποιον ίσως να μην τον ενθουσιάσει ιδιαίτερα, όταν όμως εξοικειωθεί μαζί του θα διαπιστώσει ότι μπορεί να αναπτύξει πολύ “δυνατές” εφαρμογές με πολύ λίγο κόπο.

Βιβλιογραφία

1. “Beginning ruby, from novice to professional”, Peter Cooper, Μάρτιος 2006 APRESS

2. “Ruby On Rails, Enterprice application development” Elliot Smith & Rob Nichols, Οκτώβριος 2007, PACKT

3. “Agile web development with rails. third edition” Sam Ruby, Dave Thomas & David Heinemeier Hansson, Ιούνιος 2008, Pragmatic Bookself

4. “Practical rest on rails projects 2”, Ben Scofield, Απρίλιος 2008, APRESS

5. “Simply rails 2”, Patrick Lenz, Μάιος 2008, SitePoint

6. “The art of rails”, Edward Benson, Μάιος 2008, WROX

7. “The rails way”, ObieFernandez, Νοέμβριος 2007, Addison-Wesley

8. "Εργαστήριο - Ειδικά θέματα Βάσεων Δεδομένων.", Γ. Γκαράνη, ΤΕΙ Λάρισας

9. "Εισαγωγή στην HTML και τα CSS.", Χ. Κόπανος, ΤΕΙ Λάρισας

ΠΑΡΑΡΤΗΜΑ : ΕΓΚΑΤΑΣΤΑΣΗ ΕΡΓΑΛΕΙΩΝ ΕΦΑΡΜΟΓΗΣ

i. Εγκατάσταση του Framework Ruby on Rails

Το να εγκαταστήσεις την Rails στα Windows ήταν παλιότερα μια αρκετά επώδυνη διαδικασία. Ακολουθεί παρουσίαση της εγκατάστασης του Framework βήμα βήμα:

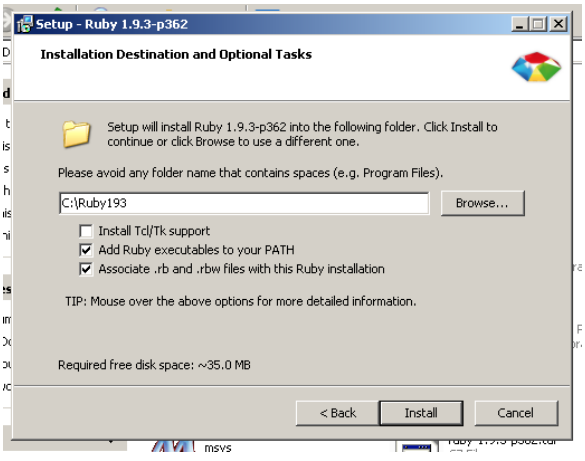
a) Πρώτο βήμα είναι να εγκαταστήσουμε έναν Ruby Version Manager (RVM) ο οποίος θα μας επιτρέψει να διαχειριστούμε διαφορετικές εκδόσεις της Ruby καθώς και να ελέγξουμε και να δοκιμάσουμε μια νέα έκδοση στο ήδη υπάρχον project μας αφήνοντας το ανέπαφο.

```
$ curl -L https://get.rvm.io | bash -s
```

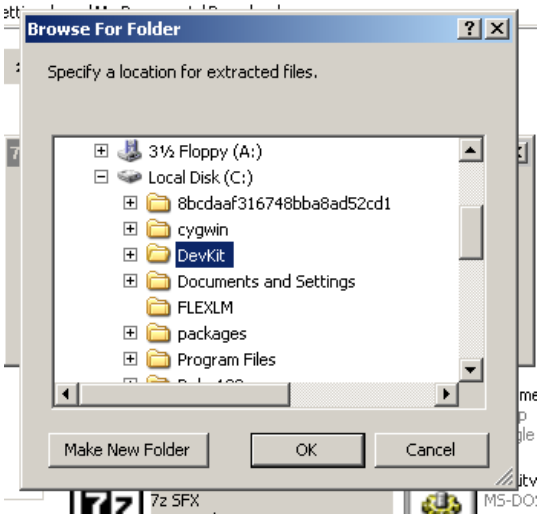
b) Το δεύτερο βήμα είναι να εγκαταστήσουμε τη Ruby την οποία θα πρέπει να κατεβάσουμε στην τελευταία έκδοσή της με τον installer.

"Στην εφαρμογή έχω χρησιμοποιήσει Ruby 1.9.3 και αυτή θα υποδείξω στη συνέχεια."

