



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών σε ενδιαφερόμενους από επαγγελματίες με χρήση οπτικού και ακουστικού υλικού.

Γιώργος-Ραφαήλ Ρέντης

Εισηγητής: Παναγιώτης Γιαννακόπουλος, Καθηγητής

**ΑΘΗΝΑ
ΙΑΝΟΥΑΡΙΟΣ 2016**

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξης διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών σε ενδιαφερόμενους από επαγγελματίες με χρήση οπτικού και ακουστικού υλικού

**Γιώργος-Ραφαήλ Ρέντης
Α.Μ. 40566**

**Εισηγητής:
Παναγιώτης Γιαννακόπουλος, Καθηγητής**

Εξεταστική επιτροπή:

Ημερομηνία εξέτασης:

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Γιώργος – Ραφαήλ Ρέντης, του Κωνσταντίνου, με αριθμό μητρώου 40566 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω ιδιαίτέρως τον καθηγητή κ. Γιαννακόπουλο για τη βοήθεια που παρείχε στην πραγματοποίηση της παρούσας εργασίας, επίσης θα ήθελα να ευχαριστήσω τον κ. Τέμενο για τη βοήθεια του. Τέλος θα ήθελα να ευχαριστήσω τον πατέρα μου.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

ΠΕΡΙΛΗΨΗ

Ο σκοπός της παρούσας πτυχιακής εργασίας, είναι η ανάπτυξη διαδικτυακής εφαρμογής με στόχο την παροχή γνώσεων και πληροφοριών από επαγγελματίες προς ενδιαφερόμενους χρήστες. Η λειτουργία της εφαρμογής βασίζεται στην ανάρτηση εκπαιδευτικού υλικού (π.χ. παρουσιάσεις, pdf) από επικυρωμένους επαγγελματίες στο αντικείμενο τους (π.χ. προγραμματιστές, φυσικούς) στο οποίο θα έχουν πρόσβαση οι ενδιαφερόμενοι χρήστες. Για διευκόλυνση των χρηστών θα υπάρχει η δυνατότητα αξιολόγησης του υλικού αυτού, έτσι ώστε να γνωρίζουν και οι υπόλοιποι χρήστες το επίπεδο γνώσης του επαγγελματία για μελλοντικές χρήσεις.

ABSTRACT

The goal of the present thesis is the development of a web application that provides knowledge and information from professionals to anyone interested in the subject. The functionality of this application is based on the upload of educational resources (PowerPoint presentations, pdf) from certified professionals in their field (programmers, physicists) in which all of the interested users have access to. For the comfort of the users a rating system of the content is accessible to use so other users can know the educational level of the professional for future reference.

Επιστημονική περιοχή: Προγραμματισμός

Λέξεις κλειδιά: e-learning, ruby, rails, μοντέλα, χειριστές

Scientific area: Programming

Keywords: e-learning, ruby, rails, models, controllers

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	11
ΚΕΦΑΛΑΙΟ 1.....	13
ΕΙΣΑΓΩΓΗ	13
1.1 Περιγραφή του αντικειμένου της πτυχιακής.....	13
1.2 Ιστορική αναδρομή	14
1.2.1 Στην αρχαιότητα	14
1.2.2 Η πρώτη μορφή μάθησης από απόσταση	15
1.2.3 Η μηχανή του Pressey	15
1.2.4 Εκπαίδευση κατά τη διάρκεια του 2 ^{ου} παγκοσμίου πολέμου.....	18
1.2.5 Μετά τον πόλεμο	19
1.2.6 Τα τέλη του 20 ^{ου} αιώνα και το παρόν.....	20
1.3 Το μέλλον του e-learning.....	20
1.4 Συγχρονισμένο και ασύγχρονο e-learning	23
1.5 Συστήματα διαχείρισης μάθησης (Learning management systems - LMS) .	24
1.6 Massive open online courses (MOOC).....	24
1.7 Οι κύριες τάσεις του e-learning.....	24
1.7.1 Ανάμεικτη μάθηση (Blended learning)	25
1.7.2 Κοινωνική και συνεργατική εκπαίδευση (social and collaborative learning) ..	25
1.7.3 Παιχνιδοποίηση (Gamification)	26
1.8 Στατιστικά στοιχεία σχετικά με το e-learning	26
1.9 Υπηρεσίες που θα χρησιμοποιηθούν στην πτυχιακή εργασία	30
1.10 Τεχνολογίες που χρησιμοποιούνται στο e-learning.....	32
ΚΕΦΑΛΑΙΟ 2.....	33
ΕΙΣΑΓΩΓΗ	33
2.1 Η γλώσσα Ruby	33
2.2 Η Ruby On Rails	35
2.3 Η αρχιτεκτονική MVC (Model-View –Controller).....	36
2.4 Ανάλυση των M-V-C.....	37
2.5 Τα Ruby Gems.....	37

2.6 Τα εργαλεία της Rails	38
2.7 Associations μεταξύ των μοντέλων στη Rails	39
2.7.1 belongs_to association	40
2.7.2 has_many association	41
2.7.3 has_many :through association	42
2.8 Βασικές εντολές της rails & δημιουργία ενός project	43
ΚΕΦΑΛΑΙΟ 3.....	53
ΕΙΣΑΓΩΓΗ	53
3.1 Οι δυνατότητες των χρηστών της εφαρμογής	53
3.2 Το σχέδιο της βάσης δεδομένων.....	55
3.3 Ανάλυση της εφαρμογής.....	55
3.3.1 Η αρχική σελίδα	56
3.3.2 Η σελίδα εγγραφής χρήστη.....	59
3.3.3 Το προφίλ του χρήστη	64
3.3.4 Σύνδεση του admin.....	65
3.3.5 Η επιλογή Lessons	68
3.3.6 Η επιλογή Categories	70
3.3.7 Δημιουργία Μαθήματος	71
3.4 Τα test της εφαρμογής.....	72
3.4.1 Test για τα validations του μοντέλου User	73
3.4.2 Τεστ πραγματικών σεναρίων χρήσης της εφαρμογής.....	74
ΕΠΙΛΟΓΟΣ.....	78
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	79

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Εικόνα 1.1: Τάξη στην αρχαία Ρώμη.....	14
Εικόνα 1.2: Η μηχανή του Pressey.....	16
Εικόνα 1.3: Παράδειγμα τεστ μέσω Questionmark Perception.....	17
Εικόνα 1.4: Η teaching machine του BF Skinner.....	19
Εικόνα 2.1: Στατιστικά στοιχεία της TIOBE software	34
Εικόνα 2.2: Αρχιτεκτονική MVC	36
Εικόνα 2.3α: Παράδειγμα belongs_to association	40
Εικόνα 2.3β: Παράδειγμα has_many association.....	41
Εικόνα 2.3γ: Παράδειγμα has_many, :through association.....	42
Εικόνα 2.4α: Δημιουργία του βασικού φακέλου της εφαρμογής.....	43
Εικόνα 2.4β: Κατασκευή μοντέλου	44
Εικόνα 2.4γ: Κατασκευή του controller.....	46
Εικόνα 2.4δ: Δημιουργία της βάσης δεδομένων.....	47
Εικόνα 2.4ε: Μεταφορά των στοιχείων στη βάση	48
Εικόνα 2.4στ: Έναρξη του τοπικού server	49
Εικόνα 2.4ζ: Έναρξη της κονσόλας.....	50
Εικόνα 2.5: Ένα rails project	51
Εικόνα 3.1: Δυνατότητες επισκέπτη	53
Εικόνα 3.2: Δυνατότητα απλού χρήστη	54
Εικόνα 3.3: Δυνατότητες του διαχειριστή.....	54
Εικόνα 3.4: Το σχηματικό της βάσης δεδομένων	55
Εικόνα 3.5: Αρχική σελίδα της εφαρμογής	56
Εικόνα 3.6: Φόρμα επικοινωνίας Contact Us	57
Εικόνα 3.7: Η FAQ σελίδα.....	58
Εικόνα 3.8: Σελίδα εγγραφής χρήστη.....	59
Εικόνα 3.9: Το προφίλ του χρήστη.....	64
Εικόνα 3.10: Το admin panel.....	65
Εικόνα 3.11: Εμφάνιση όλων των λογαριασμών της εφαρμογής	66
Εικόνα 3.12: Λίστα με τις κατηγορίες της εφαρμογής	67

Εικόνα 3.13: Λίστα με όλα τα μαθήματα.....	68
Εικόνα 3.14: Σελίδα ενός μαθήματος	69
Εικόνα 3.15: Δημόσιο προφίλ χρήστη	69
Εικόνα 3.16: Οι κατηγορίες της εφαρμογής.....	70
Εικόνα 3.17: Φόρμα δημιουργίας μαθήματος.....	71
Εικόνα 3.18: Αποτελέσματα των validation tests.....	74
Εικόνα 3.19: Αποτελέσματα των test χρήσης	77

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Στο 1^ο κεφάλαιο αναφέρονται στοιχεία σχετικά με την ηλεκτρονική εκπαίδευση (e-learning). Πιο συγκεκριμένα θα γίνει ιστορική αναφορά, περιγραφή διαφόρων χαρακτηριστικών, αναφορά στις υπηρεσίες που χρησιμοποιούνται στην εφαρμογή καθώς και ανάλυση πρόσφατων στατιστικών στοιχείων.

1.1 Περιγραφή του αντικειμένου της πτυχιακής

Στις αρχές του 21^{ου} αιώνα έγινε η είσοδος των υπολογιστών και του Internet στην καθημερινότητα των ανθρώπων. Η λειτουργία της εκπαίδευσης, που μέχρι τότε ακολουθούσε την τυπική διαδικασία της παράδοσης από έναν διδάσκοντα σε ένα κοινό, άλλαξε καθώς η χρήση των ηλεκτρονικών συσκευών όπως υπολογιστές, smartphones και tablets έγινε εφικτή στη διδασκαλία. Η ηλεκτρονική εκπαίδευση (e-learning) αποτελεί μια μορφή εκπαίδευσης η οποία κάνει χρήση τέτοιων πολυμέσων με σκοπό να προσφέρει τη δυνατότητα στο χρήστη να μορφωθεί από οποιοδήποτε σημείο βρίσκεται. Στις μέρες μας το e-learning παρέχεται κυρίως μέσω διαδικτύου ή με τη χρήση άλλων μέσων όπως cd, DVD κ.α. [1].

Το e-learning έχει ως σκοπό να κάνει το μαθητή να αισθάνεται ότι βρίσκεται σε μια παραδοσιακή τάξη, δίνοντάς του τη δυνατότητα να μοιράζεται ηλεκτρονικό υλικό (βίντεο με διαλέξεις, παρουσιάσεις PowerPoint, αρχεία word, pdf) με άλλους μαθητές και να παρέχει τρόπους επικοινωνίας με τον υπεύθυνο καθηγητή μέσω διαδικτύου όπως email, προσωπικά μηνύματα, χρήση κοινοτήτων (forum) καθώς επίσης και προγράμματα απευθείας ανταλλαγής μηνυμάτων όπως το Skype. Το κύριο θετικό του e-learning είναι ότι ο κάθε ενδιαφερόμενος μπορεί ανεξαρτήτως ώρας μέσα στην ημέρα, να παρακολουθήσει το εκπαιδευτικό υλικό που επιθυμεί καθώς επίσης και να συνομιλήσει με άλλους ενδιαφερόμενους.

Το e-learning, αποτελεί μια γρηγορότερη αλλά και φθηνότερη εναλλακτική λύση συγκριτικά με τον παραδοσιακό τρόπο εκπαίδευσης όπου τις περισσότερες φορές

είναι ακριβός, χρονοβόρος σε ορισμένες περιπτώσεις και τα αποτελέσματα του δεν είναι τόσο επιτυχημένα.

1.2 Ιστορική αναδρομή

Ο όρος "e-learning" υπάρχει από τα τέλη του προηγούμενου αιώνα και πιο συγκεκριμένα, το 1999 όπου χρησιμοποιήθηκε για πρώτη φορά σε σεμινάριο σχετικά με προγράμματα εκπαίδευσης σε υπολογιστή (computer based training program, CBT). Για να υπάρχει μια πιο ακριβής περιγραφή της διαδικασίας υπήρξαν και άλλοι όροι όπως "online training" και "εικονική εκπαίδευση" (virtual training), ωστόσο υπάρχουν αποδείξεις ύπαρξης μιας πρώιμης μορφής e-learning από τις αρχές του 19^{ου} αιώνα [2].

1.2.1 Στην αρχαιότητα

Στην παρακάτω εικόνα φαίνεται μια μαρμάρινη πλάκα που χρονολογείται περίπου στο 200 π.Χ. στην αρχαία Ρώμη και παρουσιάζει μία σχολική τάξη. Ο δάσκαλος βρίσκεται στη μέση με 2 μαθητές γύρω του, όπου ο καθένας διαβάζει από πάπυρους ενώ ένας τρίτος μαθητής έχει στα χέρια του μια πλάκα στην οποία μπορούσε να γράψει. Αυτή η τεχνική γραφής χρησιμοποιήθηκε αρκετά στα σχολεία της Ευρώπης μέχρι το 1950. Ο Dr. Marcel Mirande αναφέρει στο βιβλίο του «De onstuitbare orkomst van de leermachine», ότι η πλάκα που έγραφαν ήταν το laptop της τότε εποχής και σκοπός της ήταν η ανάπτυξη των τεχνικών γραφής των μαθητών [3].



Εικόνα 1.1: Τάξη στην αρχαία Ρώμη

1.2.2 Η πρώτη μορφή μάθησης από απόσταση

Ακόμα και πριν τη δημιουργία του διαδικτύου πραγματοποιούνταν μαθήματα από απόσταση για να παρέχεται στους μαθητές η εκπαίδευση που χρειαζόντουσαν σε πιο εξειδικευμένα πεδία ή ειδικότητες. Στη δεκαετία του 1840 ο Isaac Pitman δίδασκε τους μαθητές του ένα είδος στενογραφίας μέσω αλληλογραφίας (shorthand), που χρησιμοποιούταν για να βελτιώσει το χρόνο γραψίματος δίνοντας, έτσι μεγάλη βοήθεια σε επαγγέλματα όπως γραμματείς, δημοσιογράφοι και γενικότερα, επαγγέλματα που η συνεχής γραφή κειμένου ήταν απαραίτητη. Ο Pitman που ήταν εξειδικευμένος καθηγητής λάμβανε μέσω αλληλογραφίας ολοκληρωμένες ασκήσεις από τους μαθητές του και έπειτα εκείνος τους έστελνε με το ίδιο σύστημα επιπλέον υλικό για να ολοκληρώσουν [3].

1.2.3 Η μηχανή του Pressey

Στις αρχές της δεκαετίας του 1920 ο Sidney Pressey, καθηγητής ψυχολογίας στο πανεπιστήμιο του Ohio, δημιούργησε μια μηχανή η οποία παρείχε στους μαθητές των εισαγωγικών τάξεων του δοκιμασίες και αντικείμενα εξάσκησης. Το 1926 ο Pressey δήλωσε ότι η διαδικασία για την κατανόηση των δοκιμασιών και του υλικού πληροφόρησης πολλές φορές ήταν αρκετά απλή και κατανοητή έτσι ώστε να επιτρέπεται η διαχείριση τέτοιων ρουτίνων διδασκαλίας από μηχανικά μέσα. Η μηχανή του Pressey όπως απεικονίζεται παρακάτω, έμοιαζε με έναν τυπογράφο όπου είχε ένα παράθυρο που έδειχνε μια ερώτηση συνοδευόμενη από 4 απαντήσεις, ο χρήστης πάταγε το κλειδί που αντιστοιχούσε στην απάντηση που επέλεξε και το μηχάνημα κατέγραφε την απάντηση του σε έναν μετρητή που βρίσκονταν στο πίσω μέρος του και εμφάνιζε την επόμενη ερώτηση. Μόλις ο χρήστης είχε απαντήσει σε όλες τις ερωτήσεις ο υπεύθυνος του τεστ έβαζε το χαρτί πάλι στο μηχάνημα και έβλεπε το σκορ του χρήστη στο μετρητή.

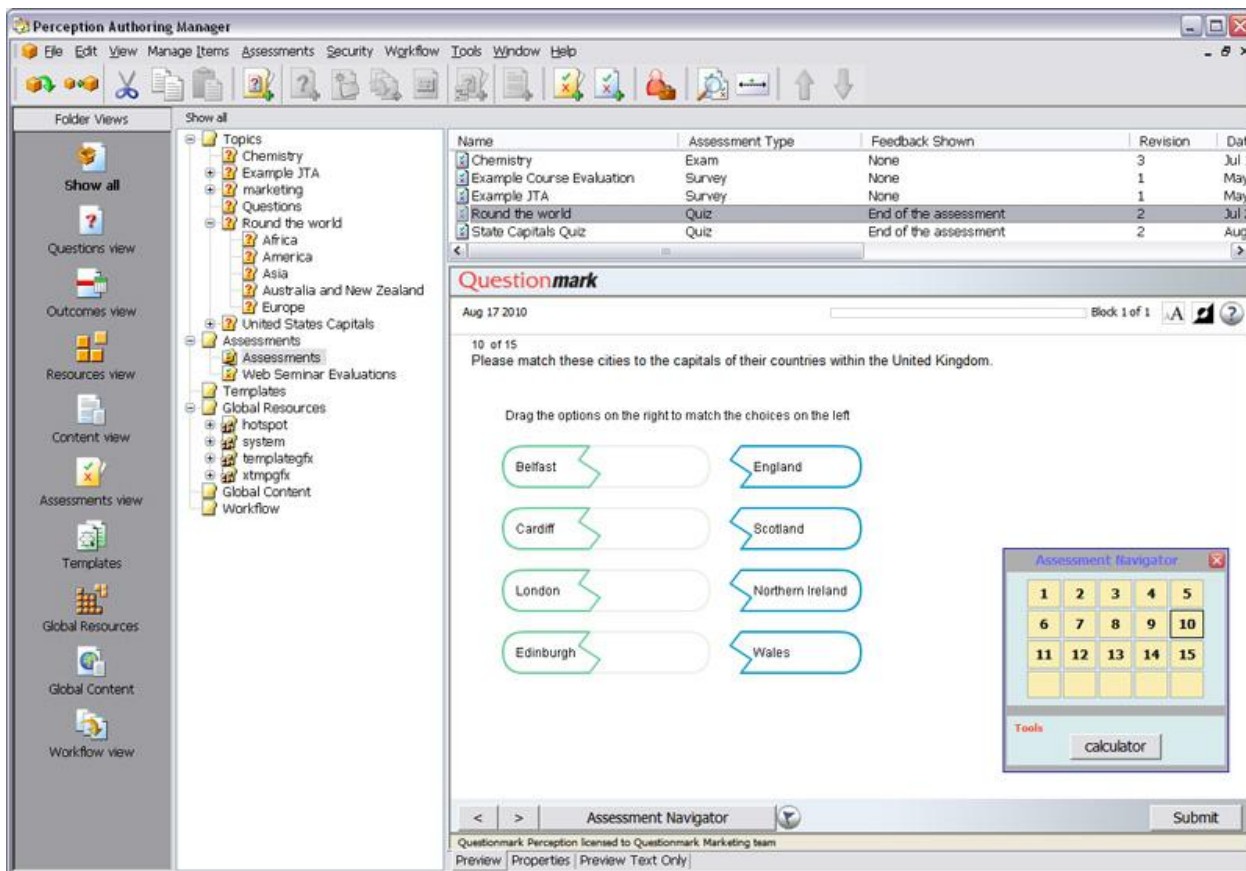


Εικόνα 1.2: Η μηχανή του Pressey

Παρά τα ενδιαφέροντα αποτελέσματα στην ερευνά του, η μηχανή του Pressey δεν είχε επιτυχία στον εκπαιδευτικό κόσμο και δεν χρησιμοποιήθηκε από κανένα φορέα εκπαίδευσης αλλά ούτε και από εταιρίες. Κύριος λόγος ήταν η χρονική περίοδος αφού τότε ξέσπασε η παγκόσμια οικονομική ύφεση του 1929, γνωστή και ως Great Depression. Τον ίδιο χρόνο με την επανάσταση του Pressley το ποσοστό ανεργίας έφτανε το 23.6% και οι νέες καινοτομίες στην τεχνολογία που αφορούσαν την εκπαίδευση καθυστέρησαν λόγω του 2^{ου} Παγκοσμίου πολέμου που ξέσπασε αργότερα την ίδια δεκαετία. Αρκετοί διδάσκοντες ήταν σκεπτικοί σχετικά με την μηχανή του γιατί φοβόντουσαν για τις δουλειές τους ενώ στον ίδιο ασκήθηκε πολύ άσχημη κριτική από συναδέλφους του. Ο B.F. Skinner απέδωσε την αποτυχία της μηχανής στο γεγονός ότι ο κόσμος της εκπαίδευσης δεν ήταν ακόμα έτοιμος ενώ δήλωσε ότι η μηχανή είχε σχεδιαστεί με σκοπό τη χρήση της αφού οι χρήστες είχαν λάβει προηγουμένως βασική εκπαίδευση. Σήμερα αυτός ο τρόπος εξέτασης χρησιμοποιείται σε διαδικτυακά

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

συστήματα όπως το Questionmark Perception το οποίο επιτρέπει στους εκπαιδευτές και τους διδάσκοντες να συγγράψουν, προγραμματίσουν, να παραδίδουν και να ελέγχουν ερωτηματολόγια, κουίζ, τεστ και διαγωνίσματα. Αυτό δείχνει ότι οι εξετάσεις και οι αξιολογήσεις μπορούν να γίνουν τέλεια με έναν αυτοματοποιημένο τρόπο διευκολύνοντας έτσι αυτές τις διαδικασίες και ταυτόχρονα να βελτιώνει την ποιότητα των ερωτήσεων και των τεστ [3].



Εικόνα 1.3: Παράδειγμα τεστ μέσω Questionmark Perception

1.2.4 Εκπαίδευση κατά τη διάρκεια του 2^{ου} παγκοσμίου πολέμου

Παρόλο που η βιομηχανία του κινηματογράφου είχε γνωρίσει άνθηση εκείνη την εποχή, οι ταινίες δεν χρησιμοποιούντουσαν ως μέσα εκπαίδευσης. Με τον ερχομό του 2^{ου} παγκοσμίου πολέμου αυτό άλλαξε καθότι παρουσιάστηκε πρόβλημα εκπαίδευσης αφού χιλιάδες νέοι στρατιώτες έπρεπε να περάσουν από βασική εκπαίδευση σε γρήγορο χρονικό διάστημα και εξαιτίας της πολυπλοκότητας των νέων όπλων απαιτούνταν ένα πάρα πολύ υψηλό επίπεδο κατάρτισης. Έτσι λόγω του πολέμου οι ταινίες υπέστησαν μεγάλη ανάπτυξη και έγιναν γρήγορα μια κυρίαρχη τεχνολογία που έβρισκε χρήση παντού [3]. Ένα ενδιαφέρον στατιστικό στοιχείο είναι ότι μεταξύ του 1941 και του 1945 το τμήμα οπτικών βοηθειών για την εκπαίδευση πολέμου στο γραφείο εκπαίδευσης των Ηνωμένων Πολιτειών Αμερικής παράγαγε:

- 457 ταινίες με ήχο
- 432 ταινίες χωρίς ήχο
- 457 ταινίες με οδηγίες για τους εκπαιδευτές

1.2.5 Μετά τον πόλεμο

Το 1954 ο BF Skinner, καθηγητής στο Harvard, εφηύρε τη μηχανή διδασκαλίας (teaching machine) που επέτρεπε στα εκπαιδευτικά ιδρύματα να παρέχουν προγραμματισμένες οδηγίες στους μαθητές τους.



Εικόνα 1.4: Η teaching machine του BF Skinner

Το πρώτο πρόγραμμα εκπαίδευσης βασισμένο σε υπολογιστή παρουσιάστηκε το 1960 και ήταν γνωστό σαν PLATO (Programmed Logic for Automated Teaching Operations). Αρχικά είχε σχεδιαστεί για χρήση από τους μαθητές του πανεπιστημίου του Illinois αλλά τελικά κατέληξε να χρησιμοποιείται σε εκπαιδευτικά ιδρύματα σε όλη την περιοχή. Τα πρώτα συστήματα e-learning είχαν προγραμματιστεί έτσι ώστε να παρέχουν πληροφορίες στους μαθητές αλλά στη δεκαετία του '70 το e-learning άρχισε να γίνεται διαδραστικότερο.

Το ανοιχτό πανεπιστήμιο της Βρετανίας, όπου η ήδη κυρίαρχη μέθοδος εκπαίδευσης εκείνη την περίοδο ήταν αυτή από απόσταση, εκμεταλλεύτηκε το e-learning σε μεγάλο βαθμό. Στο παρελθόν η διδασκαλία στο ανοιχτό πανεπιστήμιο και η ανταπόκριση με τους καθηγητές γίνονταν μέσω ταχυδρομείου. Με την εξάπλωση του διαδικτύου το ανοιχτό πανεπιστήμιο άρχισε να προσφέρει στους μαθητές του μια

μεγάλη γκάμα από διαδραστική εκπαίδευση καθώς και πιο γρήγορη επικοινωνία μαθητών και εκπαιδευτικών μέσω e-mail [1].

1.2.6 Τα τέλη του 20^{ου} αιώνα και το παρόν

Στα τέλη του 20^{ου} αιώνα με την εισαγωγή του υπολογιστή και του διαδικτύου τα εργαλεία e-learning και οι μέθοδοι παροχής του επεκτάθηκαν. Τη δεκαετία του '80 χάρις την κυκλοφορία του πρώτου MAC πολλοί άνθρωποι απέκτησαν έναν υπολογιστή στο σπίτι τους με αποτέλεσμα να αρχίσουν να εξοικειώνονται με στη χρήση του. Στη δεκαετία του '90 υπήρξε άνθηση των εικονικών περιβαλλόντων διδασκαλίας με δυνατότητα για πολλούς ανθρώπους την πρόσβαση σε μία τεράστια πληθώρα πληροφοριών και e-learning ευκαιριών. Στην αρχή της δεκαετίας πολλά πανεπιστήμια παρείχαν αποκλειστικά διαδικτυακά μαθήματα προσφέροντας έτσι εκπαίδευση σε ανθρώπους που δεν μπορούσαν να παρακολουθήσουν κανονικά μαθήματα εξαιτίας γεωγραφικής θέσης ή χρόνου. Ένα ακόμα θετικό στοιχείο είναι πως χάρις την τεχνολογική ανάπτυξη δόθηκε η δυνατότητα μείωσης του κόστους της διδασκαλίας από απόσταση τόσο στα εκπαιδευτικά ιδρύματα όσο και στους μαθητές προσφέροντας έτσι εκπαίδευση σε περισσότερους ανθρώπους. Στην πρώτη δεκαετία του 21^{ου} αιώνα άρχισε να χρησιμοποιείται το e-learning από πολλές επιχειρήσεις για εκπαίδευση των υπαλλήλων τους, δίνοντας τους έτσι τη δυνατότητα να βελτιώσουν τις ικανότητες και τις γνώσεις του πάνω στην αντίστοιχη βιομηχανία που εργάζονταν. Επίσης πλέον δίνεται η δυνατότητα σε πολλούς να αποκτήσουν πτυχίο και διαδικτυακά ή και να πιστοποιήσουν τις γνώσεις τους μέσω certifications [1].

1.3 Το μέλλον του e-learning

Στις μέρες μας όπου στα περισσότερα σπίτια υπάρχει ένας υπολογιστής και σύνδεση στο διαδίκτυο, το e-learning είναι πλέον πιο προσβάσιμο από ποτέ ενώ παράλληλα με την επέκταση των δικτύων για τα κινητά τηλέφωνα ο ενδιαφερόμενος μπορεί να μορφωθεί από την καφετέρια, τα διαλείμματα της δουλειάς του ακόμα και όταν είναι στο λεωφορείο. Με την ενσωμάτωση των social media υπάρχει τρόπος για περισσότερη και πιο εύκολη επικοινωνία μεταξύ των ενδιαφερόμενων. Μερικές από τις

τάσεις που υπάρχουν στο e-learning δείχνει πως θα διαμορφωθεί αυτό αλλά και γενικότερα όλα εργαλεία μάθησης στο μέλλον [1]. Αυτές οι τάσεις είναι:

Μικρό-μάθηση (Micro-learning): Η μικρό-μάθηση επικεντρώνεται στην δημιουργία μικροδραστηριοτήτων για την εκπαίδευση. Ένα παράδειγμα μικρό-μάθησης είναι όταν κάποιος θέλει να μάθει μια ξένη γλώσσα, στην αρχή μαθαίνει κάποιες βασικές φράσεις ή λέξεις και μετά πάει σταδιακά σε πιο δύσκολα κομμάτια όπως η γραμματική.

Παιχνιδοποίηση (gamification): Στην παιχνιδοποίηση γίνεται χρήση μηχανισμών παιχνιδιών σε περιβάλλοντα είναι περιπτώ, έχοντας ως αποτέλεσμα να αυξήσει σε μεγαλύτερο βαθμό ενδιαφέρον του χρήστη (π.χ. η ύπαρξη βαθμολογίας σε ένα μάθημα με ερωτήσεις- απαντήσεις).

Προσωπική μάθηση (Personalized Learning): Στην προσωπική μάθηση γίνεται κατασκευή των περιβαλλόντων μάθησης και όσων στοιχείων τα αποτελούν (π.χ. τις δραστηριότητες του μαθήματος) με στόχο την ικανοποίηση των αναγκών και φιλοδοξιών κάθε χρήστη ξεχωριστά. Αυτό είναι πολύ σημαντικό για το χρήστη, γιατί του δίνεται η δυνατότητα να επιλέξει εκείνος τον τρόπο που θέλει να μορφωθεί χωρίς να επηρεάζονται οι υποχρεώσεις που έχει στο μάθημα π.χ. ασκήσεις που πρέπει να παραδοθούν πριν από μια καταληκτική ημερομηνία.

1.4 E-learning: πλεονεκτήματα & μειονεκτήματα

Σημαντικά πλεονεκτήματα του e-learning είναι :

- **Μάθηση χωρίς περιορισμούς**: Στην παραδοσιακή μάθηση υπάρχουν οι τοπικοί και οι χρονικοί περιορισμοί. Πολλοί άνθρωποι δεν μπορούν να παραστούν σε μια τάξη για διάφορους λόγους (λόγω της γεωγραφικής τους θέσης, περιορισμένου χρόνου), αλλά με τη βοήθεια του e-learning το φαινόμενο αυτό εξαλείφεται αφού όλοι μπορούν να μάθουν από όπου θέλουν και όποτε θέλουν.

- **Διασκεδαστικό** : Με έναν καλό σχεδιασμό (π.χ. διαδραστικός σχεδιασμός, χρήση πολυμέσων, μέσω της μεθόδου παιχνιδιοποίησης) ένα ηλεκτρονικό μάθημα (course) μπορεί να αυξήσει το ενδιαφέρον του χρήστη σε μεγάλο βαθμό καθώς και να το βοηθήσει να αντέξει περισσότερο στο χρόνο.
- **Χαμηλό κόστος** : Το κόστος και για τις 2 πλευρές (μαθητές και καθηγητές) είναι σημαντικά πιο χαμηλό στην εκπαίδευση μέσω e-learning με κυριότερο παράδειγμα τη μη ύπαρξη εξόδων για τα βιβλία τα οποία πολλές φορές είναι πολύ δαπανηρά και μετά από κάποια χρόνια που το περιεχόμενό τους θα έχει αλλάξει οι παλιές εκδόσεις που έχουν αγοραστεί γίνονται περιττές. Άλλα έξοδα που εξαλείφονται είναι τα έξοδα μεταφοράς κ.α.

Μερικά μειονεκτήματα του e-learning :

- **Δυσκολία μετάδοσης γνώσης** : Σε κάποιους τομείς κρίνεται απαραίτητη η φυσική παρουσία του εκπαιδευόμενου γιατί η αναπαράσταση του αντικειμένου που διδάσκεται δεν είναι επαρκής για την πλήρη κατανόησή του. Για παράδειγμα είναι δύσκολο να διδάξεις μηχανική για αυτοκινήτα χωρίς την εξάσκηση στην πράξη.
- **Απομόνωση** : Κατά κύριο λόγο αυτή τη στιγμή το e-learning αποτελεί ατομική δραστηριότητα. Έτσι δημιουργείται μια αίσθηση απομόνωσης για τους μαθητές σε σχέση με την παραδοσιακή εκπαίδευση. Το φαινόμενο αυτό έχει αρχίσει να μειώνεται λόγω της άμεσης επικοινωνίας μέσω e-mail και chat με τους καθηγητές και τους συμμαθητές.
- **Θέματα υγείας** : Το e-learning απαιτεί τη χρήση ηλεκτρονικών συσκευών (υπολογιστές, laptops, smartphones) με αποτέλεσμα την πιθανότητα δημιουργίας προβλήματος στα μάτια, την πιθανότητα ο μαθητής να μην έχει σωστή στάση όταν κάθεται στην καρέκλα καθώς και άλλα σωματικά προβλήματα. Γι'αυτό το λόγο πλέον στα περισσότερα ηλεκτρονικά μαθήματα υπάρχουν ενδείξεις για τακτικά διαλείμματα.

1.4 Συγχρονισμένο και ασύγχρονο e-learning

Η εκπαίδευση που παρέχεται μέσω των e-learning συστημάτων χωρίζεται σε 2 κατηγορίες, την συγχρονισμένη και την ασύγχρονη εκπαίδευση. Για το ποιος τρόπος είναι καλύτερος αυτό εξαρτάται κυρίως από τις ανάγκες του κάθε ενδιαφερόμενου που χρησιμοποιεί την υπηρεσία. Ιδανικά ένα καλό ηλεκτρονικό μάθημα θα πρέπει να παρέχει ένα συνδυασμό των δύο τρόπων έτσι ώστε να εξυπηρετούνται όλοι, μαθητές και καθηγητές [1]. Παρακάτω υπάρχει ανάλυση των 2 κατηγοριών.

- **Συγχρονισμένη εκπαίδευση**

Συγχρονισμένη εκπαίδευση είναι η εκπαίδευση όπου το μάθημα γίνεται σε πραγματικό χρόνο (πχ. Live - βίντεο, διαδικτυακές συνομιλίες, άμεσα μηνύματα) με αποτέλεσμα να υπάρχει άμεση επικοινωνία μεταξύ μαθητών και καθηγητή. Το κύριο πλεονέκτημα της συγχρονισμένης εκπαίδευσης είναι ότι οι μαθητές δεν έχουν το αίσθημα της απομόνωσης αφού αποτελούν μέρος μιας τάξης και μπορούν να συνομιλούν μεταξύ τους. Το μειονέκτημα είναι ότι αφού το μάθημα γίνεται σε συγκεκριμένη ώρα πολλοί μαθητές δεν θα μπορούν να το παρακολουθήσουν λόγω του προγράμματός τους.

- **Ασύγχρονη εκπαίδευση**

Στην ασύγχρονη εκπαίδευση γίνεται το ακριβώς αντίθετο, το μάθημα μπορεί να γίνει ακόμα και όταν ο καθηγητής ή ο μαθητής δεν είναι συνδεδεμένος. Ένα παράδειγμα ασύγχρονης εκπαίδευσης είναι το ανέβασμα ασκήσεων στο διαδίκτυο έτσι ώστε ο μαθητής να μπορεί να έχει πρόσβαση σε αυτά όποτε του επιτρέπει το πρόγραμμα του χωρίς την πίεση του χρόνου. Με αυτόν τον τρόπο όμως είναι δυνατόν αρκετοί μαθητές που δεν έχουν κάποιο κίνητρο να μην επωφεληθούν, ενώ μπορεί να υπάρξει και το αίσθημα της απομόνωσης αφού ο μαθητής δε θα αποτελεί μέρος μιας “τάξης”.

1.5 Συστήματα διαχείρισης μάθησης (Learning management systems - LMS)

Ο όρος σύστημα διαχείρισης μάθησης (Learning management systems - LMS) αναφέρεται σε ένα υπολογιστικό σύστημα που έχει αναπτυχθεί με σκοπό τη διαχείριση διαδικτυακών μαθημάτων, τη διανομή υλικού σχετικό με αυτά και την επικοινωνία και συνεργασία μεταξύ μαθητών και καθηγητών. Ένα LMS θα επιτρέπει στον υπεύθυνο να ελέγξει κάθε στοιχείο ενός μαθήματος από την εγγραφή των μαθητών μέχρι την αποθήκευση αποτελεσμάτων διαγωνισμάτων καθώς και την αποδοχή των ασκήσεων ψηφιακά και την επικοινωνία μαθητών - καθηγητών. Στην ουσία τα LMS είναι το βασικό συστατικό για τις περισσότερες δραστηριότητες e-learning. Κατασκευάζεται με διάφορες πλατφόρμες με πιο συνηθισμένες τις PHP, .Net ή Java όπου θα συνδέονται με μια βάση δεδομένων PostgreSQL, MySQL ή SQL Server. Σε ένα επιχειρησιακό περιβάλλον ένα τέτοιο σύστημα μπορεί να χρησιμοποιηθεί για την παρακολούθηση των επιδόσεων του προσωπικού και την τήρηση αρχείων των αποδόσεων και της εκπαίδευσης του [1].

1.6 Massive open online courses (MOOC)

Ένα Massive Open Online Courses (MOOC) είναι ένα διαδικτυακό μάθημα με ελεύθερη εγγραφή που παρέχει ίδιο πρόγραμμα μαθημάτων για όλους έχοντας ανοικτά τελικά αποτελέσματα. Τα MOOCs έχουν κοινωνικά δίκτυα, προσβάσιμο online υλικό και εποπτεύονται από κορυφαίους επαγγελματίες στο πεδίο της εκπαίδευσης. Το πιο σημαντικό είναι ότι είναι κατασκευασμένα με βάση το ενδιαφέρον των εκπαιδευόμενων που τα χρησιμοποιούν παρέχοντας τους τη δυνατότητα να οργανώνουν τη συμμετοχή τους σύμφωνα με τους μαθησιακούς τους στόχους, τις γνώσεις που ήδη διαθέτουν καθώς και ικανότητες αλλά και άλλα κοινά ενδιαφέροντα [4].

1.7 Οι κύριες τάσεις του e-learning

Παρακάτω εξετάζονται οι κύριες τάσεις που υπάρχουν αυτή τη στιγμή στο e-learning και δείχνουν σε κάποιο βαθμό την εξέλιξη του τις επόμενες δεκαετίες [1].

1.7.1 Ανάμεικτη μάθηση (Blended learning)

Η ανάμεικτη μάθηση (Blended learning) αποτελεί συνδυασμό της παραδοσιακής και της ηλεκτρονικής μάθησης με τρόπο που αλληλοσυμπληρώνονται. Για παράδειγμα θα δίνεται η δυνατότητα σε ένα μαθητή να παρακολουθεί ένα μάθημα αυτοπροσώπως και ύστερα να διαβάσει υλικό είτε να λύσει ασκήσεις που έχουν ανέβει στο διαδίκτυο. Μερικά από τα εργαλεία που βοηθούν στην ανάμεικτη μάθηση είναι τα LMS, τα tablets και τα smartphones. Για την επιτυχία της ανάμεικτης μάθησης συμβάλουν 2 κύριες αρχές:

- Οι μαθητές που συνεργάζονται μεταξύ τους έχουν μια πιο εμπλουτισμένη εμπειρία μάθησης και η συνεργασία τους μπορεί να βελτιωθεί αν οι δραστηριότητες τους βασίζονται σε διαφορετικό υλικό που έχουν μάζεψει ο καθένας ξεχωριστά από το διαδίκτυο.
- Έχει παρατηρηθεί ότι οι μαθητές που ακολουθούν την ολοκλήρωση ενός ηλεκτρονικού μαθήματος μαζί με την παρακολούθηση ενός παραδοσιακού μαθήματος σε μια τάξη έχουν στατιστικά καλύτερες εμπειρίες από την εκπαίδευση τους.

1.7.2 Κοινωνική και συνεργατική εκπαίδευση (social and collaborative learning)

Η συνεργατική εκπαίδευση είναι μια μέθοδος e-learning όπου οι μαθητές μπορούν να επικοινωνούν μεταξύ τους, αλλά και με τους καθηγητές τους, με το αποτέλεσμα αυτής της συνεργασίας να είναι η επέκταση των γνώσεων τους στο συγκεκριμένο μάθημα. Στα περιβάλλοντα e-learning αυτό γίνεται μέσω διαδικτυακού chat, forum ή με αποστολή άμεσων μηνυμάτων. Η βασική έννοια αυτού του τρόπου εκπαίδευσης είναι πως όλοι οι μαθητές μπορούν να επωφεληθούν μέσω της μεταξύ τους συνεργασίας μαθαίνοντας περισσότερα πράγματα ο ένας από τον άλλον επάνω σε διάφορα αντικείμενα. Ακόμα αναπτύσσεται η εργασιακή εμπειρία καθώς η ομαδικότητα και η συνεργασία στις περισσότερες εργασίες είναι αναγκαίες. Μερικά πλεονεκτήματα είναι:

- Με αυτή τη μέθοδο τα μαθήματα μπορούν να πραγματοποιηθούν είτε offline είτε online, σύγχρονισμένα ή ασύγχρονα.
- Επιτρέπει στους μαθητές να μάθουν πολλά πράγματα από τους υπόλοιπους συμμαθητές τους. Όταν υπάρχει μια ομαδική εργασία τα μέλη μιας ομάδας μπορούν να μάθουν πως είναι να συνεργάζεσαι με τα υπόλοιπα μέλη της αναπτύσσοντας έτσι το αίσθημα της ομαδικότητας και της συνεργασίας.
- Επιπλέον πολλοί μαθητές που δεν μπορούν να παρευρεθούν σε μια συνάντηση μπορούν να μιλούν με τους υπόλοιπους μέσω των forum, μηνυμάτων κ.α.

1.7.3 Παιχνιδοποίηση (Gamification)

Η παιχνιδοποίηση είναι η χρήση μηχανισμών, αισθητικών και σκέψεων βασισμένων σε παιχνίδια έτσι ώστε να προσελκύουν και να ενθαρρύνουν τους μαθητές, να προωθούν την εκπαίδευση και να λύνουν προβλήματα. Με απλά λόγια είναι η χρήση της τεχνολογίας των παιχνιδιών για λύσεις προβλημάτων ανεξάρτητων από τη βιομηχανία των παιχνιδιών.

1.8 Στατιστικά στοιχεία σχετικά με το e-learning

Παρακάτω ακολουθούν μερικά σημαντικά στοιχεία σχετικά με τη βιομηχανία του e-learning (και όχι μόνο) για το 2015 [5].

- **Η παγκόσμια αγορά του e-learning**

Η παγκόσμια αγορά του e-learning αναμένεται να φτάσει τα 107 δις δολάρια το 2015. Το 2010 τα έσοδα της αγοράς του είχαν φτάσει τα 32.1 δις δολάρια με ένα ετήσιο ποσοστό αύξησης του 9.2%, αυτό σημαίνει ότι το 2015 η αγορά του e-learning θα έχει έσοδα της τάξεως των 49.9 δις δολαρίων.

- **Οι χώρες με τα μεγαλύτερα ποσοστά αύξησης του e-learning**

Το ποσοστό αύξησης μας δείχνει πώς κάθε χώρα χρησιμοποιεί το e-learning, αποτελεί ένα σημαντικό μέγεθος καθώς μπορεί να αναδείξει νέες ευκαιρίες. Οι χώρες αυτές είναι:

1. Ινδία: 55%
2. Κίνα: 52%
3. Μαλαισία: 41%
4. Ρουμανία: 38%
5. Πολωνία: 28%
6. Τσεχία: 27%
7. Βραζιλία: 26%
8. Ινδονησία: 25%
9. Κολομβία: 20%
10. Ουκρανία: 20%

- **Η αγορά των συστημάτων διαχείρισης γνώσης (LMS)**

Η αγορά των LMS το 2013 είχε αξία 2.55 δις δολάρια με προβλεπόμενο ετήσιο ποσοστό αύξησης 25.2%, έτσι εκτιμάται ότι η αξία της το 2015 θα ανέβει στα 4 δις δολάρια ενώ το 2018 παραπάνω από 7 δις. Τη μεγαλύτερη συνεισφορά σε αυτό εκτιμάται ότι θα την έχει η βόρεια Αμερική.