Κατεβάζουμε το αρχείο από επίσημη ιστοσελίδα http://rubyinstaller.org/downloads/ κι αφού έχει κατέβει με διπλό κλικ ξεκινά ο installer.



c) Πρέπει επίσης να κατεβάσουμε και τον DevKit από την ίδια τοποθεσία http://rubyinstaller.org/downloads/ . Πατώντας διπλό κλικ στο αρχείο θα μας ζητηθεί το path στον υπολογιστή μας στο οποίο θα εξάγει τα αχρεία. Δημιουργήστε έναν νέο φάκελο με το όνομα "C:\DevKit"



d) Τώρα θα πρέπει να ανοίξουμε το command prompt και να εισάγουμε τις ακόλουθες εντολές.

> chdir C:\DevKit

> ruby dk.rb init

> ruby dk.rb install

e) Εξετάζοντας πλέον αν έχουμε εγκατεστημένα τα gems (βιβλιοθήκες της Ruby)

```
$ which gem
/Users/mhartl/.rvm/rubies/ruby-2.0.0-p0/bin/gem
```

Στη περίπτωση που δεν υπάρχουν στο σύστημα θα χρειαστεί να τα κατεβάσουμε http://rubyforge.org/frs/?group_id=126 να τα εξάγουμε στο δίσκο μας και στη συνέχεια να τρέξουμε το πρόγραμμα εγκατάστασης

```
$ ruby setup.rb
```

Αν είναι ήδη εγκατεστημένα πρέπει να βεβαιωθούμε ότι το σύστημά μας χρησιμοποιεί τη τελευταία έκδοση.

```
$ gem update --system 2.0.3
```

f) Τελευταίο βήμα είναι να εγκαταστήσουμε το Rails

```
$ gem install rails --version 4.0.0
```

Ο έλεγχος για την έκδοση του ήδη εγκατεστημένου είναι.

```
$ rails -v
Rails 4.0.0
```

ii. Εγκατάσταση των CMS (Joomla)

Στην επόμενη ενότητα θα εξεταστεί ο τρόπος εγκατάστασης του web server WAMP τοπικά (localhost) η οποία πρέπει να γίνει πρώτα ώστε να έχουμε τη δυνατότητα να εγκαταστήσουμε τη Joomla.

Ξεκινώντας λοιπόν θα πρέπει να κατεβάσουμε την έκδοση της Joomla που επιθυμούμε από http://www.joomla.org/download.html σε zip αρχείο το οποίο πρέπει να εξάγουμε στο root (www) φάκελο του Web server (Wamp) σε έναν νέο φάκελο που θα δημιουργήσουμε το όνομα του οποίου θα χρησιμοποιήσουμε ως domain για να συνδεόμαστε στη Joomla.

Ανοίγοντας λοιπόν τον browser πληκτρολογούμε στη γραμμή διεύθυνσης Url το domain στο οποίο βρίσκεται η Joomla που θέλουμε να εγκαταστήσουμε.

Για παράδειγμα : http://localgost/myFisrtJoomla

a) Στην πρώτη οθόνη εγκατάστασης εμφανίζονται οι διαθέσιμες γλώσσες εγκατάστασης.

b) Στη δεύτερη οθόνη το Joomla κάνει κάποιους ελέγχους σχετικά με τη συμβατότητα του συστήματός μας. Στην παρακάτω ομάδα είναι απαραίτητο να είναι όλες οι ενδείξεις NAI. Σε περίπτωση που όλη η επάνω ομάδα είναι NAI, προχωράμε στο επόμενο βήμα κάνοντας κλικ στο κουμπί Επόμενο. Σε περίπτωση που κάποιο από αυτά είναι με κόκκινο χρώμα, το διορθώνουμε και πατάμε Επανέλεγχος.

c) Στην επόμενη οθόνη εμφανίζεται η Άδεια Χρήσης GNU/GPL που χρησιμοποιεί το Joomla. Κάνουμε κλικ στο κουμπί Επόμενο.

d) Είμαστε στο πιο σημαντικό βήμα. Εδώ πρέπει να εισαγάγουμε τα στοιχεία της βάσης δεδομένων με την οποία θα συνεργάζεται το Joomla. Εμφανίζεται η παρακάτω οθόνη. Το όνομα διακομιστή παραμένει localhost αφού είμαστε τοπικά, το όνομα χρήστη root και ο κωδικός κενός. Στο όνομα Βάσης Δεδομένων εισάγουμε το όνομα της Βάσης που δημιουργήσαμε στο phpmyadmin (web server).