- **Η αγορά της mobile γνώσης**

Η παγκόσμια αγορά για προϊόντα και υπηρεσίες Mobile Learning έφτασε τα 5.3 δις το 2012. Με ετήσιο ποσοστό αύξησης περίπου στο 18.2 για τα επόμενα πέντε χρόνια εκτιμάται ότι το 2015 θα φτάσει τα 8.7 δις δολάρια και τα 12.2 δις μέχρι το 2017. Άξιο αναφοράς είναι ότι το 2012 οι χώρες με τις περισσότερες αγορές σε αυτά ήταν οι ΗΠΑ, η Ιαπωνία, η Νότια Κορέα, η Κίνα και η Ινδία ενώ εκτιμάται πως μέχρι το 2017 οι κορυφαίοι αγοραστές θα είναι η Κίνα, οι ΗΠΑ, η Ινδονησία, η Ινδία και η Βραζιλία.

- **MOOCs στην εκπαίδευση εταιρικών υπάλληλων**

Αυτή τη στιγμή 8% των εταιριών χρησιμοποιούν MOOCs, ενώ 7% των εταιριών σκέπτονται να τα χρησιμοποιήσουν. Προβλέπεται ότι στα επόμενα 2 χρόνια το ποσοστό αυτό θα ανέβει στο 28%. Ενδεικτικό παράδειγμα είναι το ότι 350 εταιρίες συνεργάζονται με το Coursera και το Udacity για να αναγνωρίσουν τους καλύτερους μαθητές που θα μπορούσαν να είναι και οι καλύτεροι υποψήφιοι για σχετικές δουλειές. Ένα σημαντικό παράδειγμα είναι αυτό της google όπου έχει γράψει 80000 από τους υπαλλήλους της στο μάθημα για την HTML5 του Udacity.

- **Online εκπαίδευση εταιρικών υπαλλήλων**

Η αγορά για την online εκπαίδευση εταιρικών υπαλλήλων εκτιμάται ότι θα αναπτύσσεται με ποσοστό περίπου 13% κάθε χρόνο μέχρι το 2017. Σήμερα 77% των Αμερικανικών εταιριών προσφέρουν online εκπαίδευση στους υπαλλήλους τους για να βελτιώσουν την επαγγελματική τους ανάπτυξη.

- **Οι κορυφαίοι αγοραστές προϊόντων e-learning**

Οι μεγάλες εταιρίες αποτελούν τους κύριους αγοραστές προϊόντων και υπηρεσιών e-learning, μάλιστα αποτελούν το 30% των συνολικών αγοραστών.

- **Μέθοδοι διανομής της εταιρικής εκπαίδευσης**

Οι μέθοδοι διανομής για το 2014 έχουν ως εξής:

1. 1.47% των ωρών εκπαίδευσης έγιναν σε αίθουσα από εκπαιδευτή, το ποσοστό υπέστη αύξηση 3% συγκριτικά με τον προηγούμενο χρόνο.
2. 29.1% των ωρών εκπαίδευσης έγιναν με ανάμεικτες μεθόδους μάθησης, το ποσοστό υπέστη αύξηση 0.8% συγκριτικά με τον προηγούμενο χρόνο.
3. 28.5% των ωρών εκπαίδευσης έγιναν online ή με τεχνολογίες βασισμένες σε υπολογιστή (χωρίς εκπαιδευτή), το ποσοστό υπέστη αύξηση 2.6% συγκριτικά με τον προηγούμενο χρόνο.
4. 15% των ωρών εκπαίδευσης έγιναν μέσω εικονικής τάξης/ webcast με παρουσία εκπαιδευτή σε απομακρυσμένη τοποθεσία, το ποσοστό μειώθηκε κατά 1%

συγκριτικά με τον προηγούμενο χρόνο.

5. 4.2% των ωρών εκπαίδευσης έγιναν μέσω κοινωνικής εκπαίδευσης, το ποσοστό υπέστη αύξηση 0.9% συγκριτικά με τον προηγούμενο χρόνο.
6. 1.4% των ωρών εκπαίδευσης έγιναν μέσω mobile συσκευών, το ποσοστό μειώθηκε κατά 0.5% συγκριτικά με τον προηγούμενο χρόνο.

- **Μαθησιακές τεχνολογίες**

Οι μαθησιακές τεχνολογίες που χρησιμοποιήθηκαν το 2014 έχουν ως εξής:

1. 74% των εταιριών χρησιμοποιούν συστήματα διαχείρισης γνώσεων (LMS) και εικονικές τάξεις / webcasting.
2. 48% των εταιριών χρησιμοποιούν το Rapid e-learning Tool (εργαλείο μετατροπής ppt).
3. 33% των εταιριών χρησιμοποιούν εξομοιωτή εφαρμογών.
4. 25% των εταιριών χρησιμοποιούν σύστημα διαχείρισης περιεχομένου μάθησης.
5. 21% των εταιριών χρησιμοποιούν online υποστήριξη της επίδοσης ή σύστημα διαχείρισης γνώσεων.
6. 18% των εταιριών χρησιμοποιούν mobile εφαρμογές.
7. 11% των εταιριών χρησιμοποιούν Podcasting.

- **Ζήτηση για λογισμικό και υπηρεσίες e-learning το 2015**

Τα ακόλουθα στατιστικά παρουσιάζουν την αγοραστική πρόθεση των εταιριών το 2015 με βάση το Training Industry Report του 2014.

1. 44% των εταιριών σκοπεύουν να αγοράσουν διαδικτυακά εργαλεία και συστήματα μάθησης.
2. 41% των εταιριών σκοπεύουν να αγοράσουν συστήματα διαχείρισης μάθησης.
3. 37% των εταιριών σκοπεύουν να αγοράσουν εργαλεία / συστήματα εγγραφής.
4. 33% των εταιριών σκοπεύουν να αγοράσουν εργαλεία και συστήματα τάξεων.
5. 29% των εταιριών σκοπεύουν να αγοράσουν προϊόντα και υπηρεσίες ανάπτυξης περιεχομένου.

6. 27% των εταιριών σκοπεύουν να αγοράσουν courseware σχέδια και λογισμικό καθώς και εργαλεία παρουσίασης.
7. 18% των εταιριών σκοπεύουν να αγοράσουν ηχητικά προϊόντα και συστήματα καθώς επίσης και εργαλεία που βοηθούν στις διαδικτυακές συσκέψεις.

Αν αυτά τα στατιστικά σχετικά με το eLearning για το 2015 μας δείχνουν κάτι, είναι πως το μέλλον της βιομηχανίας του e-learning είναι στρωμένο με γεωμετρική ανάπτυξη και τεράστιες προοπτικές για κέρδος. Τώρα περισσότερο από ποτέ, οι εκπαιδευόμενοι και οι εταιρίες στρέφονται σε μαθήματα μέσω e-learning και online εκπαίδευσης για να πετύχουν όλους τους στόχους, προσωπικούς ή επαγγελματικούς.

1.9 Υπηρεσίες που θα χρησιμοποιηθούν στην πτυχιακή εργασία

Σε αυτή την ενότητα περιγράφονται οι υπηρεσίες και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

- **Heroku**

Το heroku είναι μια δωρεάν (για συγκεκριμένο αριθμό project) πλατφόρμα η οποία επιτρέπει στους προγραμματιστές να τρέχουν τις εφαρμογές τους στο cloud. Στην περίπτωση της rails το heroku δίνει μια σημαντική βοήθεια αφού παρέχει ένα domain στην εφαρμογή για προσομοίωση της σε πραγματικό περιβάλλον (μορφή domain project.herokuapp.com) [6].

- **Amazon Web Service**

Το Amazon Web Service είναι μια υπό πληρωμή υπηρεσία η οποία παρέχει στους χρήστες χώρο cloud. Στην συγκεκριμένη εργασία χρησιμοποιείται ως αποθηκευτικός χώρος των αρχείων των χρηστών στο cloud [7].

- **Github - bitbucket**

Τα github-bitbucket είναι site αποθήκευσης κώδικα βασισμένα στο git. Χρησιμοποιούνται από το μεγαλύτερο ποσοστό προγραμματιστών στον κόσμο γιατί διευκολύνουν στην κατασκευή του κώδικα (π.χ. επιστροφή σε παλαιότερες εκδόσεις, μεταφορά του κώδικα μεταξύ υπολογιστών, εντοπισμός λαθών από άλλους χρήστες κ.α.).

- **HTML**

Η html είναι η κύρια και η πιο χρησιμοποιούμενη γλώσσα προγραμματισμού για τη ανάπτυξη ιστοσελίδων παγκοσμίως. Ανακαλύφθηκε το 1990 από τον Tim Berners-Lee στο Cern στη Γενεύη για να συνοδεύσει το World Wide Web που είχε ανακαλυφθεί το 1989 από τον ίδιο τον Berners-Lee [8].

- **CSS**

Η CSS είναι γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαμόρφωση μιας ιστοσελίδας (π.χ. ποια γραμματοσειρά θα χρησιμοποιηθεί, τις αποστάσεις των στοιχείων της σελίδας, του μεγέθους μιας εικόνας κ.α.). Η πρώτη έκδοση της CSS δημιουργήθηκε το 1996 από την ομάδα W3C (World Wide Web Consortium) [9].

- **Bootstrap**

Το bootstrap είναι ένα framework τ' οποίο περιέχει διάφορα σχέδια σχετικά με την βελτίωση της όψης μίας σελίδας που είναι γραμμένα στις γλώσσες CSS-HTML. Αυτά τα σχέδια μπορεί να είναι ειδικά κουμπιά, text-box, γραμματοσειρές κ.α. Δημιουργήθηκε το 2010 από τους Mark Otto και Jacob Thornton όπου εκείνη την περίοδο δούλευαν για το twitter, γι'αυτό και στην αρχή ήταν γνωστό σαν twitter Blueprint. Πολλές γνωστές εταιρίες το χρησιμοποιούν για τις σελίδες τους όπως το NBA, τα Walmart, τα Target και το Bloomber business [10].

1.10 Τεχνολογίες που χρησιμοποιούνται στο e-learning

Το e-learning χρησιμοποιεί μια πληθώρα τεχνολογιών που έχουν φτιαχτεί αποκλειστικά για εκείνο είτε άλλες υπάρχουσες οι οποίες βοηθούν στο στόχο του όπως για παράδειγμα τα ηλεκτρονικά παιχνίδια μάθησης. Μια από τις κύριες τεχνολογίες του e-learning είναι τα εργαλεία επικοινωνίας, αυτά μπορεί να είναι απλά όπως ένα e-mail, ανταλλαγή προσωπικών μηνυμάτων ή σε πιο μεγάλη κλίμακα όπως η χρήση μιας αποκλειστικής κοινότητας (forum) και τα ευρέως διαδεδομένα social media.

Πολύ σημαντική επίσης είναι η χρήση των βάσεων δεδομένων για αποθήκευση του περιεχομένου ενός ηλεκτρονικού μαθήματος όπως οι παρουσιάσεις, αποτελέσματα τεστ κ.α. και με τη χρήση ενός συστήματος διαχείρισης περιεχομένου (Content Management System) μπορεί ο υπεύθυνος να κάνει διαγραφή, πρόσθεση και ανανέωση των δεδομένων του. Τέλος υπάρχουν εργαλεία για βελτίωση του υλικού αυτού όπως PowerPoint, εργαλεία για την κατασκευή διαδικτυακού τεστ για τα μαθήματα κ.α. [1].

ΚΕΦΑΛΑΙΟ 2

ΕΙΣΑΓΩΓΗ

Στο 2^ο κεφάλαιο γίνεται ανάλυση της Ruby on Rails περιγράφοντας τον τρόπο που λειτουργεί, τα εργαλεία που περιέχει καθώς και τις βασικές εντολές για τη δημιουργία μιας απλής εφαρμογής.

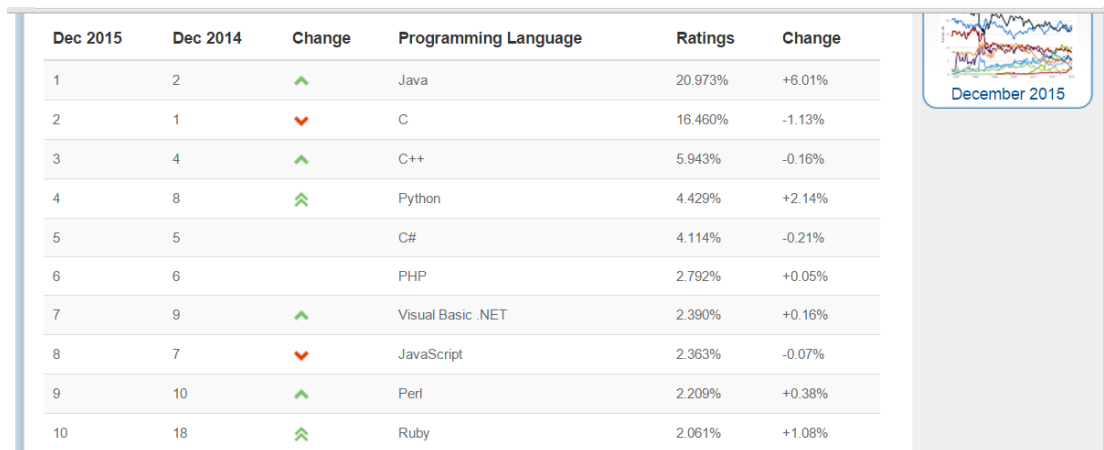
2.1 Η γλώσσα Ruby

Η ruby είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε το 1995 από τον Yukihiro “Matz” Matsumoto στην Ιαπωνία. Η ruby είναι μια γλώσσα “ισορροπίας”, αυτό σημαίνει ότι συνδυάζοντας τις αγαπημένες γλώσσες του (Perl, Smalltalk, Eiffel, Ada, List) ο Matsumoto κατάφερε να φτιάξει μια γλώσσα που ισορροπεί τον λειτουργικό προγραμματισμό με τον επιτακτικό προγραμματισμό [11]. Βασίζεται στον αντικειμενοστραφή προγραμματισμό αφού τα πάντα στη ruby λαμβάνονται ως αντικείμενα όπου το καθένα έχει τις δικές του ιδιότητες. Μερικά από τα σημαντικά πλεονεκτήματα της ruby είναι:

- Περιέχει χειριστές λαθών όπως η Java και η Python κάνοντας τον χειρισμό των λαθών ευκολότερο.
- Περιέχει threading ανεξάρτητο από το λειτουργικό σύστημα που σημαίνει ότι σε οποιοδήποτε σύστημα και να τρέξει θα υποστηρίζει multithreading.
- Υπάρχει υποστήριξη από τα πιο διαδεδομένα λειτουργικά συστήματα όπως Windows, Mac OS X, GNU/Linux. Η πιο συχνή χρήση της είναι στα συστήματα με Linux.
- Μπορεί να φορτώσει βιβλιοθήκες επεκτάσεων (extension libraries) δυναμικά αν το επιτρέπει το λειτουργικό σύστημα.

- Είναι πιο εύκολο να γραφτούν επεκτάσεις για τη C με Ruby αφού διαθέτει ένα πολύ καλό API για κλήση της από την C. Αυτό περιλαμβάνει την ενσωμάτωση της ruby σε λογισμικό ή χρήση της για δημιουργία script.
- Περιλαμβάνει mark-and-sweep garbage collector για όλα τα αντικείμενα της με αποτέλεσμα να μην χρειάζεται να κρατηθούν αναφορές σε βιβλιοθήκες επεκτάσεων και όπως λέει και ο δημιουργός της “αυτό είναι καλύτερο για την υγεία σας”.

Σύμφωνα με τα στοιχεία της TIOBE software [12], η ruby είναι μέσα στις 20 πιο χρησιμοποιούμενες γλώσσες στον πλανήτη (εικόνα 2.1). Εταιρίες που χρησιμοποιούν την ruby περιλαμβάνουν το κέντρο ερευνών Langley της NASA για προσομοιώσεις και ένα κέντρο ερευνών της MOTOROLA όπου χρησιμοποιεί την Ruby για να έναν προσομοιωτή που παράγει σενάρια και έπειτα επεξεργάζεται τα δεδομένα που απέκτησε. Κλείνοντας ένας κύριος παράγοντας της δημοτικότητας της ruby είναι το framework ruby on rails του οποίου η αναφορά θα γίνει στην επόμενη ενότητα.



Εικόνα 2.1: Στατιστικά στοιχεία της TIOBE software

2.2 Η Ruby On Rails

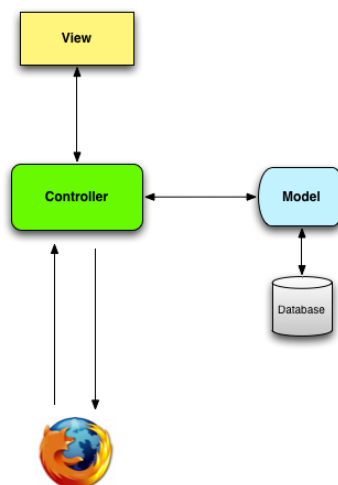
Η ruby on rails (RoR ή πιο απλά rails) αποτελεί ένα framework ανάπτυξης διαδικτυακών εφαρμογών γραμμένο σε Ruby. Δημιουργήθηκε το 2004 από τον David Heinemeier Hansson και την ομάδα του. Βασίζεται στη φιλοσοφία "μην επαναλαμβάνεσαι" δηλαδή ένα μέρος μιας πληροφορίας πρέπει να έχει μόνο 1 αναπαράσταση μέσα σε ένα σύστημα κάνοντας έτσι των κώδικα και την εφαρμογή γενικότερα πιο προσιτή και με λιγότερα σφάλματα [13].

Τα τελευταία χρόνια η rails χρησιμοποιείται όλο και πιο πολύ για την ανάπτυξη διαδικτυακών εφαρμογών σε μεγάλες εταιρίες όπως η Airbnb, η Basecamp, η Disney, το Github, το Hulu, το Kickstarter, το Shopify και φυσικά το Twitter. Μερικοί από τους λόγους της γρήγορης ανάπτυξης και της δημοτικότητας της rails είναι:

- Είναι γλώσσα ανοιχτού λογισμικού (open source) διαθέσιμη υπό την εποπτεία της MIT LICENSE με αποτέλεσμα η χρήση της να είναι δωρεάν.
- Ο κομψός σχεδιασμός της. Εκμεταλλεύοντας την ελαστικότητα της ruby προσφέρει μια γλώσσα συγκεκριμένου τομέα (domain specific language) για την κατασκευή διαδικτυακών εφαρμογών με αποτέλεσμα κάποια βασικά στοιχεία του διαδικτυακού προγραμματισμού (web programming) όπως η παραγωγή HTML κώδικα και η δημιουργία URL να γίνονται εύκολα με τη βοήθεια της.
- Η κοινότητα της. Ίσως ο πιο σημαντικός λόγος της δημοτικότητας της rails είναι η κοινότητα της, μέσω συνεδριάσεων, blog post, video, gemfiles η κοινότητα έχει παίξει τεράστιο ρόλο στην γρήγορη ανάπτυξη της.

2.3 Η αρχιτεκτονική MVC (Model-View –Controller)

Οι εφαρμογές που έχουν κατασκευαστεί με rails δουλεύουν με βάση την αρχιτεκτονική MVC (model-view-controller) που επιβάλλει τη διαφοροποίηση μεταξύ της “επιχειρηματικής λογικής” (“business logic”) και της λογικής εισαγωγής και παρουσίασης σε σχέση με ένα γραφικό περιβάλλον χρήστη (GUI). Στις διαδικτυακές εφαρμογές η “επιχειρηματική λογική ” αποτελείται συνήθως από μοντέλα δεδομένων (data models) για αντικείμενα όπως είναι οι χρήστες, προϊόντα κ.α. ενώ το GUI είναι συνήθως μια ιστοσελίδα σε έναν web browser. Ο τρόπος που δουλεύει μια rails εφαρμογή είναι ο ακόλουθος, στην αρχή ο χρήστης αλληλεπιδρά με μια rails εφαρμογή ενώ ο browser στέλνει αίτημα (request) το οποίο υποδέχεται ο web server και το περνάει στον αντίστοιχο χειριστή (controller) της rails. Στη συνέχεια ο controller είναι υπεύθυνος για ό,τι θα ακολουθήσει με πιο συνηθισμένη ενέργεια τη φόρτωση μιας σελίδας στο browser (ο controller έχει και άλλες δυνατότητες). Τις περισσότερες φορές ο controller αλληλεπιδρά με ένα μοντέλο (model) όπου είναι αντικείμενο της Ruby και αποτελεί στοιχείο της εφαρμογής (στις περισσότερες εφαρμογές ένα μοντέλο που υπάρχει σχεδόν πάντα είναι το μοντέλο του χρήστη) και είναι υπεύθυνο για την επικοινωνία με τη βάση δεδομένων. Τέλος μετά το κάλεσμα στο μοντέλο ο controller επιστρέφει την ολοκληρωμένη σελίδα στον browser υπό την μορφοποίηση της html [14].



Εικόνα 2.2: Αρχιτεκτονική MVC

2.4 Ανάλυση των M-V-C

Παρακάτω ακολουθεί η ανάλυση των 3 συστατικών της αρχιτεκτονικής MVC (model-view-controller) [15].

- **Μοντέλο (Model)**

Το μοντέλο αντιπροσωπεύει τα δεδομένα πληροφοριών της εφαρμογής και τους κανόνες για την επεξεργασία τους. Στη rails τα μοντέλα χρησιμοποιούνται για τη διαχείριση των κανόνων που αφορούν την αλληλεπίδραση με τον αντίστοιχο πίνακα στη βάση δεδομένων. Τις περισσότερες φορές ένας πίνακας στην βάση αντιπροσωπεύει ένα μοντέλο στην εφαρμογή.

- **Views**

Τα Views αντιπροσωπεύουν το περιβάλλον χρήστη της εφαρμογής (user interface). Στη rails είναι συνήθως HTML αρχεία με ενσωματωμένο κώδικα σε Ruby που εκτελεί εργασίες που αφορούν την παρουσίαση δεδομένων (π.χ. εμφάνιση δεδομένων ενός πίνακα). Τα views χρησιμοποιούνται για την παροχή δεδομένων στον web browser ή σε κάποιο άλλο εργαλείο που χρησιμοποιείται για να κάνει requests από την εφαρμογή.

- **Χειριστές (Controllers)**

Οι controllers παρέχουν τη σύνδεση μεταξύ των μοντέλων και των views. Στη rails οι χειριστές είναι υπεύθυνοι για την επεξεργασία εισερχόμενων αιτημάτων από τον web browser, έλεγχο των μοντέλων για δεδομένα και μεταφορά τους στα views για παρουσίαση.

2.5 Τα Ruby Gems

Τα gems στη ruby αποτελούν επεκτάσεις και μπορούν να παρέχουν πολλές επιπλέον λειτουργίες σε μια εφαρμογή όπως π.χ. χρήση άλλης βάσης δεδομένων, καινούρια τεστ κ.α. Στην εφαρμογή δηλώνονται στο αρχείο gemfile που βρίσκεται στον κυρίως φάκελο της. Άξιο αναφοράς είναι ότι τα περισσότερα gems έχουν φτιαχτεί από την ίδια την κοινότητα των Rails /Ruby.

2.6 Τα εργαλεία της Rails

Η rails παρέχει πολλά εργαλεία που αποτελούν τις πιο σημαντικές της κλάσεις για την δημιουργία διαδικτυακών εφαρμογών [15].

- **Action Controller (ActionController::Base)**

Το Action Controller διαχειρίζεται τους controllers σε μια εφαρμογή. Επεξεργάζεται εισερχόμενες αιτήσεις σε μια εφαρμογή ύστερα εξάγει παραμέτρους και τις στέλνει στην επιθυμητή ενέργεια (action). Μια action δηλώνεται ως μια δημόσια μέθοδο στον controller η οποία γίνεται διαθέσιμη αυτόματα στον web-server μέσω των routes της rails. Οι δυο κύριοι τύποι action που χρησιμοποιούνται στους action controllers είναι οι get-and-show και do-and-redirect, όπου οι περισσότερες actions αποτελούν παραλλαγές τους. “Υπηρεσίες” που παρέχει ο Action Controller είναι η session management, template rendering και redirect management.

- **Action View (ActionView::Base)**

Το Action View διαχειρίζεται τα views σε μια εφαρμογή rails. Παράγει εξόδους html και XML. Μερικές από τις δυνατότητες του είναι η φόρτωση template σελίδων (συμπεριλαμβανομένων nested και partial template), επίσης παρέχει ενσωματωμένη υποστήριξη AJAX. Οι τρόποι που γραφόνται τα template του Action View είναι με συνδυασμό ενσωματωμένου κώδικα Ruby και HTML με κατάληξη αρχείων .html.erb., με χρήση της βιβλιοθήκης Builder::XmlMarkup του Jim Weirich με κατάληξη .builder ή .rxml και με χρήση του ActionView::Helpers::PrototypeHelper::JavaScriptGenerator module με κατάληξη .rjs.

- **Active Record (ActiveRecord::Base)**

Το Active Record είναι το object - relational mapping (ORM) επίπεδο που παρέχεται μαζί με τη rails και ακολουθεί σχεδόν ευλαβικά το στάνταρ ORM μοντέλο με τους πίνακες να αντιστοιχούν σε κλάσεις, οι γραμμές σε αντικείμενα και οι στήλες σε ιδιότητες των αντικειμένων. Παρέχει ανεξαρτητοποίηση της βάσης δεδομένων, βασική CRUD (Create – Read – Update - Delete) λειτουργικότητα, ανεπτυγμένες ικανότητες αναζήτησης και μπορεί να συσχετίζει μοντέλα μεταξύ τους. Τα αντικείμενα του Active

Record δεν ορίζουν τα χαρακτηριστικά τους άμεσα αλλά τα συμπεραίνουν από τους ορισμούς των πινάκων με τους οποίους είναι συνδεδεμένα. Μερικές λειτουργίες όπως η πρόσθεση, η απομάκρυνση και η αλλαγή χαρακτηριστικών και των τύπων τους γίνεται απευθείας στη βάση με κάθε αλλαγή αντικατοπτρίζεται άμεσα στα αντικείμενα του active record.

- **Action Mailer (ActionMailer::Base)**

Το Action Mailer είναι ένα framework για την κατασκευή υπηρεσιών ηλεκτρονικού ταχυδρομείου και χρησιμοποιείται για την αποστολή email βασισμένων σε ευέλικτα πρότυπα ή για να λαμβάνει και να επεξεργάζεται εισερχόμενα email.

- **Active Resource(ActiveResource::Base)**

Το Active Resource παρέχει ένα framework που έχει σκοπό τη διαχείριση της σύνδεσης μεταξύ business αντικειμένων και των RESTful διαδικτυακών υπηρεσιών. Παρέχει τα μέσα για έναν τρόπο έτσι ώστε να συνδεθούν web-based resources με τοπικά αντικείμενα μέσω CRUD semantics.

- **Active Support(ActiveSupport::Base64)**

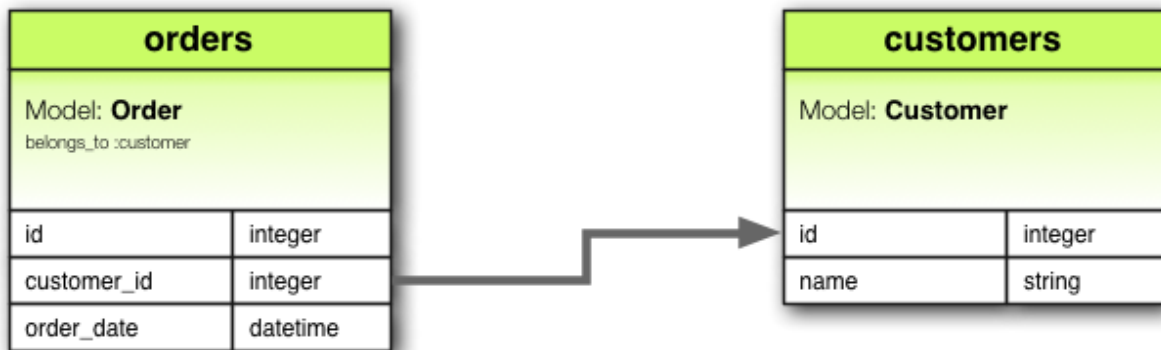
Το Active Support είναι μια εκτεταμένη συλλογή χρήσιμων κλάσεων και βασικών επεκτάσεων της βιβλιοθήκης της Ruby που χρησιμοποιούνται στη rails από τον κυρίως κώδικα και από τον κώδικα της εφαρμογής.

2.7 Associations μεταξύ των μοντέλων στη Rails

Στην rails association ονομάζεται η σύνδεση μεταξύ 2 μοντέλων της εφαρμογής με σκοπό την πρόσθεση χαρακτηριστικών στα μοντέλα, για παράδειγμα με το association belongs_to η rails κρατάει primary key – foreign key πληροφορίες ανάμεσα σε 2 καταγραφές (instances) των μοντέλων δίνοντας τους επιπλέον βοηθητικές μεθόδους. Οι τύποι των associations είναι belongs_to, has_many, has_many :through ενώ υπάρχουν και τα has_one, has_one :through, has_and_belongs_to_many αλλά δε χρησιμοποιούνται συχνά [13]. Παρακάτω θα γίνει ανάλυση στους 3 πιο σημαντικούς τύπους χρησιμοποιώντας παραδείγματα.

2.7.1 belongs_to association

Ένα belongs_to association παρέχει μια σχέση ένα προς ένα μεταξύ 2 μοντέλων έτσι ώστε μία καταγραφή ενός μοντέλου να “ανήκει” σε μια καταγραφή του άλλου. Για παράδειγμα αν έχουμε μια εφαρμογή για ένα κατάστημα μια παραγγελία ανήκει σε έναν πελάτη. Η σχέση μεταξύ των 2 κλειδιών και ο τρόπος δήλωσης στο μοντέλο της παραγγελίας είναι όπως φαίνεται παρακάτω.

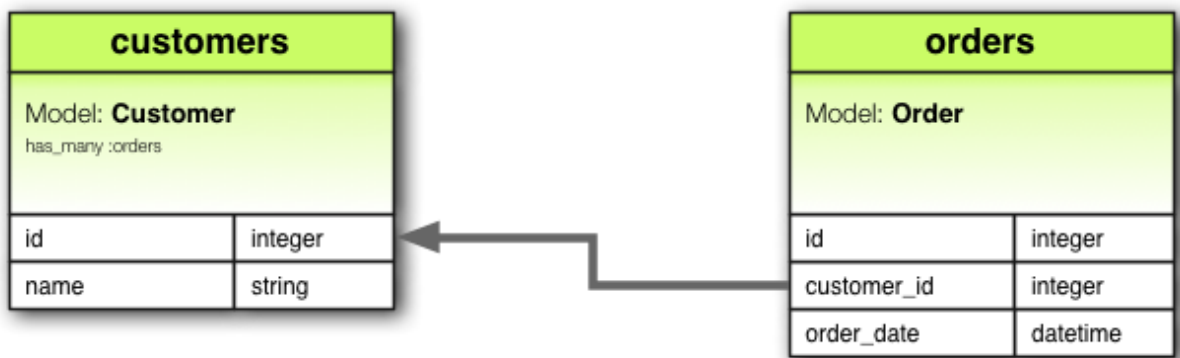


```
class Order < ActiveRecord::Base
  belongs_to :customer
end
```

Εικόνα 2.3α: Παράδειγμα belongs_to association

2.7.2 has_many association

Ένα has_many association παρέχει μια σχέση ένα προς πολλά μεταξύ 2 μοντέλων. Αποτελεί το αντίθετο του belongs_to και δείχνει ότι ένα μοντέλο μπορεί να περιέχει 0 ή περισσότερες καταγραφές ενός άλλου μοντέλου. Στο παραπάνω παράδειγμα το has_many περιγράφεται με την έννοια ότι ένας πελάτης θα μπορεί να έχει καμία η περισσότερες παραγγελίες. Ο τρόπος δήλωσης και η σύνδεση μεταξύ των 2 μοντέλων είναι η ακόλουθη.

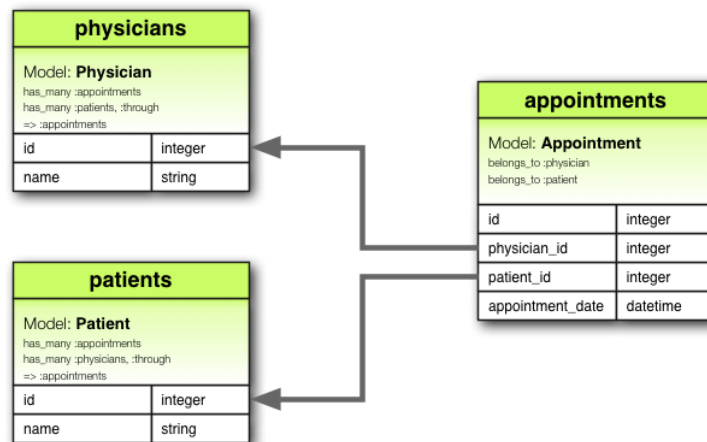


```
class Customer < ActiveRecord::Base
  has_many :orders
end
```

Εικόνα 2.3β: Παράδειγμα has_many association

2.7.3 has_many :through association

Ένα has_many :through association δηλώνει μια σύνδεση πολλά προς πολλά μεταξύ 2 μοντέλων και ότι ένα μοντέλο μπορεί να συνδεθεί με μηδέν ή περισσότερες καταγραφές ενός άλλου μοντέλου μέσω ενός τρίτου. Για παράδειγμα εάν υπήρχε μια εφαρμογή για ιατρείο και οι γιατροί έβλεπαν τους ασθενείς τους μέσω ραντεβού η σχέση μεταξύ των μοντέλων και η δήλωση θα είναι όπως παρακάτω.



```
class Physician < ActiveRecord::Base
  has_many :appointments
  has_many :patients, :through => :appointments
end

class Appointment < ActiveRecord::Base
  belongs_to :physician
  belongs_to :patient
end

class Patient < ActiveRecord::Base
  has_many :appointments
  has_many :physicians, :through => :appointments
end
```

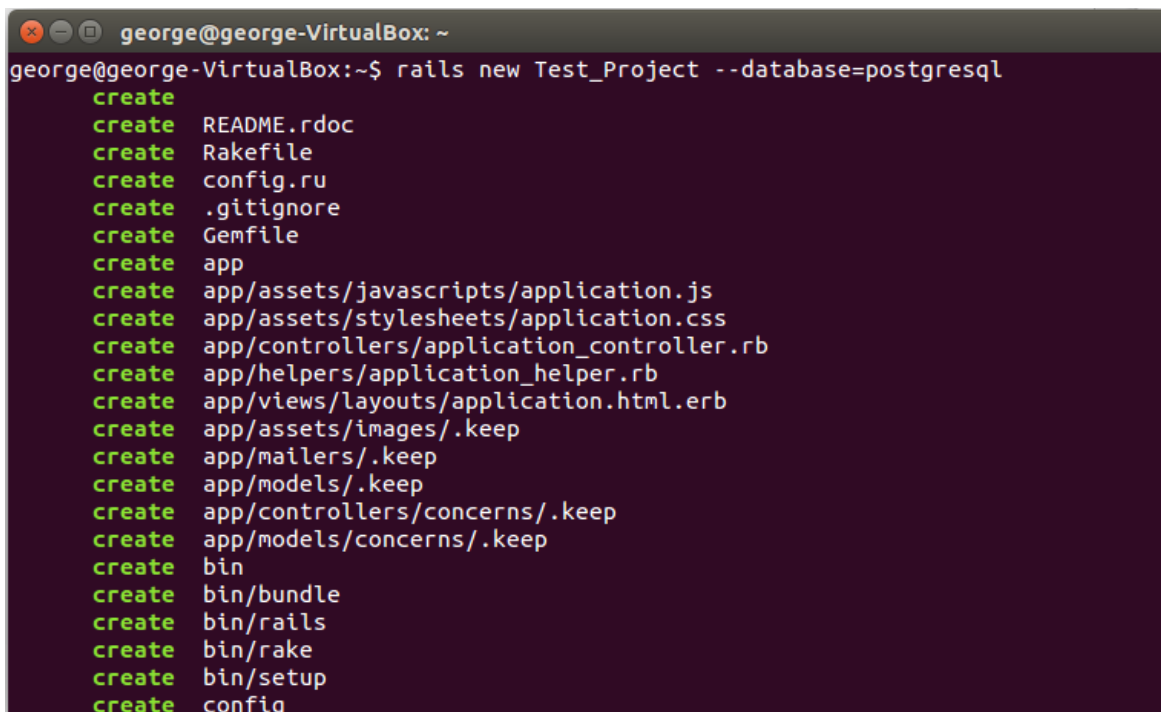
Εικόνα 2.3γ: Παράδειγμα has_many, :through association

Ο πίνακας appointments περιέχει 2 πεδία, τα physician_id και patient_id που αποτελούν τα κλειδιά που πραγματοποιούν τη σύνδεση physicians - patients μέσω αυτού του πίνακα.

2.8 Βασικές εντολές της rails & δημιουργία ενός project

Σε αυτή την ενότητα θα δοθούν πληροφορίες για τις βασικές εντολές και τα βήματα που χρειάζονται για να δημιουργηθεί ένα project με rails. **Σημείωση:** Οι παρακάτω ενέργειες έγιναν σε virtual machine με εγκατεστημένο το λειτουργικό σύστημα ubuntu.

Βήμα 1^ο: Δημιουργία του root φακέλου της εφαρμογής.



```
george@george-VirtualBox: ~
george@george-VirtualBox:~$ rails new Test_Project --database=postgresql
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/javascripts/application.js
create  app/assets/stylesheets/application.css
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/assets/images/.keep
create  app/mailers/.keep
create  app/models/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  bin/setup
create  config
```

Εικόνα 2.4α: Δημιουργία του βασικού φακέλου της εφαρμογής

Επεξήγηση της εντολής: **rails new Test_Project --database=postgresql:**

rails new: Το rails new χρησιμοποιείται όταν ο χρήστης θέλει να φτιάξει μια καινούρια εφαρμογή rails.

Test_Project: Το όνομα της εφαρμογής.

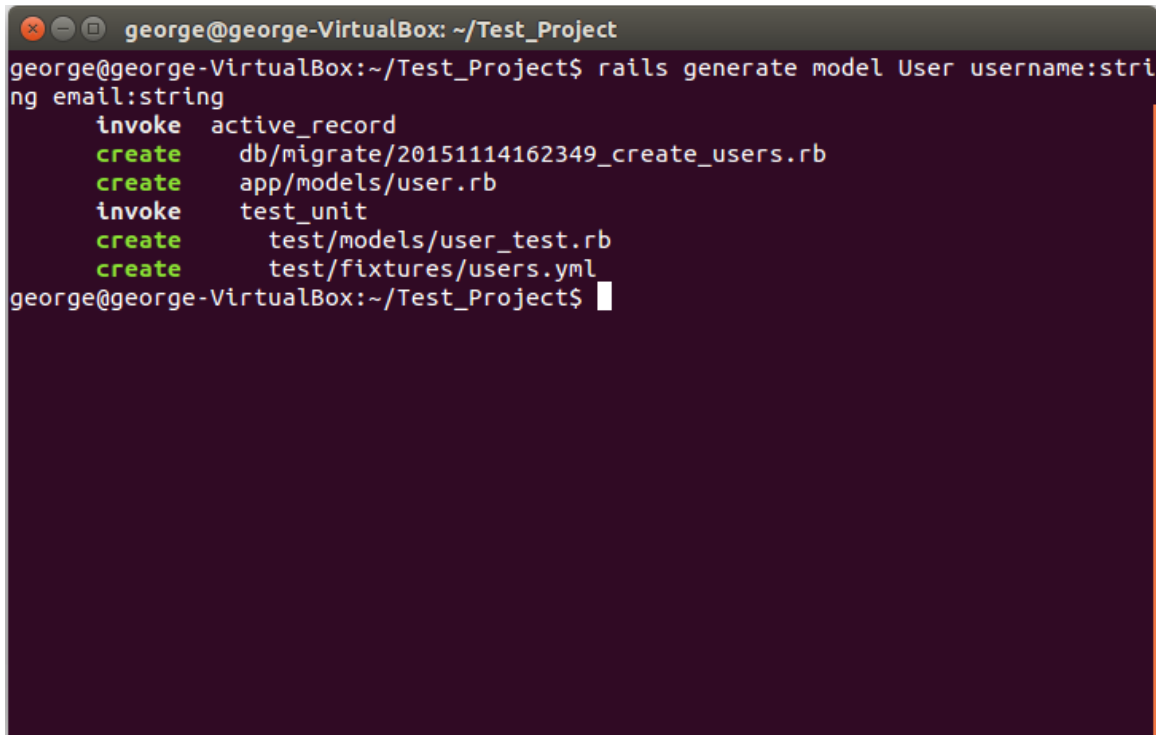
--database = postgresql: Η δήλωση της βάσης δεδομένων που θα χρησιμοποιηθεί

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

στην εφαρμογή. Στην πτυχιακή εργασία χρησιμοποιείτε η postgresql.

Μόλις εκτελεστεί η εντολή δημιουργείται ο root φάκελος της εφαρμογής καθώς και όλοι οι υποφάκελοι που τον αποτελούν (ανάλυση του κάθε φακέλου στην επόμενη ενότητα). Επίσης εκτελείται αυτόματα η εντολή `bundle install` που θα αναλυθεί σε επόμενο βήμα.

Βήμα 2^ο: Κατασκευή του 1^{ου} μοντέλου της εφαρμογής (συνήθως το μοντέλο του χρήστη).



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rails generate model User username:string email:string
  invoke  active_record
  create  db/migrate/20151114162349_create_users.rb
  create  app/models/user.rb
  invoke  test_unit
  create  test/models/user_test.rb
  create  test/fixtures/users.yml
george@george-VirtualBox:~/Test_Project$
```

Εικόνα 2.4β: Κατασκευή μοντέλου

Επεξήγηση της εντολής **rails generate model User username:string email:string** :

rails generate: Με το rails generate ο χρήστης μπορεί να φτιάξει τα κύρια στοιχεία μιας εφαρμογής όπως τα μοντέλα, τους controllers καθώς και άλλα αρχεία που χρησιμοποιούνται στην εφαρμογή π.χ. migration files. **Σημείωση**: Συνήθως

χρησιμοποιείται η συντομογραφία rails g.

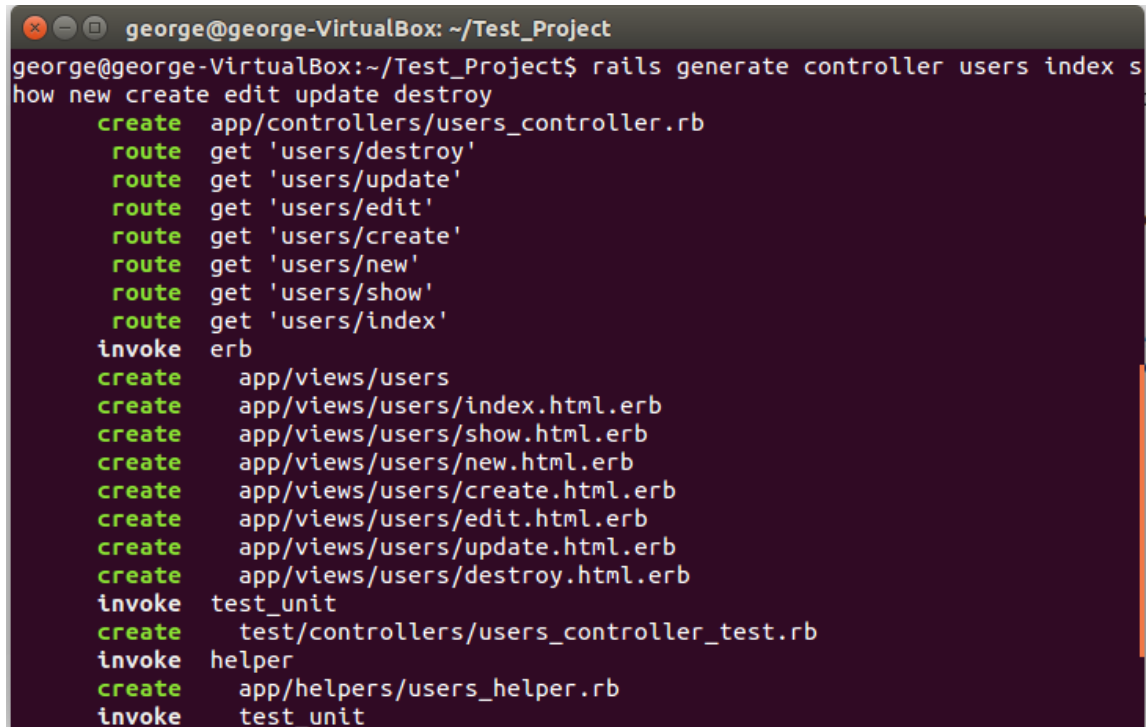
model: Το στοιχείο που κατασκευάζεται με την εκτέλεση της εντολής. Στη συγκεκριμένη περίπτωση δημιουργείται ένα μοντέλο. Σε επόμενο βήμα θα κατασκευαστεί ένας controller για την εφαρμογή.

user: Το όνομα του μοντέλου που κατασκευάστηκε. Στις περισσότερες εφαρμογές το πρώτο μοντέλο που δημιουργείται είναι αυτό του χρήστη (User) . **Σημείωση:** Το όνομα του μοντέλου πρέπει να ξεκινάει πάντα με κεφαλαίο γράμμα.

username: string, email: string: Τα χαρακτηριστικά του μοντέλου. Στη συγκεκριμένη εφαρμογή ο χρήστης θα έχει username και email όπου και τα 2 θα είναι string. Μπορούν να προστεθούν και άλλα χαρακτηριστικά όπως password, ημερομηνία γέννησης κ.α.

Μετά την εκτέλεση της εντολής δημιουργούνται τα αρχεία που φαίνονται στην παραπάνω εικόνα. Αυτά που χρησιμοποιούνται είναι το αρχείο create_users.rb που κατασκευάζει στη βάση δεδομένων τον πίνακα για τον User και το αρχείο user.rb που αποτελεί το κεντρικό αρχείο του μοντέλου όπου εκεί γράφονται τα associations με άλλα μοντέλα, διάφορες μέθοδοι για αυτο κ.α.

Βήμα 3^ο: Κατασκευή του controller για το συγκεκριμένο μοντέλο



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rails generate controller users index s
how new create edit update destroy
  create  app/controllers/users_controller.rb
  route  get 'users/destroy'
  route  get 'users/update'
  route  get 'users/edit'
  route  get 'users/create'
  route  get 'users/new'
  route  get 'users/show'
  route  get 'users/index'
  invoke erb
  create  app/views/users
  create  app/views/users/index.html.erb
  create  app/views/users/show.html.erb
  create  app/views/users/new.html.erb
  create  app/views/users/create.html.erb
  create  app/views/users/edit.html.erb
  create  app/views/users/update.html.erb
  create  app/views/users/destroy.html.erb
  invoke test_unit
  create  test/controllers/users_controller_test.rb
  invoke helper
  create  app/helpers/users_helper.rb
  invoke test_unit
```

Εικόνα 2.4γ: Κατασκευή του controller

Επεξήγηση της εντολής **rails generate controller users index show new create edit update destroy**:

rails generate: Αναφέρθηκε σε προηγούμενο βήμα.