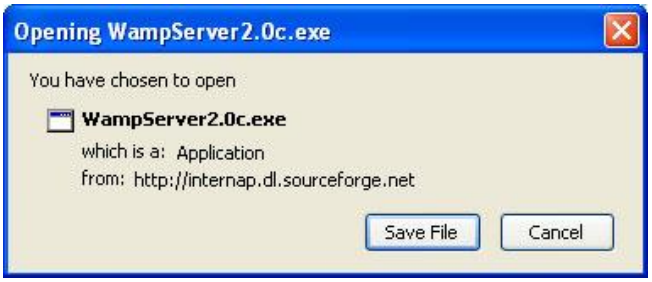
e) Για λόγους ασφαλείας και προστασίας των αρχείων που χρησιμοποιεί το Joomla εδώ μας δίνεται η δυνατότητα δημιουργίας ενός FTP (File Transfer Protocol) λογαριασμού. Θα το προσπεράσουμε κάνοντας κλικ στο κουμπί Επόμενο.

f) Στην επόμενη οθόνη χρειάζεται να συμπληρώσουμε κάποια πεδία όπως το Όνομα του ιστοτόπου, τη διεύθυνση email μας, τον όνομα και το κωδικό διαχειριστή. Επίσης μπορούμε να εισάγουμε προεπιλεγμένο περιεχόμενο στο front-end της ιστοσελίδας μας.

g) Τελευταία είναι η οθόνη στην οποία η Joomla μας ενημερώνει ότι η εγκατάσταση ολοκληρώθηκε δίνοντας μας παράλληλα την οδηγία να σβήσουμε το φάκελο installation από το root.

iii. Εγκατάσταση web server (Wamp) τοπικά

Κατεβάζουμε το πρόγραμμα εγκατάστασης από την επίσημη ιστοσελίδα http://www.wampserver.com/en/ στον υπολογιστή μας και τρέχουμε το exe αρχείο.

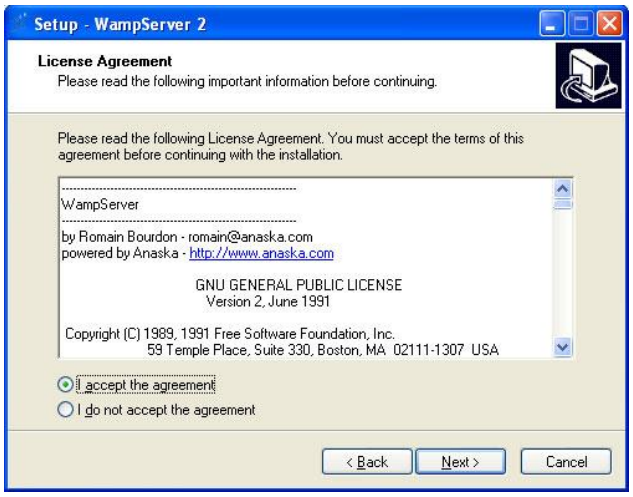


Εικόνα 19 Εγκατάσταση Wamp Server

Επιλέγουμε Run

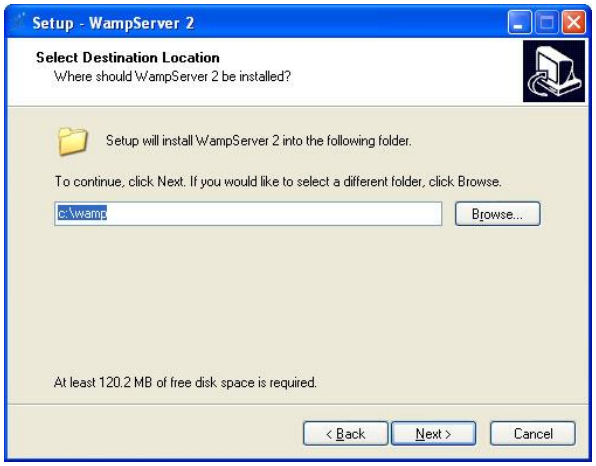
Εικόνα 20 Εγκατάσταση Wamp Server

Εικόνα 21 Εγκατάσταση Wamp Server

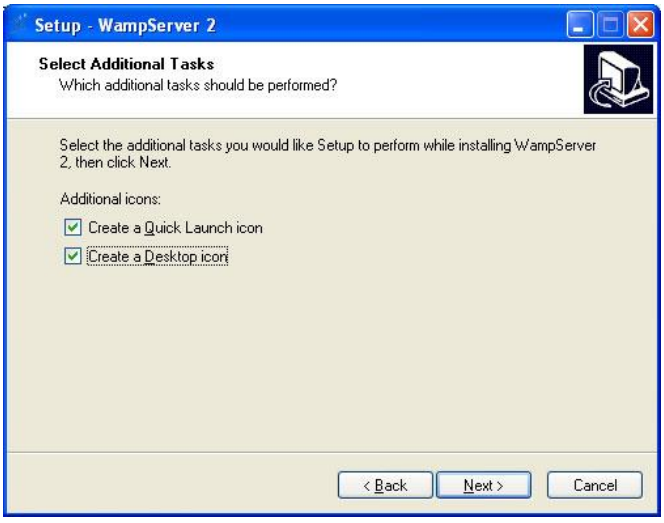


Εικόνα 22 Εγκατάσταση Wamp Server

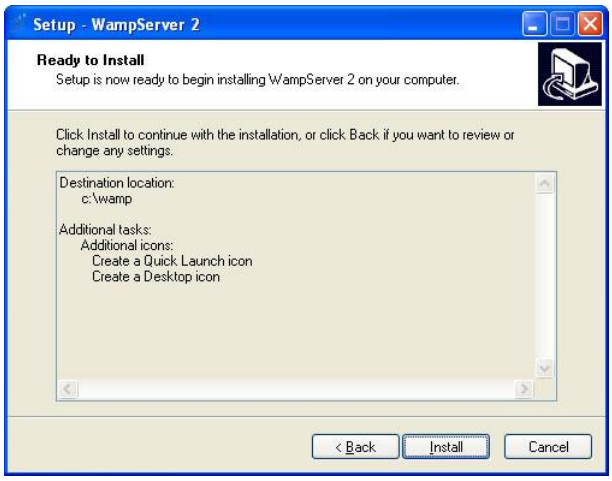
Επιλογή path εγκατάστασης



Εικόνα 23 Εγκατάσταση Wamp Server

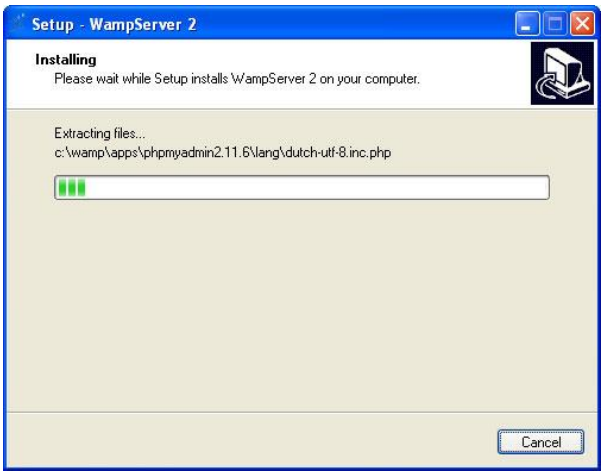


Εικόνα 24 Εγκατάσταση Wamp Server



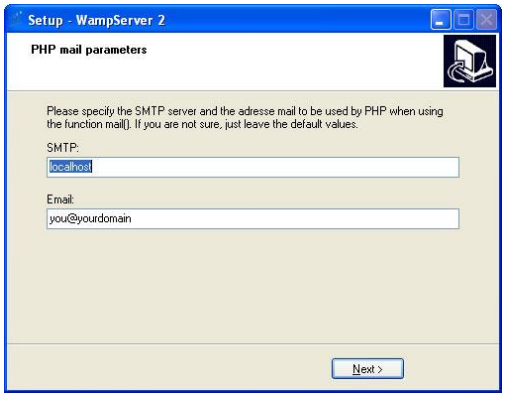
Εικόνα 25 Εγκατάσταση Wamp Server

Εγκατάσταση λογισμικού



Εικόνα 26 Εγκατάσταση Wamp Server

Ρύθμιση του διακομιστή και του λογαριασμού email



Εικόνα 27 Εγκατάσταση Wamp Server

Ολοκλήρωση της εγκατάστασης

Εικόνα 28 Εγκατάσταση Wamp Server

Εικόνα 29 Εγκατάσταση Wamp Server

Μενού του Wamp server μετά την ολοκλήρωση της εγκατάστασής του. Η παρούσα εφαρμογή χρησιμοποιεί Wamp Server 2.4 ο οποίος περιλαμβάνει:

- Apache server 2.4.4
- MySQL server 5.6.16
- PHPMyAdmin 4.0.4