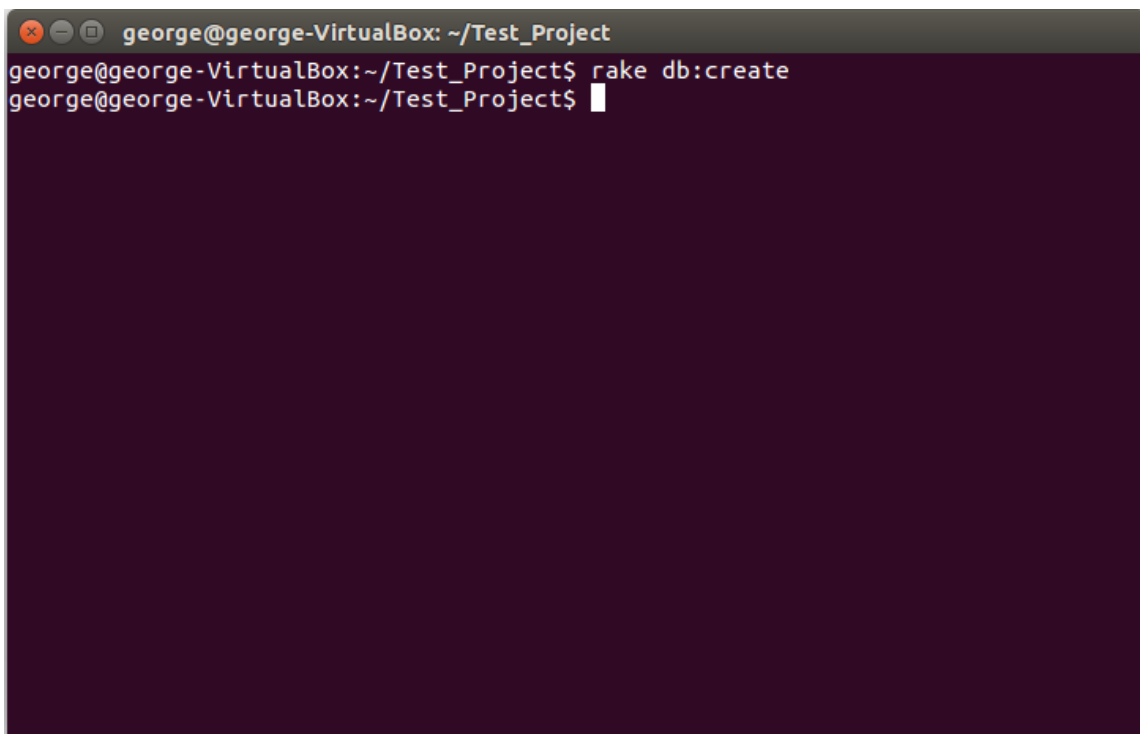
controller: Η λέξη που αντιστοιχεί στην δημιουργία του controller όπως στο προηγούμενο βήμα ήταν η λέξη model για το μοντέλο.

users: Το όνομα του controller όπου συνήθως είναι και το όνομα του μοντέλου στο οποίο αντιστοιχεί. **Σημείωση**: Το όνομα πρέπει να είναι στον πληθυντικό αριθμό και να ξεκινάει με μικρό γράμμα.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

index, show, new, create, edit, update, destroy: Οι μέθοδοι που θα δημιουργηθούν στο αρχείο του controller, αποτελούν τις μεθόδους C.R.U.D (create read update delete) με τα *index, show* να αντιστοιχούν στην ανάγνωση (read), τα *new - create* στη δημιουργία (create), τα *edit - update* στην αναβάθμιση (update) και το *destroy* στη διαγραφή (delete). Τέλος δημιουργούνται αυτόματα τα views για κάθε μία από αυτές περιλαμβάνοντας ένα default μήνυμα.

Βήμα 4^ο: Δημιουργία της τοπικής βάσης δεδομένων



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rake db:create
george@george-VirtualBox:~/Test_Project$
```

Εικόνα 2.4δ: Δημιουργία της βάσης δεδομένων

Επεξήγηση της εντολής **rake db:create**:

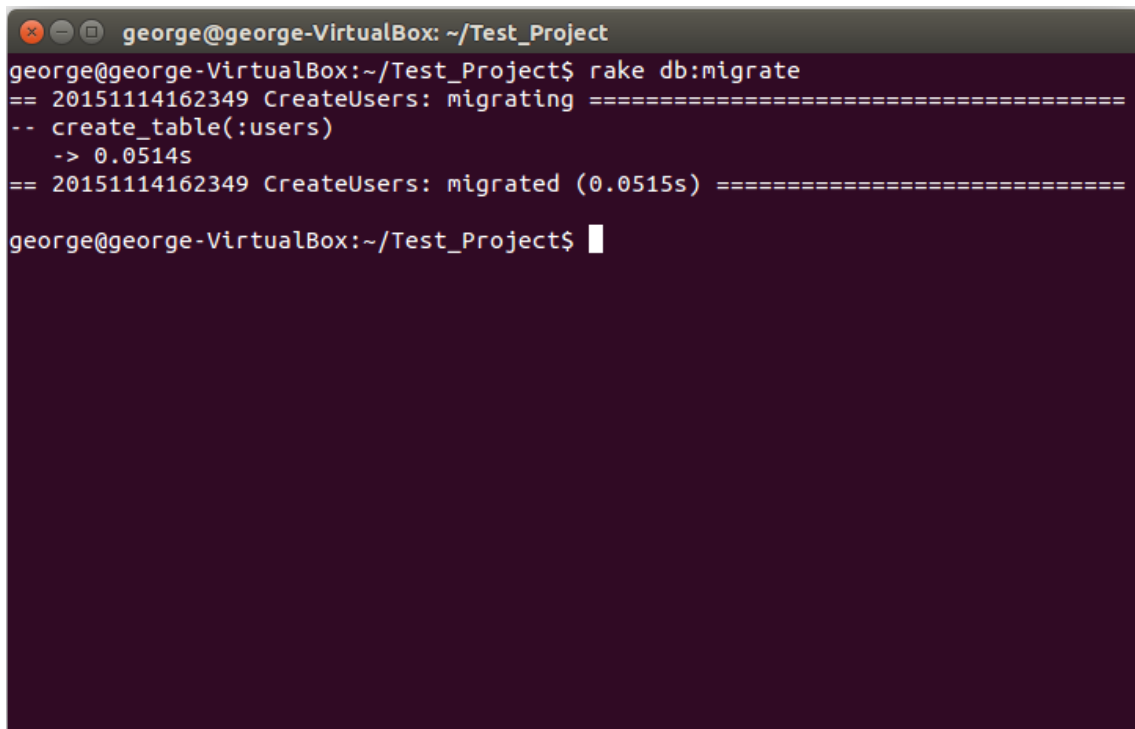
rake: Η λέξη *rake* χρησιμοποιείται όταν εκτελούνται εντολές που αφορούν βάσεις δεδομένων.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

db:create: Με το *db:create* δημιουργείται η βάση δεδομένων τοπικά στον υπολογιστή.

Με την εκτέλεση της εντολής δημιουργείται η βάση δεδομένων της εφαρμογής στον τοπικό υπολογιστή.

Βήμα 5^ο: Μεταφορά στη βάση των στοιχείων που έχουν δημιουργηθεί στην εφαρμογή



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rake db:migrate
== 20151114162349 CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0514s
== 20151114162349 CreateUsers: migrated (0.0515s) =====
george@george-VirtualBox:~/Test_Project$
```

Εικόνα 2.4ε: Μεταφορά των στοιχείων στη βάση

Επεξήγηση της εντολής **rake db:migrate**:

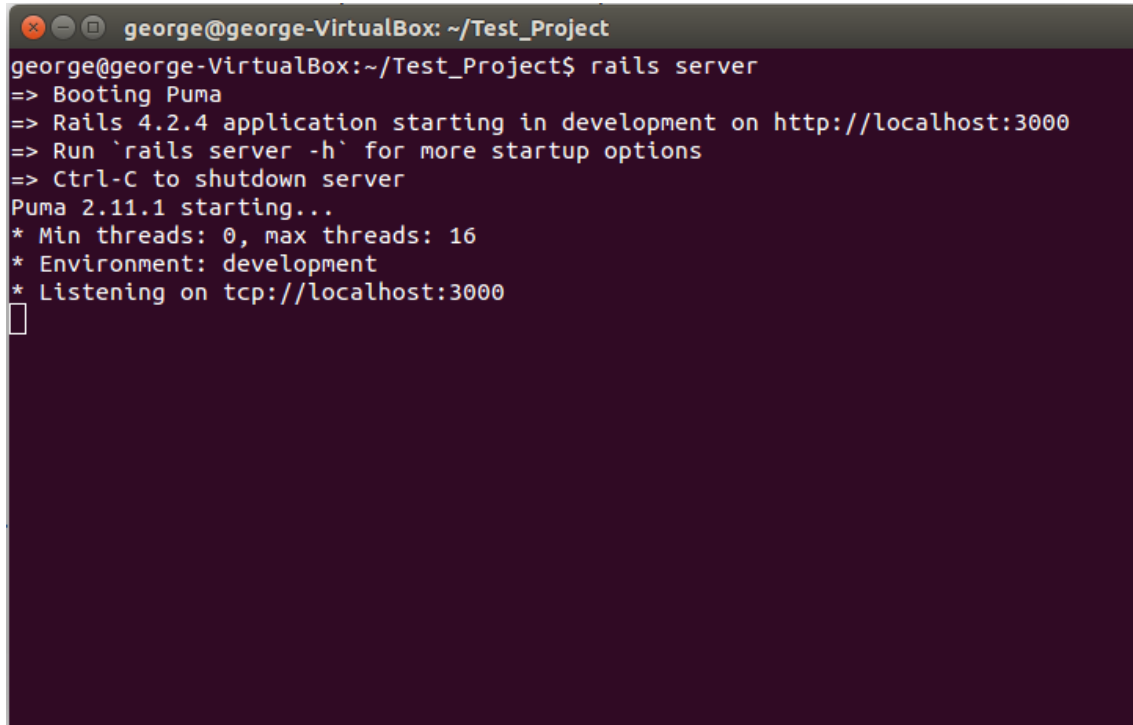
rake: Αναφέρθηκε στο προηγούμενο βήμα.

db:migrate: Με το *db:migrate* δημιουργούνται στη βάση δεδομένων οι πίνακες που αντιστοιχούν στα μοντέλα της εφαρμογής. Εδώ δημιουργείται ο πίνακας Users με τη βοήθεια του αρχείου *create_users.rb* που αναφέρθηκε παραπάνω.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Με την εκτέλεση της εντολής δημιουργείτε ο πίνακας Users στη τοπική βάση δεδομένων και είναι πλέον δυνατή η επεργασία του.

Βήμα 6^ο: Έναρξη του τοπικού server για έλεγχο της λειτουργίας της εφαρμογής.



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rails server
=> Booting Puma
=> Rails 4.2.4 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
Puma 2.11.1 starting...
* Min threads: 0, max threads: 16
* Environment: development
* Listening on tcp://localhost:3000
█
```

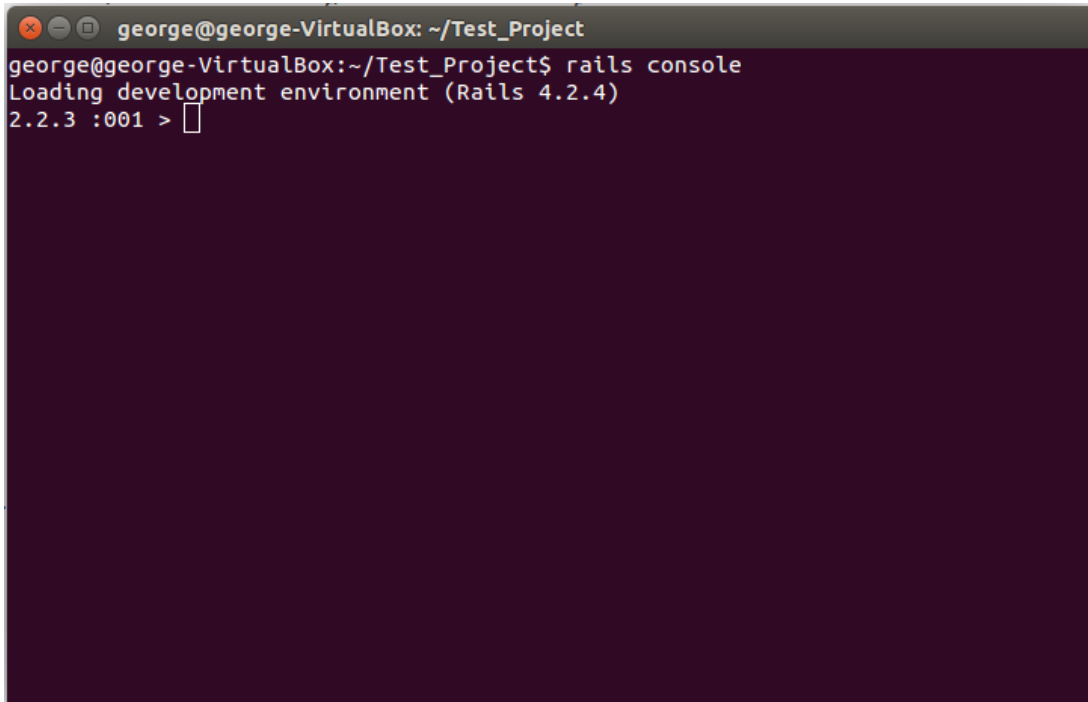
Εικόνα 2.4στ: Έναρξη του τοπικού server

Επεξήγηση της εντολής **rails server**:

rails server. Με την εντολή rails server (rails s για συντομία) γίνεται εκκίνηση του τοπικού server που απαντά στη διεύθυνση localhost:3000. **Σημείωση:** Για αλλαγή της πόρτας 3000 προστίθεται η φράση -p “αριθμός πόρτας” ‘έτσι π.χ. για την πόρτα 3001 η εντολή θα γίνονταν rails server -p 3001. **Σημείωση:** Η πόρτα για τον τοπικό server δε πρέπει να χρησιμοποιείται από άλλες εφαρμογές αλλιώς θα εμφανίσει μήνυμα λάθους.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Βήμα 7^ο: Ένα βήμα το οποίο μπορεί και να παραληφθεί είναι η έναρξη της κονσόλας της rails.



```
george@george-VirtualBox: ~/Test_Project
george@george-VirtualBox:~/Test_Project$ rails console
Loading development environment (Rails 4.2.4)
2.2.3 :001 > 
```

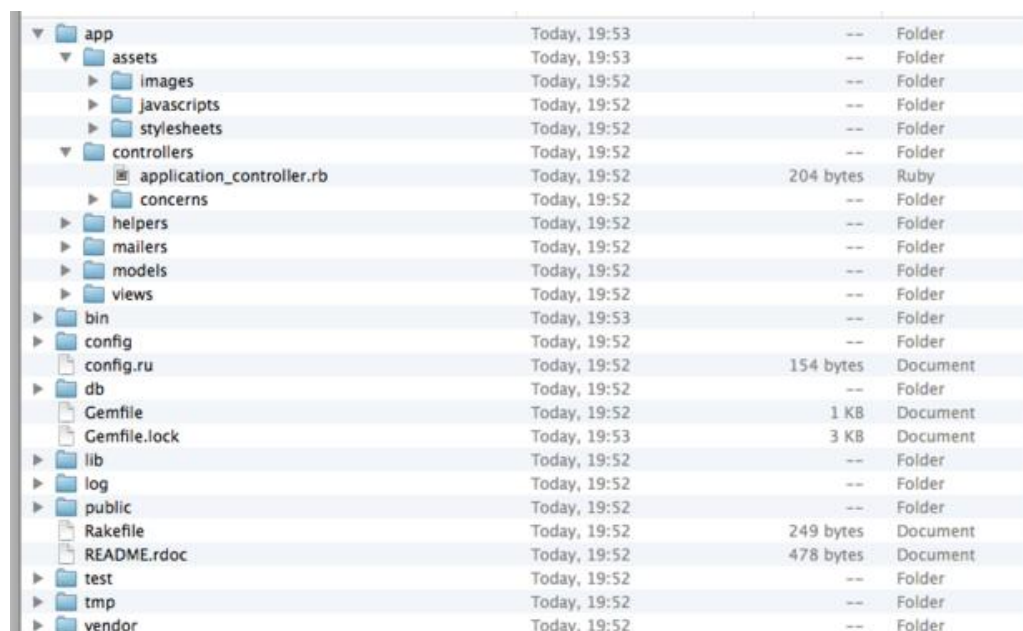
Εικόνα 2.4ζ: Έναρξη της κονσόλας

Επεξήγηση της εντολής **rails console**:

`rails console` : Με την εντολή αυτή ξεκινάει η κονσόλα της rails, όπου μπορούν να εκτελεστούν διάφορες διεργασίες και δοκιμές π.χ. πρόσθεση δεδομένων σε κάποιον πίνακα.

2.9 Οι φάκελοι ενός rails project

Στην παρακάτω εικόνα φαίνονται οι φάκελοι που αποτελούν ένα τυπικό rails project [14].



Εικόνα 2.5: Ένα rails project

Οι πιο σημαντικοί φάκελοι που χρησιμοποιούνται περισσότερο είναι:

Φάκελος *app*: Ο σημαντικότερος φάκελος της εφαρμογής, περιέχει τα σημαντικότερα στοιχεία όπως είναι τα αρχεία για τα μοντέλα (*models*), τις σελίδες (*views*), τους χειριστές (*controllers*) άλλα και διάφορα βοηθήματα.

Υποφάκελος *app/assets*: Περιλαμβάνει διάφορα στοιχεία (*assets*) της εφαρμογής όπως αρχεία CSS, αρχεία Javascript και εικόνες.

Φάκελος *config*: Περιλαμβάνει τη διαμόρφωση της εφαρμογής.

Φάκελος *db*: Περιλαμβάνει τα αρχεία που σχετίζονται με τη βάση δεδομένων της εφαρμογής.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Φάκελος test: Περιλαμβάνει τα test αρχεία της εφαρμογής.

Αρχείο README.rdoc: Είναι ένα αρχείο που περιέχει μια συνοπτική περιγραφή της εφαρμογής.

Αρχείο Rakefile: Κάνει διαθέσιμες τις απαραίτητες διεργασίες μέσω της εντολής rake (π.χ. rake db create για τη δημιουργία βάσης δεδομένων για την εφαρμογή).

Αρχείο Gemfiles: Περιέχει όλα τα gems που χρησιμοποιούνται στην εφαρμογή.

Αρχείο Gemfiles.lock: Επιβεβαιώνει ότι όλες οι αντιγραφές της εφαρμογής χρησιμοποιούν τις ίδιες εκδόσεις των gem.

Αρχείο .gitignore: Περιλαμβάνει τα στοιχεία (π.χ. φακέλους, αρχεία) που θέλουμε να αγνοήσει το git.

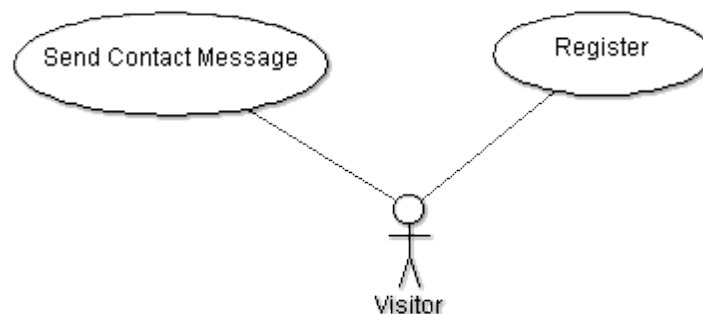
ΚΕΦΑΛΑΙΟ 3

ΕΙΣΑΓΩΓΗ

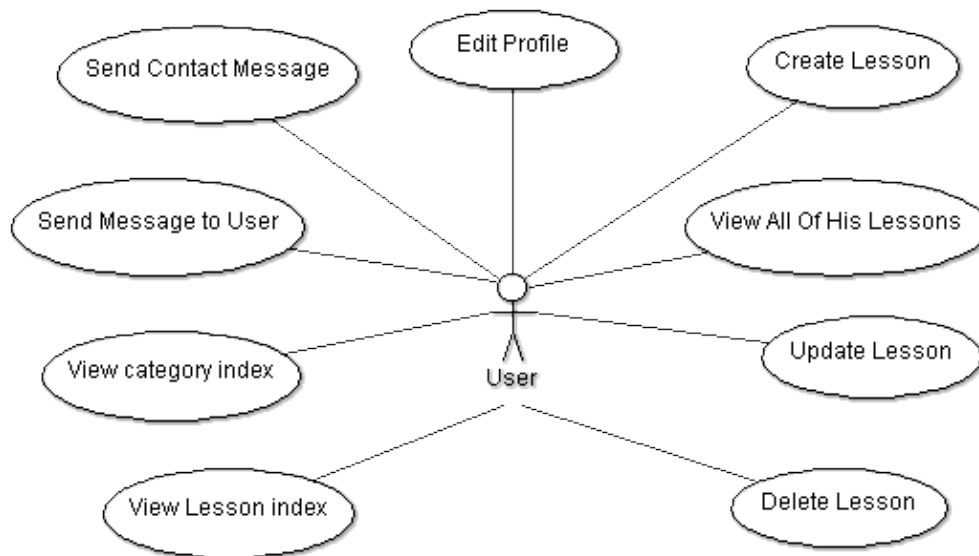
Στο 3^ο και τελευταίο κεφάλαιο θα γίνει ανάλυση της εφαρμογής. Θα γίνουν αναφορές στα στοιχεία που την αποτελούν, το user interface της και πιο σημαντικό στις λειτουργίες της και τις δυνατότητες που προσφέρει σε κάθε χρήστη.

3.1 Οι δυνατότητες των χρηστών της εφαρμογής

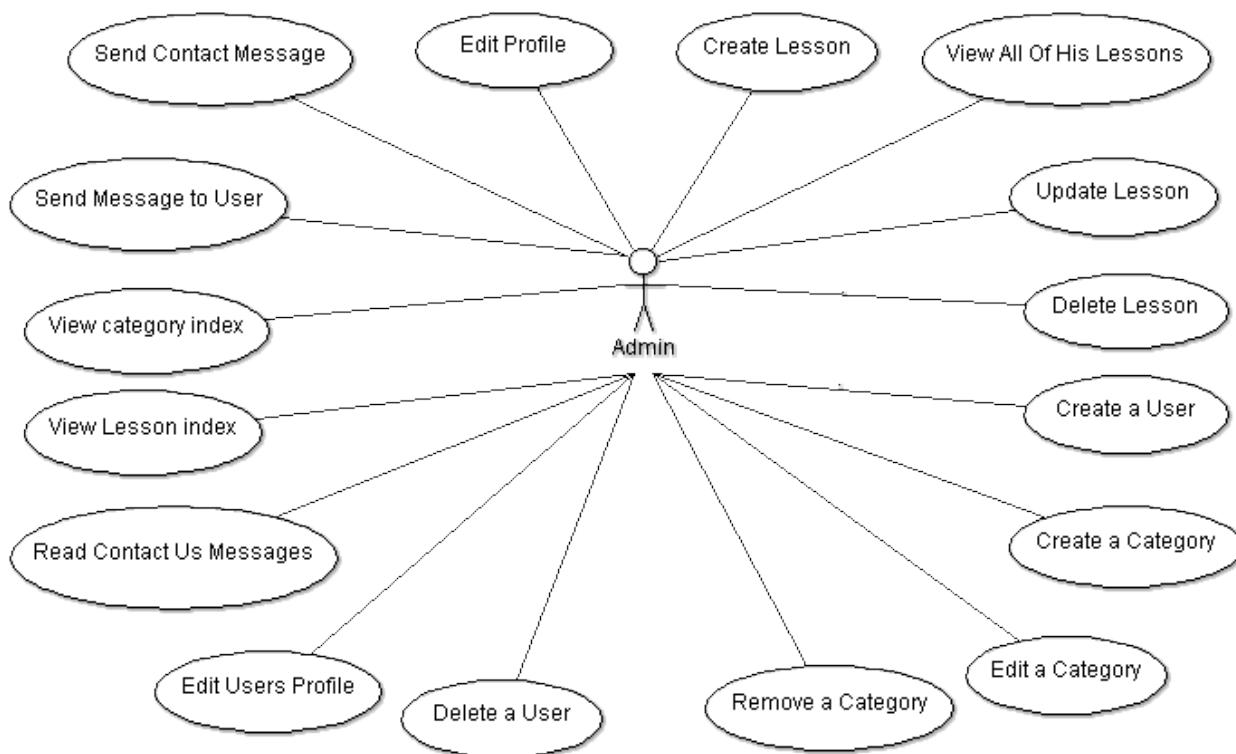
Κάθε χρήστης της εφαρμογής έχει διαφορετικά δικαιώματα και δυνατότητες στη χρήση της. Τα 3 είδη χρηστών είναι ο επισκέπτης (visitor), ο απλός χρήστης (user), και ο διαχειριστής (admin). Ο επισκέπτης περιορίζεται στις βασικές επιλογές (π.χ. εγγραφή στην εφαρμογή, επίσκεψη βοηθητικών σελίδων), στον απλό χρήστη διατίθενται οι βασικές επιλογές της εφαρμογής (π.χ. δημιουργία μαθήματος) ενώ στον διαχειριστή είναι διαθέσιμες όλες οι επιλογές που αφορούν την εφαρμογή (π.χ. αλλαγή των στοιχείων όλων των χρηστών). Στα παρακάτω use case UML διαγράμματα περιγράφονται οι δυνατότητες που έχουν τα 3 πιθανά είδη χρηστών της εφαρμογής.



Εικόνα 3.1: Δυνατότητες επισκέπτη



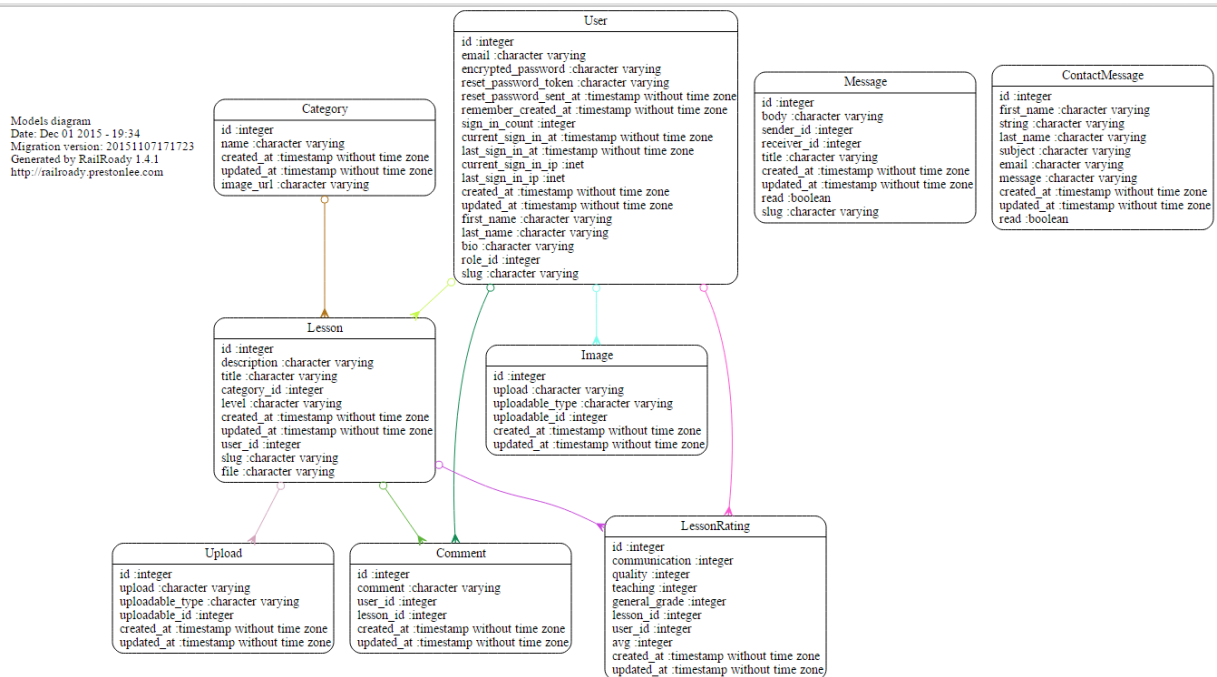
Εικόνα 3.2: Δυνατότητα απλού χρήστη



Εικόνα 3.3: Δυνατότητες του διαχειριστή

3.2 Το σχέδιο της βάσης δεδομένων

Στο παρακάτω σχέδιο αποτυπώνεται η βάση δεδομένων της εφαρμογής. Ο κάθε πίνακας αποτελεί και ένα μοντέλο στην εφαρμογή με τα πεδία του να είναι τα χαρακτηριστικά του. Με τα βελάκια συμβολίζονται τα associations μεταξύ των διάφορων μοντέλων.



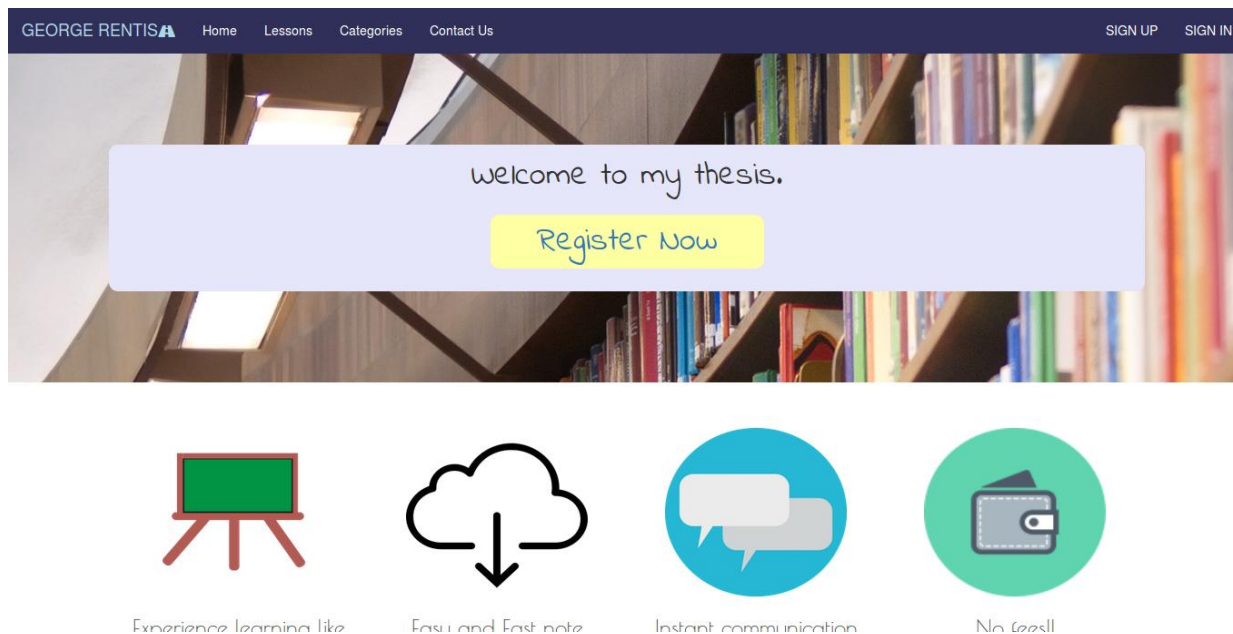
Εικόνα 3.4: Το σχηματικό της βάσης δεδομένων

3.3 Ανάλυση της εφαρμογής

Σε αυτή την ενότητα θα γίνει ανάλυση της εφαρμογής. Θα περιγραφεί το περιεχόμενό της καθώς και οι δυνατότητες που προσφέρει σε όλους τους χρήστες (επισκέπτης, απλός χρήστης, διαχειριστής).

3.3.1 Η αρχική σελίδα

Στην αρχική σελίδα της εφαρμογής υπάρχει το μενού με τις βασικές επιλογές οι οποίες είναι home (σύνδεσμος για την αρχική σελίδα), Lessons (σύνδεσμος για τη λίστα με τα διαθέσιμα μαθήματα), Categories (σύνδεσμος για τη λίστα με τις διαθέσιμες κατηγορίες), Contact Us (σύνδεσμος για τη φόρμα επικοινωνίας με το διαχειριστή), SIGN UP/SIGN IN (σύνδεσμοι για εγγραφή ή είσοδο του χρήστη αντίστοιχα).



Εικόνα 3.5: Αρχική σελίδα της εφαρμογής

Ανάλογα με τον τύπο του χρήστη (διαχειριστής ή απλός χρήστης) ξεκλειδώνονται και άλλες επιλογές που θα αναφερθούν σε επόμενη ενότητα. Οι απλοί επισκέπτες έχουν πρόσβαση στην επιλογή Contact Us, στις βοηθητικές σελίδες και φυσικά στις επιλογές SIGN UP/SIGN IN. Σε περίπτωση που ένας επισκέπτης πατήσει σε μια από τις άλλες επιλογές θα του εμφανιστεί μήνυμα λάθους.

- **Επιλογή Contact Us**

Η επιλογή Contact Us είναι διαθέσιμη προς όλους τους επισκέπτες της εφαρμογής. Στη φόρμα υπάρχουν πεδία για το first name, last name, subject, email και message. Πατώντας το κουμπί Create Contact Message στέλνεται το μήνυμα του επισκέπτη/χρήστη στον διαχειριστή της εφαρμογής. Η μορφή της φόρμας αποστολής είναι η παρακάτω.

The screenshot shows the 'Contact Us' page of the GEORGE RENTIS application. The header contains the logo and navigation links: Home, Lessons, Categories, Contact Us, SIGN UP, and SIGN IN. The main heading is 'Contact Us'. The form consists of the following elements:

- First name:** Input field with the value 'George'.
- Last name:** Input field.
- Subject:** Input field.
- Email:** Input field.
- Message:** Textarea.
- Create Contact message:** A blue button.
- Contact Information:** George, Μπουρούλινας 6, Νίκαια, Attica, TE 18453, Phone: 6945613919, email: georent123@hotmail.com.

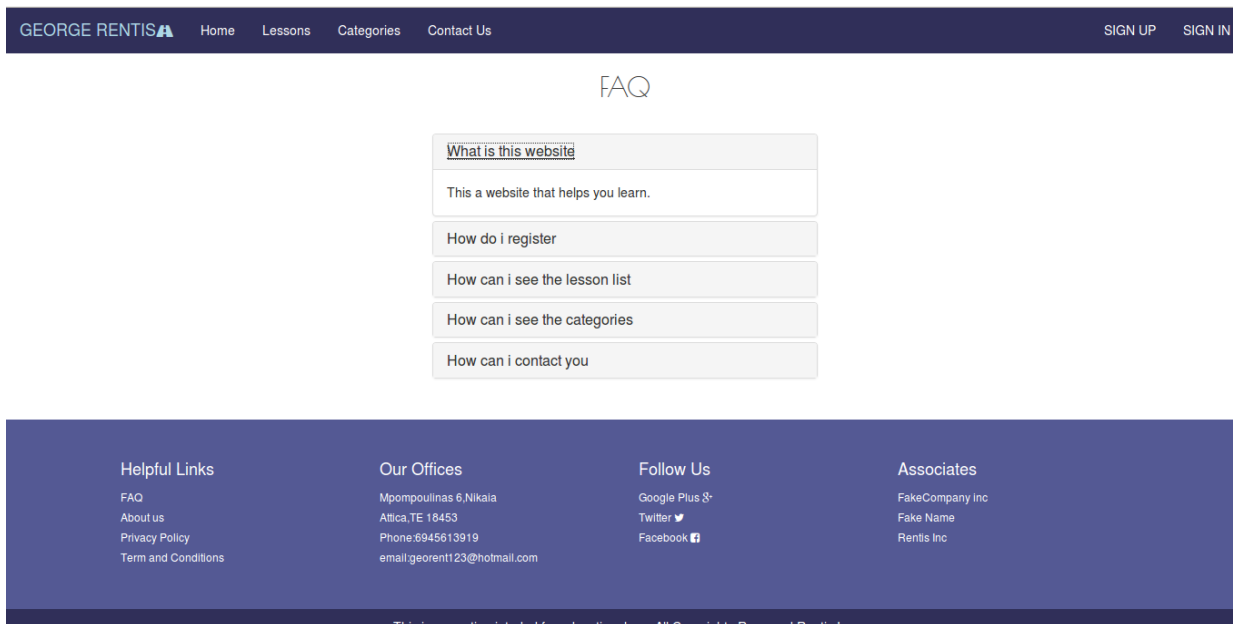
The footer contains four columns: Helpful Links, Our Offices, Follow Us, and Associates.

Εικόνα 3.6: Φόρμα επικοινωνίας Contact Us

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

- **Βοηθητικές σελίδες της εφαρμογής**

Στη παρακάτω εικόνα φαίνεται η βοηθητική σελίδα FAQ. Με παρόμοιο τρόπο έχουν κατασκευαστεί και οι υπόλοιπες 3 σελίδες (About Us, Privacy Policy, Terms and Conditions).



Εικόνα 3.7: Η FAQ σελίδα

3.3.2 Η σελίδα εγγραφής χρήστη

Πατώντας την επιλογή SIGN UP εμφανίζεται η φόρμα εγγραφής του χρήστη. Για την εγγραφή του απαιτείται η συμπλήρωση όλων των πεδίων (first name, last name, email, password, password confirmation) εκτός από το bio και το add new image διότι μπορεί να μη θέλει να χρησιμοποιήσει κάποιο avatar. Αφού συμπληρώσει τα απαραίτητα στοιχεία (και είναι σωστά) με το πάτημα του κουμπιού Create User δημιουργείται ο χρήστης και γίνεται αυτόματα sign in στην εφαρμογή αλλιώς εμφανίζεται μήνυμα λάθους.

Εικόνα 3.8: Σελίδα εγγραφής χρήστη

Στη συνέχεια θα ακολουθήσει η περιγραφή της εγγραφής του χρήστη μέσω του συγκεκριμένου κομματιού κώδικα της εφαρμογής. Πιο συγκεκριμένα θα αναλυθεί ο κώδικας του μοντέλου του χρήστη, της φόρμας εγγραφής και του user controller.

- **Ανάλυση του κώδικα για το μοντέλο χρήστη**

```
class User < ActiveRecord::Base
  extend FriendlyId
  friendly_id :email, use: :slugged
  has_many :images, as: :uploadable, dependent: :destroy
  has_many :comments, dependent: :destroy
  has_many :lessons, dependent: :destroy
  has_many :lesson_ratings
  accepts_nested_attributes_for :images, allow_destroy: true
  devise:database_authenticatable,:recoverable,:rememberable,:trackable, :validatable
  enum role_id: [ :admin, :user ]
  validates :first_name,:last_name,:email, presence: true
  validates :email, uniqueness: true
end
```

Ο κώδικας για το μοντέλο χρήστη περιλαμβάνει τα χαρακτηριστικά, τα associations και τα validations του. Με τη βοήθεια του FriendlyId gem ορίζεται το slug του χρήστη που στην συγκεκριμένη περίπτωση αντιστοιχεί στο email του **[16]**. Ύστερα ορίζονται τα associations του χρήστη με άλλα μοντέλα με το has_many, αυτό σημαίνει ότι κάθε χρήστης της εφαρμογής θα έχει πολλά images, comments, lessons, lesson_ratings. Στη συνέχεια ορίζονται οι ιδιότητες του devise για το συγκεκριμένο μοντέλο (database_authenticatable, recoverable, rememberable, trackable, validatable) **[17]**. Για τον καθορισμό του ρόλου του χρήστη στην εφαρμογή χρησιμοποιείται ένας enumerator στο πεδίο role_id, αυτός δηλώνει πως αν η τιμή του πεδίου είναι 0 ο χρήστης θα είναι admin ενώ αν είναι 1 (που είναι η default τιμή) θα είναι απλός χρήστης. Τέλος δηλώνονται τα validations του χρήστη τα οποία είναι η υποχρεωτική παρουσία first_name, last_name, email και η μοναδικότητα του e-mail του.

- **Κώδικας για τη φόρμα εγγραφής χρήστη**

```
<%= nested_form_for([:user,@user],:html => {:class => 'form-horizontal
col-md-4 col-md-offset-4 col-sm-4 col-sm-offset-4'}) do |f| %>
  <h2 class="register_title">User registration page />
  <div class = "form-group">
    <%= f.fields_for :images do |image| %>
      <%= image.file_field :upload, class: "form-control" %>
      <%= image.link_to_remove "Remove this image", class: "btn
btn-danger
      btn-xs" %>
    <% end %>
    <p><%= f.link_to_add "Add a new image", :images, class: "btn
btn-primary
      btn-xs" %></p>
  </div>
  <div class = "form-group">
    <%= f.label :first_name,class: "control-label",for: "first_name"%>
    <%= f.text_field :first_name,class: "form-control",id: "first_name"
%>
  </div>
  <div class = "form-group">
    <%= f.label :last_name,class: "control-label"%>
    <%= f.text_field :last_name,class: "form-control" %>
  </div>
  <div class = "form-group">
    <%= f.label :email,class: "control-label"%>
    <%= f.text_field :email,class: "form-control" %>
  </div>
  <div class = "form-group">
    <%= f.label :bio,class: "control-label"%>
    <%= f.text_area :bio,class: "form-control" %>
  </div>
  <div class = "form-group">
    <%= f.label :password,class: "control-label"%>
    <%= f.password_field :password,class: "form-control" %>
  </div>
  <div class = "form-group">
    <%= f.label :password_confirmation,class: "control-label"%>
    <%= f.password_field :password_confirmation,class: "form-control"
%>
  </div>
  <div class = "form-group">
    <%= f.submit class: "btn btn-success btn-md" %>
  </div>
</div>
<% end %>
```

Για την εγγραφή του χρήστη κατασκευάστηκε μία φόρμα με χρήση της html και της rails ενώ για την μορφοποίηση των στοιχείων της (textfields, labels) χρησιμοποιούνται οι κλάσεις του bootstrap. Η φόρμα αντιστοιχεί στο μοντέλο χρήστη και αποθηκεύει τα στοιχεία που θα δώσει κατά την εγγραφή στην προσωρινή μεταβλητή @user που έχει

δημιουργηθεί στον controller του. Όταν ο χρήστης συμπληρώσει όλα τα απαραίτητα πεδία πατά στο κουμπί submit και αν όλα είναι σωστά προστίθεται η καταχώρηση του στη βάση δεδομένων.

- **Ο κώδικας για το User Controller**

```
class User::UsersController < ApplicationController

  def index
    @comments = current_user.lessons.last.try(:comments)
  end

  def create
    @user = User.new(user_params)
    if @user.save
      sign_in :user, @user
      redirect_to root_path
    else
      redirect_to root_path
    end
  end

  def new
    @user = User.new

  private
  def user_params

    params[:user].permit(:first_name,:last_name,:email,:bio,:role_id,:password,:password_confirmation,images_attributes: [:id, :upload, :_destroy])
  end

end
```

Ο κώδικας του controller αποτελείται από τις μεθόδους που περιγράφουν τις λειτουργίες του χρήστη. Εδώ οι μέθοδοι είναι index, create, new, user_params (οι υπόλοιπες μέθοδοι είναι show, edit, update, destroy, find_user αλλά δε χρησιμοποιούνται στη διαδικασία εγγραφής γι'αυτό έχουν αφαιρεθεί από το συγκεκριμένο κομμάτι κώδικα). Για την διαδικασία της εγγραφής καινούργιου χρήστη, χρησιμοποιούνται οι new, create και user_params. Όταν ο χρήστης πατάει στην επιλογή sign up τρέχει η μέθοδος new δημιουργώντας έτσι τη μεταβλητή @user που θα χρησιμοποιηθεί στη φόρμα εγγραφής χρήστη όπως αναφέρθηκε στην προηγούμενη ενότητα. Αφότου ο χρήστης πατήσει το κουμπί submit εκτελείται η

μέθοδος `user_params` που δείχνει για πια πεδία επιτρέπεται να περαστούν τιμές στη βάση για αυτόν. Σε περίπτωση που λείπει κάποιο πεδίο από τα `permits` και δεν απαιτείται η παρουσία του μέσω των `validation` τότε θα πάρει την τιμή `nil`. Τέλος με τη μέθοδο `create` δημιουργείται μια νέα μεταβλητή `@user` η οποία παίρνει τις παραμέτρους που περάστηκαν μέσω της φόρμας εγγραφής και αποθηκεύεται στη βάση δεδομένων σαν τον τελικό χρήστη. Αν υπάρξει κάποιο λάθος στην εγγραφή εμφανίζεται το αντίστοιχο μήνυμα

3.3.3 Το προφίλ του χρήστη

Μετά την επιτυχή είσοδο του χρήστη οι επιλογές sign up/sign in έχουν αντικατασταθεί από το email του που ταυτόχρονα αποτελεί και την επιλογή εξόδου από την εφαρμογή (log out). Επίσης προστέθηκε η επιλογή εμφάνισης του προφίλ του (Your profile). Με το πάτημα της επιλογής αυτής ο χρήστης μεταφέρεται στην επισκόπηση του προσωπικού του λογαριασμού (εικόνα 3.9). Η σελίδα αποτελείται από 3 κύρια μέρη. Στα αριστερά εμφανίζονται οι λειτουργίες που παρέχονται στο χρήστη, στο κέντρο δίνονται διάφορα στατιστικά στοιχεία του ενώ στα δεξιά εμφανίζονται τα τελευταία 5 σχόλια που έχουν γίνει σε κάποιο δικό του μάθημα (αν υπάρχουν).

The screenshot shows the user profile page for 'GEORGE RENTIS'. The page is divided into several sections:

- Navigation:** Home, Lessons, Categories, Contact Us, Your Profile. User: testuser@test.com - Logout.
- Left Sidebar (Your Profile):**
 - Your profile
 - Account Settings
 - Forgot your password
 - Manage Lessons
 - Your Lessons
 - New Lesson
 - Your Messages
 - Received
 - Sent
- Statistics:**

Statistics	
Last signed in	2015-12-10 19:16:49 UTC
Lessons	0
Files Uploaded	0
- Lesson Statistics:**

Lesson Statistics	
Physics	0
Chemistry	0
Vocabulary and Writing	0
Others	0
Business and Economics	0
Biology	0
Programming	0
Languages	0
Electronics	0
Literature	0
Engineering	0
Math	0
- Footer:**
 - Helpful Links:** FAQ, About us
 - Our Offices:** Μπουρούλιας 6, Νίκαια, Αττική, ΤΕ 15453
 - Follow Us:** Google Plus, Twitter
 - Associates:** FakeCompany Inc, Fake Name

Εικόνα 3.9: Το προφίλ του χρήστη

Οι λειτουργίες του χρήστη χωρίζονται σε 3 κατηγορίες:

Your Profile: Σε αυτή την κατηγορία υπάρχουν οι επιλογές που αφορούν την επεξεργασία των στοιχείων του χρήστη. Στην επιλογή Account Settings μπορεί να αλλάξει τα στοιχεία του ενώ χρησιμοποιεί την επιλογή forgot your password σε περίπτωση που έχει ξεχάσει τον κωδικό του.

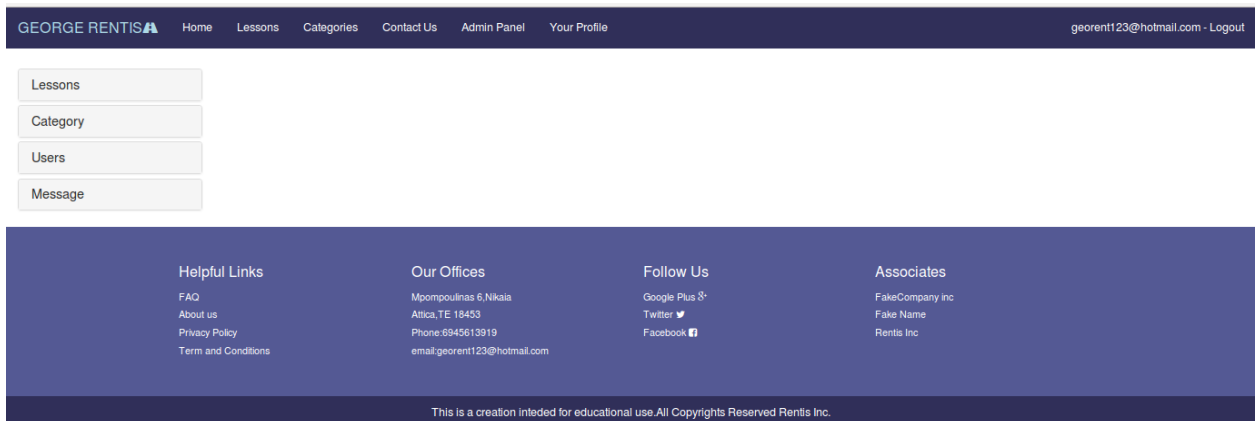
Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Manage Lessons: Σε αυτή την κατηγορία υπάρχουν οι επιλογές που αφορούν τα μαθήματα του χρήστη. Μπορεί να φτιάξει το δικό του μάθημα (επιλογή New Lesson) ή να δει τα μαθήματα που έχει ήδη φτιάξει και να κάνει αλλαγές σε αυτά (επιλογή Your Lessons).

Your Messages: Σε αυτή την κατηγορία υπάρχουν οι επιλογές που αφορούν τα μηνύματα του χρήστη. Με την επιλογή received μπορεί να δει τα μηνύματα που έχει λάβει και αντίστοιχα με την επιλογή sent αυτά που έχει στείλει.

3.3.4 Σύνδεση του admin

Σε αυτή την ενότητα θα γίνει σύνδεση στην εφαρμογή με το λογαριασμό του διαχειριστή δείχνοντας έτσι τις διαφορετικές επιλογές που έχει σε σχέση με τον απλό χρήστη. Όταν γίνεται είσοδος με λογαριασμό διαχειριστή, μαζί με την επιλογή Your Profile ξεκλειδώνεται και η επιλογή Admin Panel που ουσιαστικά αποτελεί το ειδικό μενού του διαχειριστή (εικόνα 3.10).



Εικόνα 3.10: Το admin panel

Οι επιλογές του admin είναι παρόμοιες με αυτές του χρηστή με μερικές διαφορές όπως:

- Η επιλογή categories όπου εκεί μπορεί να φτιάξει μια κατηγορία ή να αλλάξει μια υπάρχουσα.

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

- Στην επιλογή lessons εμφανίζει τα μαθήματα όλων των χρηστών της εφαρμογής.
- Η πιο σημαντική επιλογή στο μενού του admin είναι οι users όπου πατώντας την εμφανίζονται τα στοιχεία όλων των χρηστών της εφαρμογής (εικόνα 3.11). Εκεί δίνεται στον διαχειριστή η δυνατότητα να επεξεργαστεί τα στοιχεία τους, να τους στείλει ένα μήνυμα και να διαγράψει το λογαριασμό κάποιου χρήστη τελείως από την εφαρμογή.

GEORGE RENTIS Home Lessons Categories Contact Us Admin Panel Your Profile georent123@hotmail.com - Logout

Lessons Category Users Message

User List

First Name	Last Name	User Rating	Email	Bio	Role	Action
George	Rentis	No Rating	test123@test.com	test	user	Conversations Send a message Edit Delete
George	Rentis	5	georent123@hotmail.com	Ela	admin	You are the current user Edit Delete

Helpful Links: FAQ, About us, Privacy Policy, Term and Conditions

Our Offices: Μπουμπουλίνας 6, Νίκαια, Αττική, ΤΕ 18453, Phone: 6945613919, email: georent123@hotmail.com

Follow Us: Google Plus, Twitter, Facebook

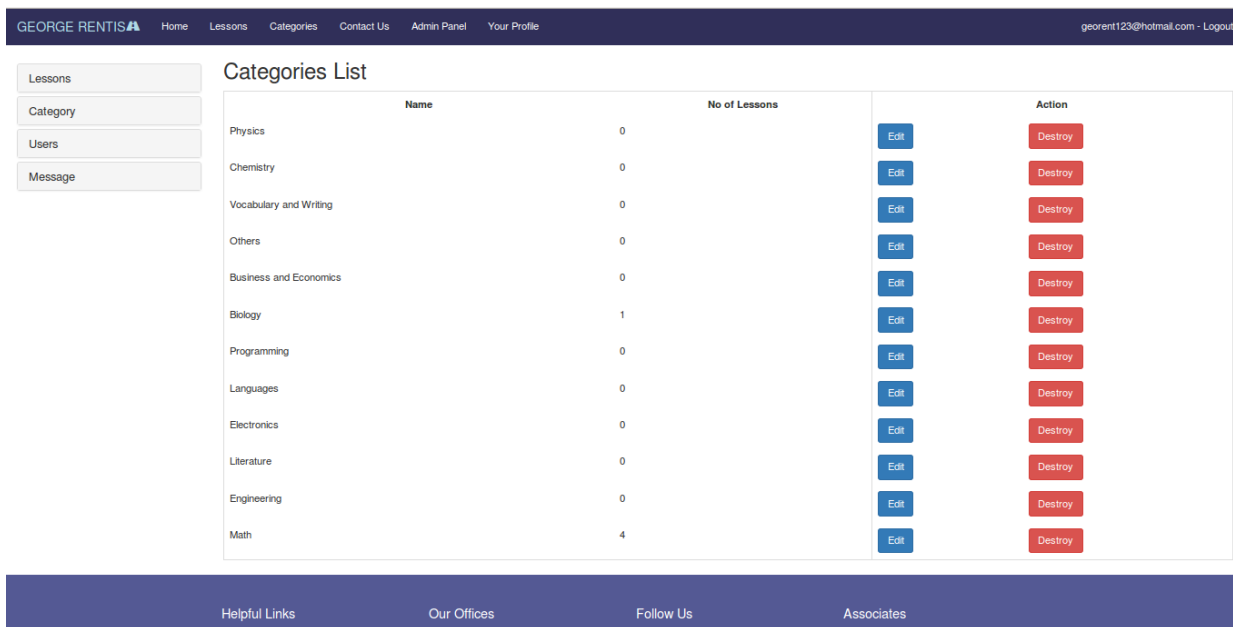
Associates: FakeCompany Inc, Fake Name, Rentis Inc

This is a creation intended for educational use. All Copyrights Reserved Rentis Inc.

Εικόνα 3.11: Εμφάνιση όλων των λογαριασμών της εφαρμογής

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

Στην επιλογή Categories εμφανίζεται ένας πίνακας που περιέχει το όνομα κάθε κατηγορίας αλλά και τον αριθμό των μαθημάτων που έχουν δημιουργηθεί για κάθε μία απο αυτές (εικόνα 3.12), ο διαχειριστής μπορεί να επεξεργαστεί μια κατηγορία ή να την διαγράψει τελείως από την εφαρμογή.

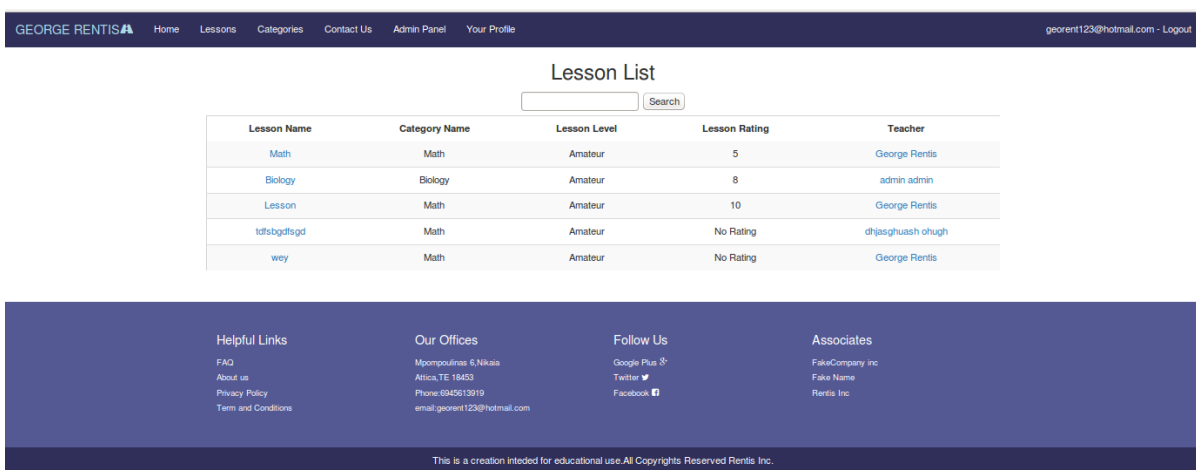


Name	No of Lessons	Action
Physics	0	Edit Destroy
Chemistry	0	Edit Destroy
Vocabulary and Writing	0	Edit Destroy
Others	0	Edit Destroy
Business and Economics	0	Edit Destroy
Biology	1	Edit Destroy
Programming	0	Edit Destroy
Languages	0	Edit Destroy
Electronics	0	Edit Destroy
Literature	0	Edit Destroy
Engineering	0	Edit Destroy
Math	4	Edit Destroy

Εικόνα 3.12: Λίστα με τις κατηγορίες της εφαρμογής

3.3.5 Η επιλογή Lessons

Όταν ο χρήστης διαλέξει την επιλογή Lessons εμφανίζεται η λίστα με όλα τα μαθήματα της εφαρμογής (εικόνα 3.13). Στον πίνακα παρουσιάζονται όλα τα διαθέσιμα μαθήματα μαζί με τις απαραίτητες πληροφορίες (το όνομά τους, την κατηγορία που ανήκουν, το επίπεδό, το βαθμό αξιολόγησής και τον υπεύθυνο τους). Για διευκόλυνση των χρηστών έχει προστεθεί ένα πεδίο αναζήτησης των μαθημάτων με βάση την ονομασία τους ενώ αν ο χρήστης πατήσει στο τίτλο κάθε στήλης θα γίνει ταξινόμηση των περιεχομένων της (αλφαβητική ή αριθμητική).



Lesson Name	Category Name	Lesson Level	Lesson Rating	Teacher
Math	Math	Amateur	5	George Rentis
Biology	Biology	Amateur	8	admin admin
Lesson	Math	Amateur	10	George Rentis
tdfsgdfsgd	Math	Amateur	No Rating	d@jasghush ohugh
way	Math	Amateur	No Rating	George Rentis

Helpful Links
FAQ
About us
Privacy Policy
Term and Conditions

Our Offices
Μπουρούλιας Σ, Νίκαια
Αττική 15453
Phone: 6945613919
email: georent123@hotmail.com

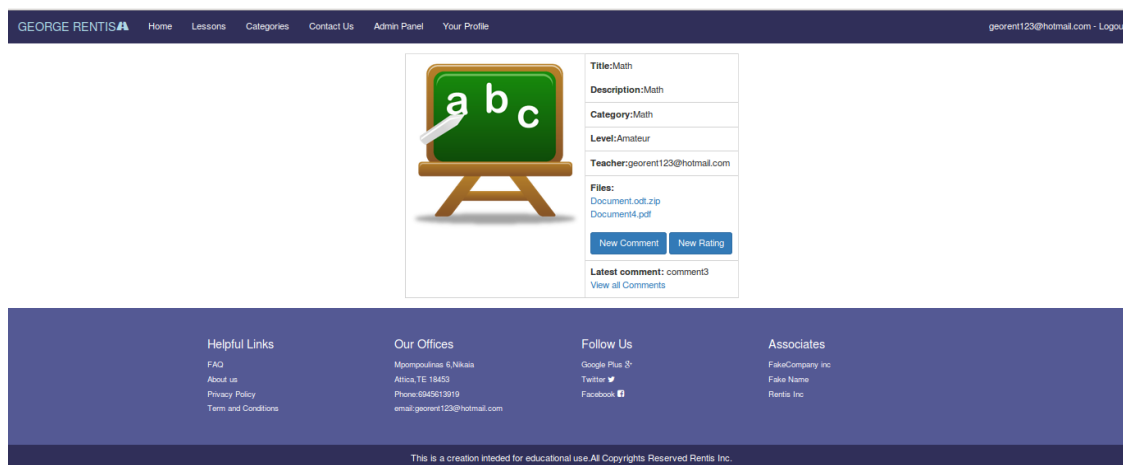
Follow Us
Google Plus
Twitter
Facebook

Associates
FakeCompany inc
Fake Name
Rentis Inc.

This is a creation intended for educational use. All Copyrights Reserved Rentis Inc.

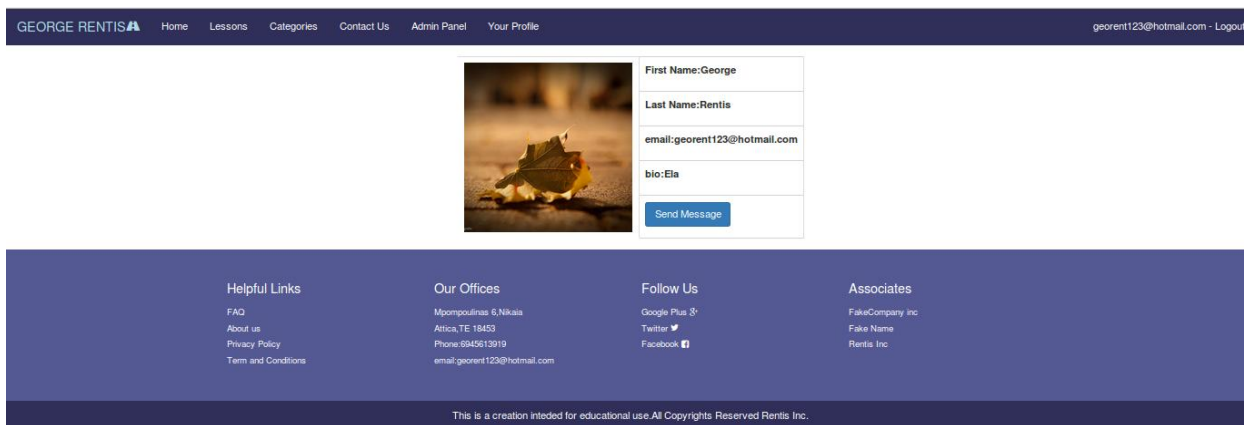
Εικόνα 3.13: Λίστα με όλα τα μαθήματα

Πατώντας πάνω στο όνομα ενός μαθήματος ο χρήστης μεταφέρεται στην ειδική σελίδα του (εικόνα 3.14). Για κάθε μάθημα εμφανίζονται περισσότερες πληροφορίες όπως μια περιγραφή (description), το email του υπευθύνου και πιο σημαντικό το υλικό που έχει ανεβάσει για το μάθημα. Επίσης περιέχει 2 κουμπιά με ονομασίες New Comment, New Rating που είναι για τη δημιουργία ενός σχολίου ή μιας αξιολόγησης για το συγκεκριμένο μάθημα. Τέλος εμφανίζεται το τελευταίο σχόλιο που έχει γίνει για το μάθημα καθώς και επιλογή να φανούν όλα τα σχόλια που έχουν γίνει μέχρι στιγμής.



Εικόνα 3.14: Σελίδα ενός μαθήματος

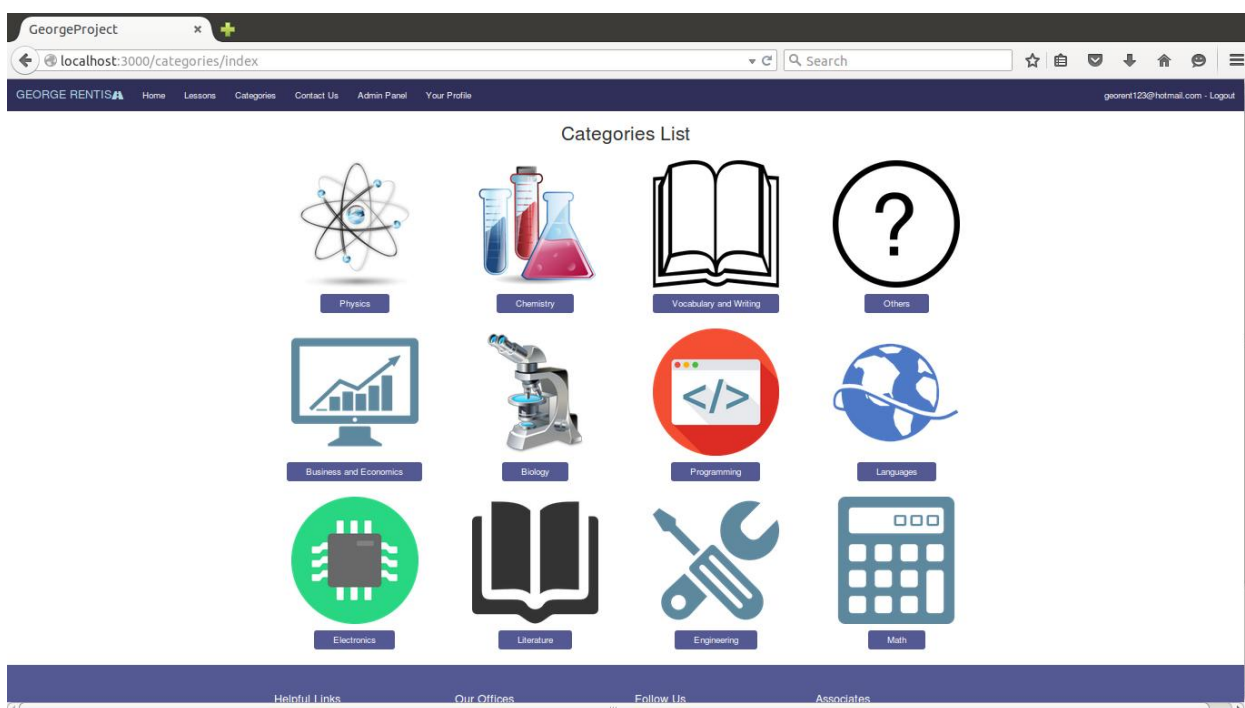
Αν ο χρήστης πατήσει στο όνομα του υπεύθυνου του μαθήματος μεταφέρεται στο δημόσιο προφίλ του (εικόνα 3.15) όπου εκεί εμφανίζονται όλα τα απαραίτητα στοιχεία του καθώς και ένα κουμπί Send Message μέσω του οποίου δημιουργείται ένα μήνυμα προς αυτόν.



Εικόνα 3.15: Δημόσιο προφίλ χρήστη

3.3.6 Η επιλογή Categories

Οι δώδεκα κατηγορίες της εφαρμογής είναι Math, Physics, Chemistry, Vocabulary and Writing, Others, Business and Economics, Biology, Programming, Reading, Languages, Electronics και Engineering. Για κάθε μια κατηγορία υπάρχει μια εικόνα – εξώφυλλο που αντιπροσωπεύει όσο το δυνατόν καλύτερα το αντικείμενο της. Ο χρήστης έχει την δυνατότητα πατώντας επάνω στο κουμπί της κάθε κατηγορίας να εμφανίσει τα μαθήματα που ανήκουν μόνο σε αυτή. Όταν ο χρήστης διαλέξει την επιλογή Categories εμφανίζεται η παρακάτω σελίδα:



Εικόνα 3.16: Οι κατηγορίες της εφαρμογής

3.3.7 Δημιουργία Μαθήματος

Η φόρμα της δημιουργίας μαθήματος φαίνεται στην επόμενη εικόνα (εικόνα 3.17). Στο πεδίο title ο χρήστης συμπληρώνει τον τίτλο που θέλει να έχει το μάθημα του, ενώ στο description γράφει μια συνοπτική περιγραφή. Με το selectbox level επιλέγει το επίπεδο δυσκολίας (Amateur, Intermediate, Professional, World Class, Master) ενώ με το category διαλέγει σε ποια κατηγορία από τις 12 διαθέσιμες ανήκει το μάθημα του. Τέλος με το κουμπί add new file ανεβάζει το υλικό του μαθήματος για να έχουν πρόσβαση σε αυτό όλοι οι ενδιαφερόμενοι χρήστες και πατώντας το κουμπί create lesson, αν δεν υπάρχει κάποιο λάθος στα στοιχεία, καταχωρείται το μάθημα στη βάση δεδομένων.

The screenshot shows a web browser window with the URL localhost:3000/user/lessons/new. The page title is 'GEORGE RENTIS'. The navigation menu includes Home, Lessons, Categories, Contact Us, Admin Panel, and Your Profile. The user is logged in as georent123@hotmail.com. The main content area is a form for creating a new lesson. It has a sidebar on the left with links like 'Your profile', 'Account Settings', 'Forgot your password', 'Manage Lessons', 'Your Lessons', 'New Lesson', 'Your Messages', 'Contact Messages', 'Received', and 'Sent'. The form fields are: 'Title' (text input), 'Description' (text area), 'Level' (dropdown menu with 'Amateur' selected), and 'Category' (dropdown menu with 'Select Category' selected). There are two buttons: 'Add a new file' (blue) and 'Create Lesson' (green). The footer contains 'Helpful Links' (FAQ, About us, Privacy Policy, Term and Conditions), 'Our Offices' (address, phone, email), 'Follow Us' (Google Plus, Twitter, Facebook), and 'Associates' (FakeCompany Inc, Fake Name, Rentis Inc). A copyright notice is at the bottom: 'This is a creation intended for educational use. All Copyrights Reserved Rentis Inc.'

Εικόνα 3.17: Φόρμα δημιουργίας μαθήματος

3.4 Τα test της εφαρμογής

Στην τελευταία ενότητα αναφέρονται τα τεστ που πραγματοποιήθηκαν έτσι ώστε να επιβεβαιωθεί ο σωστός τρόπος λειτουργίας της εφαρμογής. Στόχος των ελέγχων είναι η λογική του black-box testing όπου γίνεται η επιβεβαίωση των στοιχείων που έχει δώσει ο χρήστης βάσει των κανόνων δημιουργίας profile. Για την παραγωγή των τεστ χρησιμοποιήθηκαν τα gems Rspec, Capybara, Selenium Webdriver και FactoryGirl [18-21] με τη βοήθεια του οποίου θα φτιάξουμε έναν χρήστη αποκλειστικά για τα tests που θα έχει το ρόλο του admin. Ο κώδικας κατασκευής του είναι ο ακόλουθος:

```
FactoryGirl.define do
  factory :user do |f|
    f.email "email@email.com"
    f.first_name "George"
    f.last_name "Rentis"
    f.password "password123"
    f.password_confirmation "password123"
  end
end
```

Ο χρήστης κατασκευάζεται απονέμοντας του τα απαραίτητα στοιχεία (email, όνομα, κωδικό). Σύμφωνα με τα σενάρια χρήσης της εφαρμογής (όπου χρειάζεται) θα του αποδίδεται και ο ρόλος του admin.

3.4.1 Test για τα validations του μοντέλου User

Παρακάτω αναλύονται 4 test που έχουν γραφεί για να επιβεβαιωθεί ότι τα validation στο μοντέλο του user (υποχρεωτική παρουσία first_name, last_name, email και η μοναδικότητα του e-mail του) δουλεύουν σωστά. Γι'αυτή τη δουλειά θα χρησιμοποιηθεί ο ψεύτικος χρήστης που κατασκευάστηκε μέσω του FactoryGirl. Ο κώδικας των τεστ είναι ο ακόλουθος:

- Έλεγχος για το μοντέλο User

```
describe User do
  it "has a valid email" do
    FactoryGirl.create(:user).should be_valid
  end
  it "has invalid password" do
    FactoryGirl.build(:user,password:"password").should_not be_valid
  end
  it "has to have a first name" do
    FactoryGirl.build(:user,first_name:nil).should_not be_valid
  end
  it "has invalid email(same email twice)" do
    FactoryGirl.create(:user)
    FactoryGirl.build(:user,email: "email@email.com").should_not
be_valid
  end
end
```

Τα 4 τεστ που πραγματοποιούνται είναι έλεγχος για έγκυρο email (has a valid email), για λάθος password (has invalid password), ύπαρξη first_name (has first name) και χρήση ενός email παραπάνω από μία φορά (has invalid email - same email twice). Όταν θέλουμε το αποτέλεσμα του ελέγχου να είναι έγκυρο χρησιμοποιούμε τη μέθοδο should be_valid ενώ αν όχι τη should_not be_valid. Τελικά αφού ο χρήστης έχει email το 1^ο τεστ θα περάσει με επιτυχία, στο 2^ο τεστ απαιτείται ο κωδικός του χρήστη να μην είναι "password" πράγμα που σημαίνει ότι και αυτό το τεστ πέτυχε (ο κωδικός του ψεύτικου χρήστη είναι "password123"). Το 3^ο τεστ περνάει εφόσον ο χρήστης έχει first_name και το τελευταίο το ίδιο αφού με βάση τα validation που έχουν οριστεί δεν μπορεί κάποιος να κάνει εγγραφή με το ίδιο email.

```
george@george-VirtualBox: ~/george_project
george@george-VirtualBox:~/george_project$ rspec spec/models/user_spec.rb
RSpec's reporter has already been initialized with #<IO:STDERR> as the deprecation stream, so your change
to `deprecation_stream` will be ignored. You should configure it earlier for it to take effect, or use the
`--deprecation-out` CLI option. (Called from /home/george/george_project/spec/spec_helper.rb:39:in `block
in <top (required)>')
Run options:
  include {:focus=>true}
  exclude {:slow=>true}

All examples were filtered out; ignoring {:focus=>true}

Randomized with seed 2630
.....

Deprecation Warnings:

RSpec::Core::Configuration#treat_symbols_as_metadata_keys_with_true_values= is deprecated, it is now set to
true as default and setting it to false has no effect.

Using `should` from rspec-expectations' old `:should` syntax without explicitly enabling the syntax is depr
ecated. Use the new `:expect` syntax or explicitly enable `:should` with `config.expect_with(:rspec) { |c|
c.syntax = :should }` instead. Called from /home/george/george_project/spec/models/user_spec.rb:7:in `block
(2 levels) in <top (required)>'.

If you need more of the backtrace for any of these deprecations to
identify where to make the necessary changes, you can configure
`config.raise_errors_for_deprecations!`, and it will turn the
deprecation warnings into errors, giving you the full backtrace.

2 deprecation warnings total

Finished in 1.76 seconds (files took 4.53 seconds to load)
4 examples, 0 failures

Randomized with seed 2630
george@george-VirtualBox:~/george_project$
```

Εικόνα 3.18: Αποτελέσματα των validation tests

3.4.2 Τεστ πραγματικών σεναρίων χρήσης της εφαρμογής

Σε αυτά τα τεστ θα δημιουργηθούν σεναρία που θα προσομοιώνουν τη συμπεριφορά ενός πραγματικού χρήστη στην εφαρμογή. Για την προσομοίωση της συμπεριφοράς θα χρειαστούν οι ακόλουθες εντολές του selenium:

visit path: Με την εντολή αυτή το τεστ επισκέπτεται τη σελίδα που αντιστοιχεί στο path.

click_on "text": Η εντολή αντιστοιχεί στο πάτημα του αριστερού κλικ του ποντικιού στο αντικείμενο με αναγνωριστικό text. Το text μπορεί να είναι κείμενο, σύνδεσμος κ.α.

fill_in "field", with: "text": Η εντολή αυτή προσωμοιώνει το πληκτρολόγιο. Έτσι το τεστ

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

θα πάει στο πεδίο με id "field" (συνήθως θα είναι ένα textfield) και θα γράψει το κείμενο που υπάρχει στο "text".

`click_button "text"`: Η εντολή χρησιμοποιείται για το πάτημα ενός κουμπιού στην εφαρμογή που φέρει την ετικέτα "text".

- **Είσοδος admin και δημιουργία χρήστη από το panel του**

```
context 'admin log in' do
  it 'logs in as admin', js: true do
    user = FactoryGirl.create(:user, role_id: "admin")
    visit root_path
    click_on "SIGN IN"
    fill_in "user_email", with: user.email
    fill_in "user_password", with: user.password
    click_button "Log in"
    click_on "Users"
    click_on "New User"
    fill_in "user_first_name", with: "George"
    fill_in "user_last_name", with: "George"
    fill_in "user_email", with: "arxhgos2@arxhgos2.com"
    fill_in "user_password", with: "password123"
    fill_in "user_password_confirmation", with: "password1234"
    click_button 'Create User'
    sleep(5)
  end
end
```

Ο παραπάνω κώδικας περιγράφει το σενάριο εισόδου ενός admin στην εφαρμογή για δημιουργία ενός χρήστη από το admin panel. Στην αρχή δίνεται ένα όνομα στο σενάριο που περιγράφει τη διαδικασία ("logs in as admin") και κατασκευάζεται ο ψεύτικος χρήστης δίνοντας του ταυτόχρονα ρόλο διαχειριστή. Ύστερα το τεστ ακολουθεί το μονοπάτι που αντιστοιχεί στη δημιουργία ενός χρήστη στο πραγματικό περιβάλλον ξεκινώντας από την επίσκεψη στην αρχική σελίδα της εφαρμογής και τελειώνοντας με τη συμπλήρωση των πεδίων στην φόρμα δημιουργίας νέου χρήστη

- **Εγγραφή χρήστη και δημιουργία μαθήματος**

```
context 'create a lesson' do
  it "creates a lesson",js: true do
    FactoryGirl.create(:category)
    visit root_path
    click_button "Register Now"
    fill_in 'first_name',with: "George"
    fill_in 'user_last_name',with: "George"
    fill_in 'user_email',with: "email@email.com"
    fill_in 'user_password',with: "password123"
    fill_in 'user_password_confirmation',with: "password123"
    click_button 'Create User'
    click on "Your Profile"
    click on "New Lesson"
    fill_in "lesson_title",with: "Lesson"
    fill_in "lesson_description",with: "desc"
    page.select "Amateur",from: "lesson_level"
    page.select "Math",from: "lesson_category_id"
    click_button 'Create Lesson'
    sleep(5)
  end
end
```

Ο παραπάνω κώδικας περιγράφει ένα σενάριο στο οποίο ένας επισκεπτής της εφαρμογής θα κάνει εγγραφή και θα φτιάξει το πρώτο του μάθημα. Το test ξεκινάει με τη δημιουργία μιας κατηγορίας μέσω του FactoryGirl και στη συνέχεια όπως και στο παραπάνω σενάριο ακολουθεί το μονοπάτι δημιουργίας ενός μαθήματος (εγγραφή χρήστη – προφίλ χρήστη – νέο μάθημα).

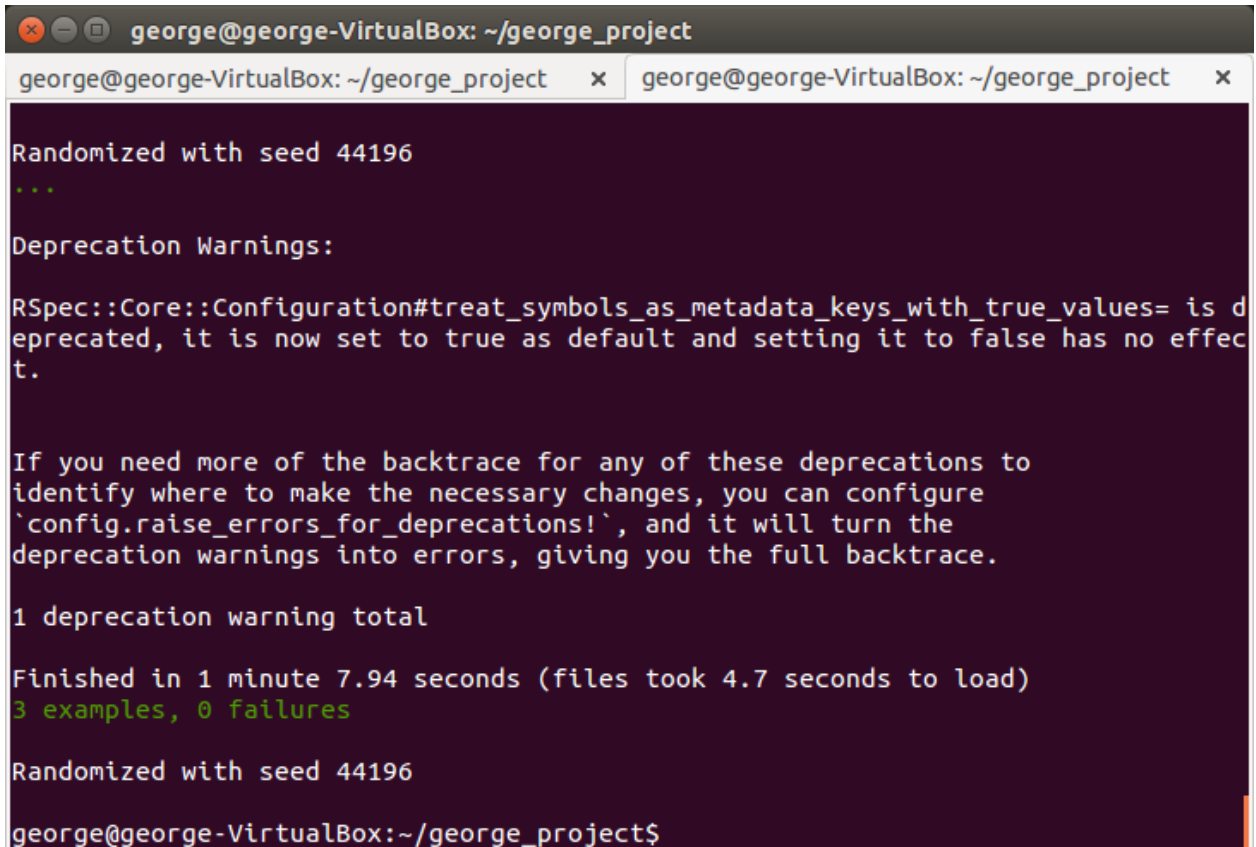
- **Αποστολή μηνύματος απο τη φόρμα Contact Us**

```
context 'send a contact us message' do
  it "sends a contact us message",js: true do
    visit root_path
    click_on "Contact Us"
    fill_in 'contact_message_first_name',with: "George"
    fill_in 'contact_message_last_name',with: "George"
    fill_in 'contact_message_subject',with: "Question"
    fill_in 'contact_message_email',with: "email@email.com"
    fill_in 'contact_message_message',with: "hello"
    click_button 'Create Contact message'
    sleep(5)
  end
end
```

Στον παραπάνω κώδικα περιγράφεται ένα σενάριο για τον απλό επισκέπτη της εφαρμογής αυτό της αποστολής μηνύματος μέσω της φόρμας Contact Us. Εδώ δε γίνεται χρήση του FactoryGirl αφού δε χρειάζεται η κατασκευή κάποιου ψεύτικου

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

μοντέλου γιατί και δίνεται κατευθείαν το μονοπάτι για την κατασκευή ενός μηνύματος (πάτημα στην επιλογή Contact Us – συμπλήρωση των πεδίων).



```
george@george-VirtualBox: ~/george_project
george@george-VirtualBox: ~/george_project x george@george-VirtualBox: ~/george_project x

Randomized with seed 44196
...
Deprecation Warnings:

RSpec::Core::Configuration#treat_symbols_as_metadata_keys_with_true_values= is deprecated, it is now set to true as default and setting it to false has no effect.

If you need more of the backtrace for any of these deprecations to identify where to make the necessary changes, you can configure `config.raise_errors_for_deprecations!`, and it will turn the deprecation warnings into errors, giving you the full backtrace.

1 deprecation warning total

Finished in 1 minute 7.94 seconds (files took 4.7 seconds to load)
3 examples, 0 failures

Randomized with seed 44196
george@george-VirtualBox:~/george_project$
```

Εικόνα 3.19:Αποτελέσματα των test χρήσης

ΕΠΙΛΟΓΟΣ

Στην παρούσα πτυχιακή εργασία αρχικά έγινε αναφορά σε πληροφορίες που αφορούν τη τέχνη του e-learning. Αναφέρθηκε η ιστορική αναδρομή, οι προοπτικές του, τα πλεονεκτήματα και μειονεκτήματα της χρήσης του. Δόθηκαν πληροφορίες για τις υπηρεσίες που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας καθώς και για τις διάφορες τεχνολογίες που χρησιμοποιεί το e-learning. Στο 2^ο κεφάλαιο αναλύθηκε εκτενώς η γλώσσα κατασκευής της εφαρμογής Ruby on Rails δίνοντας βάση στα στοιχεία που την αποτελούν, τις κλάσεις και τις βασικές της εντολές. Στο 3^ο κεφάλαιο περιγράφηκε η εφαρμογή δείχνοντας το τρόπο λειτουργίας της στα δυνατά σενάρια χρήσης της καθώς και έλεγχος της με κατασκευή διαφόρων tests.

Τελικώς έγινε αντιληπτό ότι το e-learning μπορεί να βοηθήσει το μεγαλύτερο ποσοστό του κόσμου που επιθυμεί να μορφωθεί με τρόπο διαφορετικό από τον κλασικό και με καλύτερα αποτελέσματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Epignosis LLC, “E-Learning-concepts, trends, applications” , 2014
- [2] Roberta Gogos, “A brief history of elearning (infographic) , 2013
<http://www.efrontlearning.net/blog/2013/08/a-brief-history-of-elearning-infographic.html>
- [3] e-Learning Fundamentals, History of e-Learning
http://www.leerbeleving.nl/wbts/1/history_of_elearning.html
- [4] Universitat d’Alacant, What is a mooc
<http://desarrolloweb.dlsi.ua.es/moocs/what-is-a-mooc>
- [5] eLearning Industry – Christopher Pappas, “The Top eLearning Statistics and Facts For 2015 You Need To Know”, 2015
<http://elearningindustry.com/elearning-statistics-and-facts-for-2015>
- [6] What is Heroku?
<https://www.heroku.com/what>
- [7] Amazon Web Service
<https://aws.amazon.com/>
- [8] World Wide Web Consortium (W3C),”2 – A history of HTML”
<https://www.w3.org/People/Raggett/book4/ch02.html>
- [9] Hakon Wium Lie, Bert Ros, “Cascading Style Sheets, designing for the Web 2nd edition”, 1999
<https://www.w3.org/Style/LieBos2e/history/Overview.html>

[10] About Bootstrap

<http://getbootstrap.com/about/>

[11] About Ruby

<https://www.ruby-lang.org/en/about/>

[12] TIOBE Software

<http://www.tiobe.com/index.php/content/company/Home.html>

[13] Ruby on Rails Guides (v4.2.5)

<http://guides.rubyonrails.org/index.html>

[14] Michael Hartl, “Ruby on Rails tutorial 3rd edition” , 2014

<http://www.railstutorial.org/book>

[15] Dimitar Zlatkov, “Development of Web Community Application with Ruby on Rails framework”, Bachelor Thesis, University of Applied Sciences Hamburg, 2009

http://edoc.sub.uni-hamburg.de/haw/volltexte/2009/825/pdf/DimitarZlatkov_Thesis.pdf

f

[16] FriendlyId

https://github.com/norman/friendly_id

[17] Plataformatec, “devise”

<https://github.com/plataformatec/devise>

[18] Rspec

<https://github.com/rspec/rspec-rails>

[19] Capybara

<https://github.com/jnicklas/capybara>

Ανάπτυξη διαδικτυακής εφαρμογής με σκοπό την παροχή γνώσεων και πληροφοριών

[20] Selenium – Webdriver

<https://github.com/vertis/selenium-webdriver>

[21] Thoughtbot, “factory_girl”

https://github.com/thoughtbot/factory_girl